The **lwarp** package

LaTeX to HTML

v0.46 — 2018/01/23

© 2016–2018 Brian Dunn
bd@BDTechConcepts.com

Abstract

The lwarp package allows LaTeX to directly produce HTML5 output, using external utility programs only for the final conversion of text and images. Math may be represented by SVG files or MathJax.

Documents may be produced by pdfLaTeX, LuaLaTeX, or XeLaTeX. A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `latexmk`. Configuration is automatic at the first manual compile.

Print and HTML versions of each document may coexist, each with its own set of auxiliary files. Support files are self-generated on request. Assistance is provided for import into EPUB conversion software and word processors.

A modular package-loading system uses the lwarp version of a package for HTML when available. More than two hundred LaTeX packages are supported with these high-level source compatibility replacements, and many others work as-is.

A tutorial is provided to quickly introduce the user to the major components of the package.

To update existing projects, see section 2, Updates.

Note that this is still a "beta" version of lwarp, and some things may change in response to user feedback and further project development.

License:

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in http://www.latex-project.org/lppl.txt and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

1 Supporting TEX development

TEX and related projects:

- are mostly open-sourced and a volunteer effort;
- benefit students, academics, scientists, engineers, and businesses;
- help drive education, public and private research, and commercial activity;
- are used in the fields of mathematics, science, engineering, and humanities;
- span decades of development;
- are enduring — many older packages are still actively used and maintained;
- are largely backwards compatible;
- are portable across all the major computing platforms;
- are usable even on older computers and away from internet access;
- are continuing to maintain relevance with modern improvements;
- require no yearly subscription fees;
- and are supported by an active community of knowledgeable volunteers.

Please consider helping by joining and/or contributing to the TEX Users Group, a United States 501(c)(3) tax-exempt charitable organization. Contributions are accepted by credit card, check, or Pay Pal, via the United Way, or by USA or European bank transfer. Membership in TUG supports the development of TEXLive, the major TEX distribution.

Donations may be directed towards individual projects:

TUG Bursary Fund: Assistance for attending annual TUG meetings.
CTAN: The Comprehensive TEX Archive Network — Central storage for TEX.
TeX Development Fund: Support for specific projects.
EduTeX: Teaching and using TEX in schools and universities.
GUST e-foundry fonts: Enhanced for math and additional language groups.
LaTeX Project: Modernizing the LATEX core.
Libre Font Fund: Fonts, tools (FontForge), and distribution (the Open Font Library).
LuaTeX: Combining the pdfTEX engine and the Lua language.
MetaPost: Postscript graphics.
MacTeX: TEX for Mac.
PDF Accessibility: Modern PDF standards.
Other: Additional projects may be specified.

To make a contribution: https://www.tug.org/donate.html

For users of MiKTEX: https://miktex.org/donations.html

For country-specific TEX users groups: http://tug.org/usergroups.html

Contents

1 Supporting TEX development **2**

List of Figures **36**

List of Tables **36**

2 Updates **37**

3 Introduction **46**
3.1 Supported packages and features 48

4 Alternatives **52**
4.1 Internet class 52
4.2 TeX4ht 52
4.3 Translators 52
4.4 AsciiDoc and Asciidoctor 53
4.4.1 Asciidoctor-LaTeX 53
4.5 Pandoc 53
4.6 Word processors 53
4.7 Commercial systems 53
4.8 Comparisons 53

5 Installation **55**
5.1 Installing the lwarp package 57
5.2 Installing the `lwarpmk` utility 58
5.2.1 Using a local copy of lwarpmk 60
5.3 Installing additional utilities 60

6 Tutorial **63**
6.1 Starting a new project 63
6.2 Compiling the print version with `lwarpmk` 67
6.3 Compiling the HTML version with `lwarpmk` 68
6.4 Generating the SVG images 69
6.5 Using MathJax for math 70
6.6 Changing the CSS style 71
6.7 Customizing the HTML output 71
6.8 Using `latexmk` 72
6.9 Using XeLaTeX or LuaLaTeX 73
6.10 Using a glossary 74
6.11 Cleaning auxiliary files 75
6.12 Cleaning auxiliary and output files 75
6.13 Processing multiple projects in the same directory 75
6.14 Using the `make` utility 75

7 Additional details **76**

7.1 Font and UTF-8 support . . . 76
7.1.1 Indexes and UTF-8 . . . 77
7.2 lwarp package loading and options . . . 78
7.3 Customizing the HTML output . . . 80
7.3.1 Example HTML file naming . . . 84
7.4 Customizing the CSS . . . 85
7.5 Selecting the operating system . . . 86
7.6 Selecting actions for print or HTML output . . . 87
7.7 Commands to be placed into the `warpprint` environment . . . 88
7.8 Title page . . . 88
7.9 HTML page meta descriptions . . . 89
7.10 HTML page meta title . . . 90
7.11 HTML page meta author . . . 90
7.12 Modifying `xindy` index processing . . . 90

8 Special cases and limitations 91
8.1 Formatting . . . 91
8.1.1 Text formatting . . . 91
8.1.2 Horizontal space . . . 91
8.1.3 Text alignment . . . 91
8.1.4 Accents . . . 91
8.1.5 Textcomp . . . 92
8.1.6 Superscripts and other non-math uses of math mode . . . 92
8.1.7 Empty `\item` followed by a new line of text or a nested list: . . . 92
8.1.8 Filenames and URLs in lists or footnotes . . . 92
8.1.9 relsize package . . . 92
8.2 Boxes and minipages . . . 93
8.2.1 Marginpars . . . 93
8.2.2 Save Boxes . . . 93
8.2.3 Minipages . . . 93
8.2.4 Side-by-side minipages . . . 94
8.2.5 Framed minipages and other environments . . . 94
8.2.6 fancybox package . . . 95
8.2.7 mdframed package . . . 97
8.3 Cross-references . . . 97
8.3.1 Page references . . . 97
8.3.2 cleveref and varioref packages . . . 98
8.3.3 Hyperlinks, hyperref, and url . . . 98
8.3.4 Footnotes and page notes . . . 98
8.4 Front and back matter . . . 99
8.4.1 Starred chapters and sections . . . 99
8.4.2 abstract package . . . 99
8.4.3 titling and authblk . . . 100
8.4.4 tocloft package . . . 100
8.4.5 appendix package . . . 100
8.4.6 pagenote package . . . 100

8.4.7 endnotes package . . . 100
8.4.8 glossaries package . . . 100
8.4.9 Index and the tocbibind package . . . 101
8.5 Math . . . 102
8.5.1 Rendering tradeoffs . . . 102
8.5.2 SVG option . . . 103
8.5.3 MathJax option . . . 103
8.5.4 ntheorem package . . . 105
8.5.5 siunitx package . . . 105
8.5.6 units and nicefrac packages . . . 105
8.5.7 newtxmath package . . . 106
8.6 Graphics . . . 106
8.6.1 grffile package . . . 107
8.6.2 color package . . . 107
8.6.3 xcolor package . . . 107
8.6.4 epstopdf package . . . 108
8.6.5 overpic package . . . 108
8.7 Tabbing . . . 108
8.8 Tabular . . . 108
8.8.1 longtable package . . . 111
8.8.2 supertabular and xtab packages . . . 112
8.8.3 bigdelim package . . . 112
8.9 Floats . . . 112
8.9.1 float, trivfloat, and/or algorithmicx together . . . 112
8.9.2 caption and subcaption packages . . . 112
8.9.3 subfig package . . . 113
8.9.4 floatrow package . . . 113
8.9.5 keyfloat package . . . 113
8.10 Koma-Script . . . 114
8.11 Memoir . . . 114
8.12 Miscellaneous . . . 115
8.12.1 verse and memoir . . . 115
8.12.2 newclude package . . . 115
8.12.3 babel package . . . 116
8.12.4 todonotes and luatodonotes packages . . . 116
8.12.5 fixme . . . 116
8.12.6 xparse . . . 116

9 EPUB conversion 117

10 Word-processor conversion 119
10.1 Activating word-processor conversion . . . 119
10.2 Additional modifications . . . 121
10.3 Recommendations . . . 122
10.4 Limitations . . . 123

11 Modifying lwarp **125**
11.1 Creating an lwarp version of a package . . . 125
11.1.1 Adding a package to the lwarp.dtx file . . . 126
11.2 Testing lwarp . . . 126
11.3 Modifying `lwarpmk` . . . 126

12 Troubleshooting **128**
12.1 Using the `lwarp.sty` package . . . 128
12.1.1 Debug tracing output . . . 130
12.2 Compiling the `lwarp.dtx` file . . . 131

1 lwarp.sty **131**

13 Implementation **131**

14 Section depths and HTML headings **132**

15 Source Code **134**

16 Detecting the TEX Engine — pdflatex, lualatex, xelatex **135**

17 pdfLaTeX T1 and UTF8 encoding **135**

18 Unicode input characters **135**

19 Miscellaneous tools **136**

20 Early package requirements **136**

21 Operating-System portability **137**
21.1 Common portability code . . . 137
21.2 Unix, Linux, and Mac OS . . . 137
21.3 MS-Windows . . . 137

22 Package options **138**
22.1 Conditional compilation . . . 141

23 Package load order **142**
23.1 Tests of package load order . . . 142
23.2 Enforcing package loading after lwarp . . . 144

24 Required packages **148**

25 Loading packages **153**

26 File handles **156**

27 Include a file 156

28 Copying a file 157

29 Debugging messages 158

30 HTML-conversion output modifications 159
30.1 User-level controls . . . 159
30.2 Heading adjustments . . . 161

31 Remembering original formatting macros 163

32 Accents 164

33 Configuration Files 166
33.1 project_html.tex . . . 166
33.2 lwarpmk.conf . . . 166
33.3 project.lwarpmkconf . . . 167
33.4 lwarp.css . . . 168
33.5 lwarp_sagebrush.css . . . 195
33.6 lwarp_formal.css . . . 200
33.7 sample_project.css . . . 205
33.8 lwarp.xdy . . . 205
33.9 lwarp_mathjax.txt . . . 206
33.10 lwarpmk option . . . 208

34 Stacks 219
34.1 Assigning depths . . . 220
34.2 Closing actions . . . 220
34.3 Closing depths . . . 221
34.4 Pushing and popping the stack . . . 221

35 Data arrays 223

36 Sanitizing labels and filenames 223

37 HTML entities 224

38 HTML filename generation 225

39 Homepage link 227

40 `\LWRPrintStack` diagnostic tool 228

41 Closing stack levels 228

42 PDF pages and styles 229

43 HTML tags, spans, divs, elements 231
43.1 Mapping LaTeX Sections to HTML Sections . . . 231
43.2 Babel-French . . . 231
43.3 HTML tags . . . 232
43.4 Block tags and comments . . . 235
43.5 Div class and element class . . . 236
43.6 Single-line elements . . . 237
43.7 HTML5 semantic elements . . . 237
43.8 High-level block and inline classes . . . 238
43.9 Closing HTML tags . . . 239

44 Paragraph handling 240

45 Paragraph start/stop handling 244

46 Page headers and footers 246

47 CSS 247

48 Title, HTML meta author, HTML meta description 247

49 Footnotes 249
49.1 Regular page footnotes . . . 249
49.2 Minipage footnotes . . . 249
49.3 Titlepage thanks . . . 249
49.4 Regular page footnote implementation . . . 250
49.5 Minipage footnote implementation . . . 252
49.6 Printing pending footnotes . . . 253

50 Marginpars 254

51 Splitting HTML files 255

52 Sectioning 261
52.1 User-level starred section commands . . . 261
52.2 Book class commands . . . 262
52.3 Sectioning support macros . . . 262
52.4 \section and friends . . . 269

53 Starting a new file 272

54 Starting HTML output 274

55 Ending HTML output 277

56 Title page 279
56.1 Setting the title, etc. . . . 280
56.2 Changes for `\affiliation` . . . 281

56.3 Printing the thanks . . . 282
56.4 Printing the title, etc. in HTML . . . 282
56.5 Printing the title, etc. in print form . . . 283
56.6 \maketitle for HTML output . . . 284
56.7 \published and \subtitle . . . 287

57 Abstract 288

58 Quote and verse 289
58.1 Citations and attributions . . . 289
58.2 Quotes, quotations . . . 290
58.3 Verse . . . 290

59 Verbatim and tabbing 291

60 Theorems 294

61 Lists 295
61.1 List environment . . . 295
61.2 Itemize . . . 299
61.3 Enumerate . . . 300
61.4 Description . . . 300
61.5 Patching the lists . . . 301

62 Tabular 302
62.1 Limitations . . . 302
62.2 Token lookahead . . . 305
62.3 Tabular variables . . . 306
62.4 Handling &, @, !, and bar . . . 309
62.4.1 Localizing & catcodes . . . 310
62.4.2 Handling & . . . 311
62.4.3 Filling an unfinished row . . . 312
62.5 Handling \\ . . . 314
62.6 Parsing @, >, <, !, bar columns . . . 314
62.7 Parsing 'l', 'c', or 'r' columns . . . 317
62.8 Parsing 'p', 'm', or 'b' columns . . . 318
62.9 Parsing 'D' columns . . . 318
62.10 Parsing the column specifications . . . 319
62.11 Starting a new row . . . 323
62.12 Printing vertical bar tags . . . 324
62.13 Printing at or bang tags . . . 324
62.14 Data opening tag . . . 325
62.15 Midrules . . . 327
62.16 Multicolumns . . . 332
62.16.1 Parsing multicolumns . . . 332
62.16.2 Multicolumn factored code . . . 335
62.16.3 Multicolumn . . . 337

62.16.4 Longtable captions . . . 338
62.16.5 Counting HTML tabular columns . . . 340
62.17 Multicolumnrow . . . 342
62.18 Utility macros inside a table . . . 343
62.19 Special-case tabular markers . . . 343
62.20 Checking for a new table cell . . . 345
62.21 \mrowcell . . . 347
62.22 \mcolrowcell . . . 347
62.23 New \tabular definition . . . 348

63 Cross-references 353
63.1 Setup . . . 353
63.2 Zref setup . . . 355
63.3 Labels . . . 357
63.4 References . . . 358
63.5 Hyper-references . . . 361

64 Floats 364
64.1 Float captions . . . 365
64.1.1 Caption inside a float environment . . . 367
64.1.2 Caption and LOF linking and tracking . . . 367

65 Table of Contents, LOF, LOT 370
65.1 Reading and printing the TOC . . . 371
65.2 High-level TOC commands . . . 374
65.3 Side TOC . . . 375
65.4 Low-level TOC line formatting . . . 376

66 Index and glossary 379

67 Bibliography presentation 381

68 Restoring original formatting 382

69 Math 384
69.1 Limitations . . . 384
69.1.1 Rendering tradeoffs . . . 384
69.1.2 SVG option . . . 384
69.1.3 MathJax option . . . 385
69.2 Inline and display math . . . 386
69.3 MathJax support . . . 389
69.4 Equation environment . . . 392
69.5 AMS Math environments . . . 394
69.5.1 Support macros . . . 394
69.5.2 Environment patches . . . 395

70 Lateximages 402

70.1 Description . . . 402
70.2 Support counters and macros . . . 403
70.3 Font size . . . 403
70.4 Sanitizing math expressions for HTML . . . 404
70.5 Equation numbers . . . 405
70.6 HTML <alt> tags . . . 406
70.7 lateximage . . . 407

71 center, flushleft, flushright 411

72 Pre-loaded packages 412

73 Siunitx 412

74 Graphics print-mode modifications 413
74.1 General limitations . . . 413
74.2 Print-mode modifications . . . 414

75 Xcolor boxes 414

76 Cleveref 418

77 Picture 420

78 Boxes and Minipages 420
78.1 Counters and lengths . . . 421
78.2 Footnote handling . . . 422
78.3 Minipage handling . . . 422
78.4 Parbox, makebox, framebox, fbox, raisebox . . . 426

79 Direct formatting 431

80 Skips, spaces, font sizes 437

81 \phantomsection 445

82 \LaTeX and other logos 445
82.1 HTML logos . . . 445
82.2 Print logos . . . 447

83 \AtBeginDocument, \AtEndDocument 448

84 Koma-script 448

85 Memoir 449

86 Trademarks 450

2 lwarp-a4.sty 451
87 a4 451

3 lwarp-a4wide.sty 451
88 a4wide 451

4 lwarp-a5comb.sty 451
89 a5comb 451

5 lwarp-abstract.sty 451
90 abstract 451

6 lwarp-adjmulticol.sty 454
91 adjmulticol 454

7 lwarp-addlines.sty 455
92 addlines 455

8 lwarp-afterpage.sty 455
93 afterpage 455

9 lwarp-algorithmicx.sty 456
94 algorithmicx 456

10 lwarp-alltt.sty 457
95 alltt 457

11 lwarp-amsthm.sty 457
96 amsthm 457

12 lwarp-anonchap.sty 461

97 anonchap 461

13 lwarp-anysize.sty 462

98 anysize 462

14 lwarp-appendix.sty 462

99 appendix 462

15 lwarp-arabicfront.sty 463

100 arabicfront 463

16 lwarp-array.sty 463

101 array 463

17 lwarp-atbegshi.sty 463

102 atbegshi 463

18 lwarp-authblk.sty 464

103 authblk 464

19 lwarp-backref.sty 465

104 backref 465

20 lwarp-balance.sty 465

105 balance 465

21 lwarp-bigdelim.sty 466

106 bigdelim 466

22 lwarp-bigstrut.sty 467
107 bigstrut 467

23 lwarp-bookmark.sty 467
108 bookmark 467

24 lwarp-booktabs.sty 468
109 booktabs 468

25 lwarp-boxedminipage.sty 468
110 boxedminipage 468

26 lwarp-boxedminipage2e.sty 469
111 boxedminipage2e 469

27 lwarp-breakurl.sty 469
112 breakurl 469

28 lwarp-cancel.sty 470
113 cancel 470

29 lwarp-caption2.sty 471
114 caption2 471

30 lwarp-ccaption.sty 471
115 ccaption 471

31 lwarp-changebar.sty 471
116 changebar 471

32 lwarp-changepage.sty 472

117 changepage 472

33 lwarp-chngpage.sty 472

118 chngpage 472

34 lwarp-chappg.sty 473

119 chappg 473

35 lwarp-cite.sty 473

120 cite 473

36 lwarp-color.sty 474

121 color 474

37 lwarp-continue.sty 474

122 continue 474

38 lwarp-crop.sty 474

123 crop 474

39 lwarp-cuted.sty 475

124 cuted 475

40 lwarp-cutwin.sty 475

125 cutwin 475

41 lwarp-dblfnote.sty 476

126 dblfnote 476

42 lwarp-dcolumn.sty 476

127 dcolumn 476

43 lwarp-draftwatermark.sty 477

128 draftwatermark 477

44 lwarp-easy-todo.sty 477

129 easy-todo 477

45 lwarp-ebook.sty 478

130 ebook 478

46 lwarp-ellipsis.sty 478

131 ellipsis 478

47 lwarp-emptypage.sty 479

132 emptypage 479

48 lwarp-endfloat.sty 479

133 endfloat 479

49 lwarp-endnotes.sty 479

134 endnotes 479

50 lwarp-enumerate.sty 480

135 enumerate 480

51 lwarp-enumitem.sty 480

136 enumitem 480

52 lwarp-epigraph.sty 481

137 epigraph 481

53 lwarp-eso-pic.sty 482

138 eso-pic 482

54 lwarp-everypage.sty 483

139 everypage 483

55 lwarp-everyshi.sty 483

140 everyshi 483

56 lwarp-extramarks.sty 484

141 extramarks 484

57 lwarp-fancybox.sty 484

142 fancybox 484

58 lwarp-fancyhdr.sty 488

143 fancyhdr 488

59 lwarp-fancyref.sty 489

144 fancyref 489

60 lwarp-fancyvrb.sty 490

145 fancyvrb 490

61 lwarp-figcaps.sty 496

146 figcaps 496

62 lwarp-fix2col.sty 496

147 fix2col 496

63 lwarp-fixme.sty 496

148 fixme 496

64 lwarp-fixmetodonotes.sty 498

149 fixmetodonotes 498

65 lwarp-flafter.sty 498

150 flafter 498

66 lwarp-float.sty 499

151 float and \newfloat 499

67 lwarp-floatflt.sty 500

152 floatflt 500

68 lwarp-floatpag.sty 501

153 floatpag 501

69 lwarp-floatrow.sty 501

154 floatrow 501

70 lwarp-fltrace.sty 507

155 fltrace 507

71 lwarp-flushend.sty 507

156 flushend 507

72 lwarp-fncychap.sty 508

157 fncychap 508

73 lwarp-fnpos.sty 508

158 fnpos 508

74 lwarp-fontenc.sty 509

159 fontenc 509

75 lwarp-fontspec.sty 509

160 fontspec 509

76 lwarp-footmisc.sty 509

161 footmisc 509

77 lwarp-footnote.sty 510

162 footnote 510

78 lwarp-footnotehyper.sty 512

163 footnotehyper 512

79 lwarp-footnpag.sty 512

164 footnpag 512

80 lwarp-framed.sty 512

165 framed 512

81 lwarp-ftnright.sty 515

166 ftnright 515

82 lwarp-fullpage.sty 515

167 fullpage 515

83 lwarp-fullwidth.sty 515

168 fullwidth 515

84 lwarp-fwlw.sty 516

169 fwlw 516

85 lwarp-geometry.sty 516

170 geometry 516

86 lwarp-glossaries.sty 517

171 glossaries 517

87 lwarp-graphics.sty 518

172 graphics 518
172.1 Graphics extensions 518
172.2 Length conversions and graphics options 519
172.3 Printing HTML styles 522
172.4 `\includegraphics` 522
172.5 Boxes 527

88 lwarp-graphicx.sty 530

173 graphicx 530

89 lwarp-grffile.sty 530

174 grffile 530

90 lwarp-hang.sty 530

175 hang 530

91 lwarp-hanging.sty 532

176 hanging 532

92 lwarp-hyperref.sty 533

177 hyperref 533

93 lwarp-hyperxmp.sty 537

178 hyperxmp 537

94 lwarp-idxlayout.sty 537

179 idxlayout 537

95 lwarp-ifoddpage.sty 538

180 ifoddpage 538

96 lwarp-indentfirst.sty 539

181 indentfirst 539

97 lwarp-inputenc.sty 539

182 inputenc 539

98 lwarp-keyfloat.sty 539

183 keyfloat 539

99 lwarp-layout.sty 541

184 layout 541

100 lwarp-letterspace.sty 541

185 letterspace 541

101 lwarp-lettrine.sty 542

186 lettrine 542

102 lwarp-lips.sty 543

187 lips 543

103 lwarp-listings.sty 543

188 listings 543

104 lwarp-longtable.sty 547

189 longtable 547

105 lwarp-lscape.sty 549

190 lscape 549

106 lwarp-ltcaption.sty 549

191 ltcaption 549

107 lwarp-ltxtable.sty 549

192 ltxtable 549

108 lwarp-luatodonotes.sty 550

193 luatodonotes **550**

109 lwarp-marginfit.sty 552

194 marginfit **552**

110 lwarp-marginfix.sty 552

195 marginfix **552**

111 lwarp-marginnote.sty 553

196 marginnote **553**

112 lwarp-mcaption.sty 553

197 mcaption **553**

113 lwarp-mdframed.sty 554

198 mdframed **554**
198.1 Limitations 554
198.2 Package loading 555
198.3 Patches 555
198.4 Initial setup 556
198.5 Color and length HTML conversion 556
198.6 Environment encapsulation 556
198.7 Mdframed environment 558
198.8 Titles and subtitles 559
198.9 New environments 562

114 lwarp-memhfixc.sty 564

199 memhfixc **564**

115 lwarp-metalogo.sty 565

200 metalogo **565**

116 lwarp-microtype.sty 565

201 microtype 565

117 lwarp-midfloat.sty 566

202 midfloat 566

118 lwarp-midpage.sty 566

203 midpage 566

119 lwarp-moreverb.sty 567

204 moreverb 567

120 lwarp-morewrites.sty 568

205 morewrites 568

121 lwarp-mparhack.sty 569

206 mparhack 569

122 lwarp-multicol.sty 569

207 multicol 569

123 lwarp-multirow.sty 570

208 multirow 570
208.1 Multirow 571
208.2 Combined multicolumn and multirow 572

124 lwarp-nameref.sty 573

209 nameref 573

125 lwarp-natbib.sty 573

210 natbib **573**

126 lwarp-needspace.sty 574

211 needspace **574**

127 lwarp-newclude.sty 575

212 newclude **575**

128 lwarp-newunicodechar.sty 575

213 newunicodechar **575**

129 lwarp-nextpage.sty 575

214 nextpage **575**

130 lwarp-nonumonpart.sty 576

215 nonumonpart **576**

131 lwarp-nopageno.sty 576

216 nopageno **576**

132 lwarp-nowidow.sty 576

217 nowidow **576**

133 lwarp-ntheorem.sty 577

218 ntheorem **577**
218.1 Limitations 577
218.2 Options 577
218.3 Remembering the theorem style 578
218.4 HTML cross-referencing 581
218.5 `\newtheoremstyle` 581
218.6 Standard styles 582

218.7 Additional objects . . . 583
218.8 Renewed standard configuration . . . 584
218.9 amsthm option . . . 585
218.10 Ending a theorem . . . 588
218.11 \NoEndMark . . . 588
218.12 List-of . . . 588
218.13 Symbols . . . 589
218.14 Cross-referencing . . . 589

134 lwarp-overpic.sty 589

219 overpic 589

135 lwarp-pagenote.sty 590

220 pagenote 590

136 lwarp-pagesel.sty 590

221 pagesel 590

137 lwarp-paralist.sty 590

222 paralist 590

138 lwarp-parskip.sty 591

223 parskip 591

139 lwarp-pdflscape.sty 591

224 pdflscape 591

140 lwarp-pdfsync.sty 591

225 pdfsync 591

141 lwarp-pfnote.sty 592

226 pfnote 592

142 lwarp-placeins.sty 592

227 placeins 592

143 lwarp-prelim2e.sty 592

228 prelim2e 592

144 lwarp-prettyref.sty 593

229 prettyref 593

145 lwarp-preview.sty 593

230 preview 593

146 lwarp-quotchap.sty 593

231 quotchap 593

147 lwarp-ragged2e.sty 595

232 ragged2e 595

148 lwarp-realscripts.sty 595

233 realscripts 595

149 lwarp-relsize.sty 596

234 relsize 596

150 lwarp-romanbar.sty 598

235 romanbar 598

151 lwarp-romanbarpagenumber.sty 598

236 romanbarpagenumber 598

152 lwarp-rotating.sty 598

237 rotating 598

153 lwarp-rotfloat.sty 599

238 rotfloat 599

154 lwarp-savetrees.sty 600

239 savetrees 600

155 lwarp-scalefnt.sty 600

240 scalefnt 600

156 lwarp-scrextend.sty 600

241 scrextend 600

157 lwarp-scrhack.sty 603

242 scrhack 603

158 lwarp-scrlayer.sty 603

243 scrlayer 603

159 lwarp-scrlayer-notecolumn.sty 605

244 scrlayer-notecolumn 605

160 lwarp-scrlayer-scrpage.sty 606

245 scrlayer-scrpage 606

161 lwarp-section.sty 607

246 section 607

162 lwarp-sectsty.sty 608

247 sectsty 608

163 lwarp-setspace.sty 608

248 setspace 608

164 lwarp-shadow.sty 609

249 shadow 609

165 lwarp-showidx.sty 610

250 showidx 610

166 lwarp-showkeys.sty 610

251 showkeys 610

167 lwarp-sidecap.sty 610

252 sidecap 610

168 lwarp-sidenotes.sty 611

253 sidenotes 611

169 lwarp-siunitx.sty 613

254 siunitx 613

170 lwarp-soul.sty 614

255 soul 614

171 lwarp-soulpos.sty 616

256 soulpos 616

172 lwarp-soulutf8.sty 617

257 soulutf8 617

173 lwarp-stabular.sty 617

258 stabular 617

174 lwarp-subfig.sty 618

259 subfig 618

175 lwarp-subfigure.sty 624

260 subfigure 624

176 lwarp-supertabular.sty 624

261 supertabular 624

177 lwarp-syntonly.sty 626

262 syntonly 626

178 lwarp-tabls.sty 626

263 tabls 626

179 lwarp-tabularx.sty 627

264 tabularx 627

180 lwarp-tabulary.sty 627

265 tabulary 627

181 lwarp-textarea.sty 628

266 textarea 628

182 lwarp-textcomp.sty 628

267 textcomp 628
267.1 Limitations . . . 628
267.2 Package loading . . . 629
267.3 Remembering original defintions . . . 629
267.4 HTML symbols . . . 630
267.4.1 pdfLaTeX symbols . . . 630
267.4.2 XeLaTeX and LuaLaTeX symbols . . . 630
267.5 HTML dicritics . . . 631
267.6 Inside a `lateximage` . . . 631

183 lwarp-textfit.sty 632

268 textfit 632

184 lwarp-textpos.sty 633

269 textpos 633

185 lwarp-theorem.sty 633

270 theorem 633
270.1 Remembering the theorem style . . . 634
270.2 CSS patches . . . 635

186 lwarp-threeparttable.sty 638

271 threeparttable 638

187 lwarp-tikz.sty 639

272 tikz 639

188 lwarp-titleps.sty 640

273 titleps 640

189 lwarp-titleref.sty 643

274 titleref 643

190 lwarp-titlesec.sty 643

275 titlesec 643

191 lwarp-titletoc.sty 645

276 titletoc 645

192 lwarp-titling.sty 647

277 titling 647

193 lwarp-tocbasic.sty 652

278 tocbasic 652

194 lwarp-tocbibind.sty 652

279 tocbibind 652

195 lwarp-tocloft.sty 654

280 tocloft 654

196 lwarp-tocstyle.sty 661

281 tocstyle 661

197 lwarp-todo.sty 661

282 todo 661

198 lwarp-todonotes.sty 662

283 todonotes 662

199 lwarp-transparent.sty 664

284 transparent 664

200 lwarp-trivfloat.sty 665

285 trivfloat 665
285.1 Combining \newfloat, \trivfloat, and algorithmicx 666

201 lwarp-turnthepage.sty 666

286 turnthepage 666

202 lwarp-typearea.sty 666

287 typearea 666

203 lwarp-ulem.sty 667

288 ulem 667

204 lwarp-upref.sty 669

289 upref 669

205 lwarp-verse.sty 669

290 verse 669

206 lwarp-wallpaper.sty 671

291 wallpaper 671

207 lwarp-watermark.sty 672

292 watermark **672**

208 lwarp-wrapfig.sty 672

293 wrapfig **672**

209 lwarp-xcolor.sty 674

294 xcolor **674**
294.1 Limitations . . . 674
294.2 Xcolor definitions: location and timing . . . 674
294.3 Package loading . . . 676
294.4 Remembering and restoring original definitions . . . 677
294.5 HTML color style . . . 677
294.6 HTML border . . . 679
294.7 High-level macros . . . 679

210 lwarp-xfrac.sty 683

295 xfrac **683**

211 lwarp-xltxtra.sty 685

296 xltxtra **685**

212 lwarp-xmpincl.sty 686

297 xmpincl **686**

213 lwarp-xtab.sty 686

298 xtab **686**

214 lwarp-zwpagelayout.sty 688

299 zwpagelayout **688**

215 lwarp-patch-komascript.sty 689

300 patch-komascript **689**

216 lwarp-patch-memoir.sty 691

301 patch-memoir **691**
301.1 Packages . . . 692
301.2 Preliminary setup . . . 693
301.3 Laying out the page . . . 694
301.4 Text and fonts . . . 696
301.5 Titles . . . 697
301.6 Abstracts . . . 697
301.7 Docment divisions . . . 697
301.8 Pagination and headers . . . 700
301.9 Paragraphs and lists . . . 702
301.10 Contents lists . . . 702
301.11 Floats and captions . . . 707
301.12 Page notes . . . 712
301.13 Decorative text . . . 714
301.14 Poetry . . . 714
301.15 Boxes, verbatims and files . . . 714
301.16 Cross referencing . . . 715
301.17 Back matter . . . 716
301.18 Miscellaneous . . . 716
301.19 Final patchwork . . . 717

Change History and Index 718

List of Figures

1 tutorial.tex listing 64

List of Tables

1 LaTeX–HTML generation — lwarp package — Supported functions . . 48
2 Required software programs 56
3 Files created along with the print version 66
4 Package options 78
5 HTML settings 80
6 Section HTML headings for word-processor conversion 123
7 Section depths and HTML headings 132
8 Tabular baseline 319
9 Tabular HTML column conversions 320
10 Cross-referencing data structures 354
11 Float data structures 364
12 AMSthm package — CSS styling of theorems and proofs 458
13 Ntheorem package — CSS styling of theorems and proofs 577
14 Theorem package — CSS styling of theorems and proofs 634

2 Updates

The following is intended for those updating existing projects which use lwarp, highlighting any special changes which must be made due to improvements or modifications in lwarp itself.

For a detailed list of changes, see the Change History on page 718.

v0.46:

⚠ name change
- `\PrintStack` changed to `\LWRPrintStack`.

misc. fixes
- Fix: Empty lines between `tabular` rows.
- Fix: Stack unnesting.
- Fix: SVG math and `lateximages` in numerous situations (but still more to do).
- Fix: Spaces in `\usepackage`.
- Fix: Now allows MathJax inside `verse`.

v0.45:

docs
- Improved MikTeX install instructions.
- Improved graphics and epstopdf instructions.
- Updates to the Introduction.

memoir
- Added memoir, memhfixc. See section 8.11.

cross-references
- Fix: Now allows underscores in labels.
- Fix: `\_` and `\<blank>` in section/file names.

math
- Fix: Now allows MathJax inside `tabbing`.

bibliography
- Fix: Bibliography `\em` names.
- Added cite, natbib, backref. (Also works as-is with biblatex.)

misc. fixes
- Fix: Empty lines between `tabular` rows.
- Fix: "Improper `\prevdepth`" with minipages, lists.
- Fix: Incorrect SVG math and `lateximages` with subfig.
- Fix: Lateximages from incorrect pages with Mathjax.
- Fix: Missing sidetoc if using listings.
- Fix: Added an array emulation package.

packages
- Added subfigure, prettyref, hanging, midpage, flafter, fltrace, changebar, endfloat, continue, fwlw, turnthepage, footnpag, pagesel, textfit, titleref.

v0.44:

koma-script
- Added koma-script classes (except scrlttr2, scrjura).
- Added scrextend, scrlayer, scrlayer-notecolumn, scrlayer-scrpage, scrhack, tocstyle, tocbasic.

HTML title and author
- Added `\HTMLTitle`. Fixed web page title if `\HTMLTitle` empty and no `\title` given and not using titling package.
- Fixed web page author if `\HTMLauthor` is empty and `\author` is not given.

encodings
- If using `pdflatex`, automatically loads T1 and UTF8 encodings. (Additional fontenc encodings may be loaded after lwarp.)

lists
- Added `list` and `trivlist` environments, hang.

tabular
- Fix: `\multicolumn` alignment if formatting for a word processor.
- Added ltxtable.

math
- Fix: MathJax combined with `lateximages`.
- algorithmicx: Improved comment symbol and floating.

packages
- Completed todonotes and luatodonotes.
- Added todo, easy-todo, fixmetodonotes, fixme.
- Added soulutf8, soulpos, cancel.
- Added section, fancyref, ifoddpage.
- Added preview, atbegshi, watermark.
- Improved tocloft `\newlistof` and `\newlistentry`.

v0.43:

- Docs: Reorganized HTML customization, added an HTML settings table. See section 7.3.

footnotes
- Added `FootnoteDepth` to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each `\subparagraph` or higher.

sectioning
- Fix: Expansion in section name.

`tabular`
- Fix: Ignore spaces in tabular column specification.
- Fix: Tabular rules at bottom or when finishing incomplete rows.
- Fix: `\multicolumn` at/bang/before/after specifications, trim, and vertical rules.
- Fix: supertabular and xtab column misalignment.

math
- Fix: `equation*`.
- Fix: SVG math in a section name.
- Fix: `\ref` and `\eqref` in SVG math.

packages
- Added todonotes and luatodonotes (but only `disabled`).
- Added breakurl.
- hyperref: Fix: Several macros were made robust, `\Gauge` added.

v0.42:

Support TEX!
- Added TEX development support page, Supporting TEX development.

word-processor conversion
- Improved assitance for word-processor conversions when boolean `FormatWP` is set `true`. See section 10.
  - ⚠ name change The boolean `FormatWordProcessor` has been renamed `FormatWP`.
  - ⚠ name change The boolean `HTMLMarkFloats` has been renamed `WPMarkFloats`.
  - New booleans control whether to place additional marks around minipages, at the table of contents, at the LOF and LOT, and whether to print math as LATEX source for copy/paste into the LibreOffice Writer TeXMaths extension.
  - Improved formatting for numerous objects. See section 10.

`tabbing`
- Add: `tabbing` environment.

overpic
- Add: overpic package. See section 219.

math
- Fix: Text copy/paste of $\mathcal{AMS}$ math environment numbers and names.
- Improved `\ensuremath`.
- MathJax with siunitx: Updated script and documentation.

symbols
- textcomp: Improved `\interrobangdown`.
- realscripts: Fix for subscripts in a `lateximage`.

load order
- morewrites: Enforces loading before lwarp.

v0.41:

`tabular`
- Added tabular vertical rules, subject to some limitations. See the rules section of section 8.8.
- Improved booktabs: Width and trim are honored.

⚠ new syntax
- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 208.2 on page 572.
- Fix: Unfinished tabular rows are automatically filled.
- Fix for tabular column specifiers while using babel-french. (`\NoAutoSpacing` is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

v0.40:

graphics, graphicx
- graphics and graphicx have been moved from the lwarp core, and are only loaded if requested with `\usepackage`.

`\includegraphics` path
⚠ image file extensions
- Improved graphics `\graphicspath` support. Multiple image directories may now be used. Refer to `.pdf` files without a file extension to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 8.6.

- grffile is now directly supported instead of emulated.

bigdelim
- Fix for bigdelim, and improved documentation. See section 106.

symbols
- Improved LaTeX and textcomp symbols.

fixes
- Fix for LaTeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for xltxtra with XeLaTeX.
- Fixes for tocbibind with `\simplechapter`, etc.
- Fixes for `\multicolumnrow` and `\nullfonts` with older versions of multirow and xparse.
- Added `\underline`.

margins
- Added adjmulticol.

columns
- Added cuted, midfloat.

footnotes
- Added pfnote, fnpos, dblfnote.

tabular
- Added stabular, tabls.

sectioning
- Added sectsty, anonchap, quotchap.

v0.39:

title pages
- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. titling is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubtitlePublished` replicates them using titling. See section 56.7. authblk is added, and should be loaded before titling. See section 56.

⚠ `\published` and `\subtitle`

⚠ load order

tabular
- `\multirow` now supports the new optional `vpos` argument.

multi column/row cell
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 208.2.

- Tabular special cases:
  - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 8.8.
  - Added `\ResumeTabular` for use when a `tabular` environment is defined inside another environment. See section 8.8.

⚠ macros inside tabular

⚠ tabular defined inside another environment

tabular
- Added supertabular, xtab, bigstrut, bigdelim.

margins
- Added fullwidth.

page layout
- Added addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

v0.38:

forced single-pass compile
- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.

starred sections
- Added \ForceHTMLPage and \ForceHTMLTOC to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 8.4.1.

updated tutorial
- Modified the tutorial to use the new \ForceHTMLPage and \ForceHTMLTOC macros.

packages
- Added appendix, tocbibind, fncychap, fix2col.

font size
- Added relsize, scalefnt.
- Added realscripts, metalogo, xltxtra.
- Added grffile, romanbar.

page numbering
- Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.

front & back matter
- Docs: Improved description of the use of front/back matter. See section 8.4.
- Fix: color requests xcolor.
- Fix: \part for article class.

v0.37:

\include for HTML
- \include now maintains independent .aux files for HTML versions.

latexmk
- comment, used by lwarp, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.

accents and symbols
- Improved support for LaTeX accents, textcomp, siunitx symbols.

babel-french
- Improved babel-french handling for load order and ~ tilde.

v0.36:

- Recorganized the documentation section regarding special cases and limitations. (Section 8)
- Improved source formatting.

boxes and frames
- \fbox and related now use \fboxsep and \fboxrule.
- \makebox and \framebox now use width and position.
- \fcolorbox and related now work inside a lateximage.

babel-french
- babel-french: Improvements for French variants, load order, footnotes, ellipses.

footnotes
- Improved footnote numbering. lateximage footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.

siunitx
- Improved siunitx units.
- Fix for filenames while using MathJax.
- Fix for \rule when xcolor is not loaded.
- Added transparent, upref.

v0.35: Fix: `\textbf` and related.

v0.34:

⚠ Optional arguments

- `BlockClass`'s optional argument has been moved in front of the mandatory argument:

  `BlockClass[style]{class}` (NEW)

  instead of:

  `BlockClass{class}[style]` (OLD)

  This change makes it more consistent with LaTeX standards, and avoids problems with space between arguments.

⚠ Optional arguments

- Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:

  `\InlineClass[style]{class}{text}`

spans with minipages

framing minipages

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML fomatting. See section 8.2.3. `\fboxBlock` is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 8.2.5. `\fbox` and minipages now often work in SVG math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.

`lateximage`, SVG math, `tabular`

- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, booktabs.
- Improved font control for `lateximagees` and SVG math.

`eqnarray`

- Added the `eqnarray` environments.

verbatim packages

- fancyvrb is no longer required (preloaded), but is still supported.
- Added verbatim and moreverb.

framing packages

- Added fancybox, boxedminipage2e and shadow.

list packages

- enumitem is no longer required, but is still supported.
- Added enumerate and paralist.
- titleps is no longer required, but is still supported.
- Added crop.
- Added rotfloat, marginfit, and several minor packages; see the change log.

babel-french

- Adds fixed-width HTML spaces around punctuation when using babel-french. LuaTeX does not yet use the extra punctuation spacing.

v0.33:

- Tabular `@` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 62.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for amsmath with SVG math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

v0.32: Bug fixes; no source changes needed:

- `lwarpmk` has been adjusted to work with the latest `luatex`.
- Spaces in the `\usepackage` and `\RequirePackage` package lists are now accepted and ignored.
- Fix for the glossaries package and `\glo@name`.

v0.31: Bug fix; no source changes needed:

- Improved compatibility with keyfloat, including the new `keywrap` environment.

v0.30:

⚠ lwarp-newproject

- lwarp-newproject has been removed, and its functions have been combined with lwarp.

  To modify existing documents, remove from the document source:

  ```
  \usepackage{lwarp-newproject}
  ```

  The lwarp package now produces the configuration files during print output, and also accepts the option `lwarpmk` if desired.

⚠ HTML setup changes.

- A number of macros related to HTML settings have been converted to options, and other macros and options have been renamed to create a consistent syntax:

Old Macro	New Package Option
\HomeHTMLFileName	HomeHTMLFilename
\HTMLFileName	HTMLFilename
\useLatexmk	latexmk
\warpOSwindows	OSWindows

Old Package Option	New Package Option
lwarpmklang	IndexLanguage
(new)	xdyFilename

Old Macro	New Macro
\MetaLanguage	\HTMLLanguage
\HTMLauthor	\HTMLAuthor
\NewHTMLdescription	\HTMLDescription
\SetFirstPageTop	\HTMLFirstPageTop
\SetPageTop	\HTMLPageTop
\SetPageBottom	\HTMLPageBottom
\NewCSS	\CSSFilename

- Per the above changes, in existing documents, modify the package load of lwarp, such as:

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={},
    IndexLanguage=english
]{lwarp}
```

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
  1. Make the changes shown above.
  2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
  3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `xdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 7.12.
- Improvements in index processing:
  - `xindy`'s language is now used for index processing as well as glossary.
  - Print mode without `latexmk` now uses `xindy` instead of `makeindex`.
  - `texindy`/`xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
  - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

- Add: `lwarpmklang` option for lwarp-newproject and lwarp. Sets the language to use while processing the glossary. (As of v0.30, this has been changed to the `IndexLanguage` option.)
- Fix: `\includegraphics` when no optional arguments.

v0.28:

- `\HTMLAuthor {`⟨*name*⟩`}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Booean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Name changed to `FormatWP` as of v0.42.
- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MathJax CDN repository.
- Adds tabulary.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglosssary` and `htmlglossary` actions.

3 Introduction

The lwarp project aims to allow a rich LaTeX document to be converted to a reasonable HTML5 interpretation, with only minor intervention on the user's part. No attempt has been made to force LaTeX to provide for every HTML-related possibility, and HTML cannot exactly render every possible LaTeX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

Several "modern" features of HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers shows that these new features degrade gracefully.

---

lwarp is a native LaTeX package, and operates by either patching or emulating various functions. Source-level compatibility is a major goal, but occasional user intervention is required in certain cases.

As a package running directly in LaTeX, lwarp has some advantages over other methods of HTML conversion. A deeper level of TeX understanding is of course possible, as TeX itself is still in use. Lua expressions are still available with LuaTeX. Entire categories of LaTeX packages work as-is when used with lwarp: definitions, file handling, utilities, internal data structures and calculations, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting Ti*k*z and `picture`, Numerous packages are easily adapted for HTML versions, either by loading and patching the originals, or by creating nullified or emulated replacements, and all without resorting to external programming. As a result, several hundred packages have already been adapted (table 1), and an uncounted number more work as-is.

Packages have been selected according to several criteria: importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives[1].

Assistance is also provide for modifying the HTML output to suit the creation of EPUB documents, and for modifying the HTML output to ease import into a word processor.

`pdflatex`, `xelatex`, or `lualatex` may be used, allowing lwarp to process the usual image formats. While generating HTML output, SVG files are used in place of PDF. Other formats such as JPG are used as-is.

[1] An amazing number of decades-old packages are still in modern use.

SVG images may be used for math, and are also used for `picture`, and Ti*k*z environments. This format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-ridden[2] browser support).

Furthermore, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by `<alt>` tags holding the LaTeX source for the expression, allowing it to be copy/pasted into other documents.[3] Custom LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside LaTeX.

The MathJax JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files.[4] lwarp maintains LaTeX control for cross-referencing and equation numbering, and attempts to force MathJax to tag equations accordingly.

A `texlua` program called `lwarpmk` is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from a LaTeX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

lwarp automatically generates the extra files necessary for the HTML conversion, such as CSS and `.xdy` files, and configuration files for the utility `lwarpmk`. Also included is a parallel version of the user's source document, `<sourcename>-html.tex`, which selects HTML output and then inputs the user's own source. This process allows both the printed and HTML versions to co-exist side-by-side, each with their own auxiliary files.

When requesting packages during HTML conversion, `lwarp` first looks to see if it has its own modified version to use instead of the standard LaTeX version. These `lwarp-packagename.sty` files contain code used to emulate or replace functions for HTML output.

---

[2] Firefox has had an on-again/off-again bug for quite some time regareding printing SVGs at high resolution.

[3] There seems to be some debate as to whether MathML is actually an improvement over LaTeX for sharing math. The author has no particular opinion on the matter, except to say that in this case LaTeX is much easier to implement!

[4] One SVG image file per math expression. A commmon scientific paper can easily run into several thousand files, depending on how often something like x is used. In testing one sample document it appears that hashing would only reduce the number in half.

3.1 Supported packages and features

Table 1 lists some of the various LaTeX features which may be used.

Table 1: LaTeX–HTML generation — lwarp package — Supported functions

Category	Status
Engines:	pdfLaTeX, XeLaTeX, LuaLaTeX
Classes:	book, report, article, scrbook, scrreprt, scrartcl, memoir.
Koma-script:	scrextend, scrlayer, scrhack. Others as listed below.
Memoir:	memhfixc
Page layout:	Emulates geometry, fancyhdr, titleps, scrlayer-scrpage, typearea, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, ebook, preview, draftwatermark, watermark, everyshi, atbegshi, continue, fwlw, turnthepage, pagesel.
Sectioning:	Supported, with hyperlinks. Adds `FileDepth` for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames. Emulates titlesec, fncychap, sectsty, section, anonchap, quotchap.
Table of contents, figures, tables:	Supported, with hyperlinks. Supports tocbibind. Emulates titletoc, tocloft, tocbasic, and tocstyle.
Title page:	`\maketitle`, `titlepage`, titling, authblk.
Front & back matter:	abstract, appendix.
Indexing:	`texindy` is used, with hyperlinks. idxlayout is emulated.
Glossary:	glossaries and `xindy` are used.
Bibliography:	cite, natbib, biblatex, backref.
Cross-references:	Emulated, with hyperlinks. hyperref, cleveref, varioref, fancyref, prettyref, titleref, url, breakurl.

lwarp Supported Functions — continued

Category	Status
Languages:	babel. (polyglossia is untested.)
Margin notes:	marginfit, marginfix, scrlayer-notecolumn.
Footnotes:	Adds `FootnoteDepth` to print footnotes at section breaks. footnote, footmisc, marginnote, sidenote, pagenote, endnotes, footnpag.
Math:	Supported. Converted to SVG images with HTML `<alt>` tags containing the LaTeX source for the math expression. MathJax supported as an alternative. $\mathcal{AMS}$ environments are supported. User-defined macros are available during converson, due to native LaTeX processing.
Theorems:	Support for native LaTeX theorems, plus theorem, amsthm, ntheorem.
Add'l math:	delarray, bm, math fonts via SVG images.
Units and fractions:	siunitx, xfrac, nicefrac, units
Floats:	Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, subfigure, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt, flafter, fltrace, endfloat.
Tabular	`tabular` environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, ltxtable, booktabs.
Graphics	graphics and graphicx are emulated. `\includegraphics` supports `width`, `height`, `origin`, `angle`, and `scale` tags, and adds `class`. References to PDF files are changed to SVG, other image types are accepted as well. `\rotatebox` and `\scalebox` are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture and tikz are converted to an SVG image. grffile and overpic are supported.

lwarp Supported Functions — continued

Category	Status
xcolor:	Supported. Full package color names, any color models, and mixing. `\textcolor`, `\colorbox`, `\fcolorbox`. Enhanced for HTML compatibility.
Lists:	Standard LaTeX environments are supported, along with enumitem, enumerate, paralist, hang.
Environments:	Standard LaTeX environments are supported.
`minipage`:	Supported with some HTML5-imposed limitations. Nested minipages are supported.
Quotations:	verse, csquotes, epigraph
Verbatim:	verbatim, moreverb, fancyvrb (except for verbatim footnotes), shortvrb.
Frames:	framed, fancybox, mdframed, boxedminipage2e, shadow.
multicol:	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
Margins:	midpage, hanging, fullwidth.
Todo notes:	todo, todonotes, easy-todo, fixmetodonotes, fixme, changebar.
Direct formatting:	`\emph`, `\textsuperscript`, `\textbf`, etc are supported. `\bfseries`, etc. are not yet supported. lettrine, ulem, soul, soulutf8, soulpos, cancel, relsize, scalefnt, textfit, and realscripts are supported.
Ordinals:	nth, fmtcount, and engord are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, `\enskip`, `\quad`, `\qquad`, `\hspace`.
Rules:	`\rule` with width, height, raise, text color.

lwarp Supported Functions — continued

Category	Status
HTML reserved characters:	`\&`, `\textless`, and `\textgreater` are converted to HTML entities.
Fonts:	Used as-is. Appear in SVG math expressions or embedded image environments.
Symbols:	Native LaTeX diacriticals, textcomp.
Working as-is:	Various utility, calculation, file, and text-only packags, such as calc, fileerr, somedefs, trace, xspace.

4 Alternatives

Summarized below are several other ways to convert a LaTeX or other document to HTML. Where an existing LaTeX document is to be converted to HTML, lwarp may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

4.1 Internet class

Cls `internet` The closest to lwarp in design principle is the `internet` class by Andrew Stacey (https://github.com/loopspace/latex-to-internet), an interesting project which directly produces several versions of markdown, and also HTML and EPUB.

4.2 TeX4ht

Prog `TeX4ht` http://tug.org/tex4ht/

Prog `htlatex`

This system uses native LaTeX processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In most cases, lwarp provides a better HTML conversion, while supporting more packages. TeX4ht produces several other forms of output beyond HTML.

4.3 Translators

These systems use external programs to translate a subset of LaTeX syntax into HTML. Search for each on CTAN (http:\ctan.org).

Prog `Hevea` **HEVEA:** http://hevea.inria.fr/ (not on CTAN)

Prog `TtH` **TTH:** http://hutchinson.belmont.ma.us/tth/

Prog `GELLMU` **GELLMU:** http://www.albany.edu/~hammond/gellmu/

Prog `LaTeXML` **LaTeXML:** http://dlmf.nist.gov/LaTeXML/

Prog `Plastex` **PlasTeX:** https://github.com/tiarno/plastex

Prog `LaTeX2HTML` **LaTeX2HTML:** http://www.latex2html.org/
and http://ctan.org/pkg/latex2html.

Prog `TeX2page` **TeX2page:** http://ds26gte.github.io/tex2page/index.html

Finally, GladTeX may used to directly insert LaTeX math into HTML:

Prog `GladTeX` **GladTeX:** http://humenda.github.io/GladTeX/

4.4 AsciiDoc and Asciidoctor

AsciiDoc is one of the most capable markup languages, providing enough features to produce the typical technical-writing document with cross-references, and it writes LaTeX and HTML.

Prog AsciiDoc

Asciidoctor: http://asciidoctor.org/ (More active.)

Prog AsciiDoctor

AsciiDoc: http://asciidoc.org/ (The original project.)

4.4.1 Asciidoctor-LaTeX

The Asciidoctor-LaTeX project is adding additional LaTeX-related features.

Asciidoctor-LateX:
http://www.noteshare.io/book/asciidoctor-latex-manual

Prog Asciidoctor-LaTeX

https://github.com/asciidoctor/asciidoctor-latex

4.5 Pandoc

Prog Pandoc

A markup system which also reads and writes LaTeX and HTML.

Pandoc: http://pandoc.org/

(Watch for improvements in cross-references to figures and tables.)

4.6 Word processors

Prog Word
Prog LibreOffice
Prog OpenOffice

It should be noted that the popular word processors have advanced through the years in their abilities to represent math with a LaTeX-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success. See recent developments in Microsoft ® Word ® and LibreOffice ™ Writer.

4.7 Commercial systems

Prog Adobe
Prog FrameMaker
Prog InDesign
Prog Flare
Prog Madcap

Likewise, several professional systems exist whose abilities have been advancing in the areas of typesetting, cross-referencing, and HTML generation. See Adobe ® FrameMaker ®, Adobe InDesign ®, and Madcap Flare ™.

4.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that LaTeX is comparably easy to learn, while LaTeX provides many more

advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

Text-based document-markup systems share some of the advantages of LaTeX vs. a typical word processor. Documents formats are stable. The documents themselves are portable, work well with revision control, do not crash or become corrupted, and are easily generated under program control. Formatting commands are visible, cross-referencing is automatic, and editing is responsive. Search/replace with regular expressions provides a powerful tool for the manipulation of both document contents and structure. Markup systems and some commercial systems allow printed output through a LaTeX back end, yielding high-quality results especially when the LaTeX template is adjusted, but they lose the ability to use LaTeX macros and other LaTeX source-document features.

The effort required to customize the output of each markup system varies. For print output, LaTeX configuration files are usually used. For HTML output, a CSS file will be available, but additional configuration may require editing some form of control file with a different syntax, such as XML. In the case of lwarp, CSS is used, and much HTML output is adjusted through the usual LaTeX optional macro parameters, but further customization may require patching LaTeX code.

The popular word processors and professional document systems each has a large base of after-market support including pre-designed styles and templates, and often include content-management systems for topic reuse.

5 Installation

Table 2 shows the tools which are used for the LaTeX to HTML conversion. In most cases, these will be available via the standard package-installation tools.

Detailed installation instructions follow.

Table 2: Required software programs

Provided by your LaTeX distribution:

From TeXLive: http://tug.org/texlive/.

LaTeX: `pdflatex`, `xelatex`, or `lualatex`.

The lwarp package: This package.

The `lwarpmk` utility: Provided along with this package. This should be an operating-system executable in the same way that `pdflatex` or `latexmk` is. It is possible to have the lwarp package generate a local copy of `lwarpmk` called `lwarpmk.lua`. See table 3.

luatex: Used by the `lwarpmk` program to simplify and automate document generation.

xindy: The xindy package is used by lwarp to create indexes. On a MiKTeX system this may have to be acquired separately, but it is part of the regular installer as of mid 2015.

latexmk: Optionally used by `lwarpmk` to compile LaTeX code. On a MiKTeX system, Perl may need to be installed first.

pdfcrop: Used to pull images out of the LaTeX PDF.

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

pdfseparate: Used to pull images out of the LaTeX PDF.

pdftocairo: Used to convert images to SVG.

These might be provided by your operating-system package manager.

From Poppler: poppler.freedesktop.org.

For MacOS®, see https://brew.sh/, install Homebrew, then

Enter ⇒ `brew install poppler`

For Windows, see:
https://sourceforge.net/projects/poppler-win32/ and:
http://blog.alivate.com.au/poppler-windows/

Perl:

This may be provided by your operating-system package manager, and is required for some of the Poppler PDF utilities.

strawberryperl.com (recommended), perl.org

Automatically downloaded from the internet as required:

MathJax: Optionally used to display math. From: mathjax.org

5.1 Installing the lwarp package

There are several ways to install lwarp. These are listed here with the preferred methods listed first:

Pre-installed: Try entering into a command line:

Enter ⇒ `kpsewhich lwarp.sty`

If a path to `lwarp.sty` is shown, then lwarp is already installed and you may skip to the next section.

TEX Live: If using a TEX Live distribution, try installing via `tlmgr`:

Enter ⇒ `tlmgr install lwarp`

MiKTEX: If using MiKTEX:

1. To install lwarp the first time, use the MikTeX Package Manager (Admin).
2. To update lwarp, use MikTeX Update (Admin).
3. Either way, also update the package `miktex-misc`, which will install and update the `lwarpmk` executable.

Operating-system package: The operating-system package manager may already have `lwarp`, perhaps as part of a set of TEX-related packages.

CTAN TDS archive: lwarp may be downloaded from the Comprehensive TEX Archive:

1. See http://ctan.org/pkg/lwarp for the lwarp package.
2. Download the TDS archive: `lwarp.tds.zip`
3. Find the TEX `local` directory:

   TEX Live:

   Enter ⇒ `kpsewhich -var-value TEXMFLOCAL`

   MiKTEX:

   In the "Settings" window, "Roots" tab, look for a local TDS root.

   This should be something like:

   `/usr/local/texlive/texmf-local/`
4. Unpack the archive in the TDS local directory.
5. Renew the cache:

   Enter ⇒ `mktexlsr`

   — or —

   Enter ⇒ `texhash`

   Or, for Windows MiKTEX, start the program called `MiKTeX Settings (Admin)` and click on the button called Refresh FNDB.

CTAN `.dtx` and `.ins` files: Another form of TEX package is the `.dtx` and `.ins` source files. These files are used to create the documentation and `.sty` files.

1. See http://ctan.org/pkg/lwarp for the lwarp package.
2. Download the zip archive `lwarp.zip` into your own `lwarp` directory.
3. Unpack `lwarp.zip`.
4. Locate the contents `lwarp.dtx` and `lwarp.ins`
5. Create the documentation:
   Enter ⇒ `pdflatex lwarp.dtx`
   (several times)
6. Create the `.sty` files:
   Enter ⇒ `pdflatex lwarp.ins`
7. Copy the `.sty` files somewhere such as the TEX Live local tree found in the previous CTAN TDS section, under the subdirectory:
   `<texlocal>/tex/latex/local/lwarp`
8. Copy the documentation `lwarp.pdf` to a `source` directory in the local tree, such as:
   `<texlocal>/doc/local/lwarp`
9. Renew the cache:
   Enter ⇒ `mktexlsr`
   — or —
   Enter ⇒ `texhash`
   Or, for Windows MiKTEX, start the program called `MiKTeX Settings (Admin)` and click on the button called Refresh FNDB.
10. See section 5.2.1 to generate your local copy of `lwarpmk`.
11. Once the local version of `lwarpmk.lua` is installed, it may be made available system-wide as per section 5.2.

Project-local CTAN `.dtx` and `.ins` files: The `.dtx` and `.ins` files may be downloaded to a project directory, then compiled right there, alongside the document source files. The resultant `*.sty` and `lwarpmk.lua` files may be used as-is, so long as they are in the same directory as the document source. This approach is especially useful if you would like to temporarily test lwarp before deciding whether to permanently install it.

Just testing!

5.2 Installing the `lwarpmk` utility

(Note: If `lwarpmk` is not already installed, it is easiest to use a local copy instead of installing it system-wide. See section 5.2.1.)

After the lwarp package is installed, you may need to setup the `lwarpmk` utility:

1. At a command line, try executing `lwarpmk`. If the `lwarpmk` help message appears, then `lwarpmk` is already set up. If not, it is easiest to generate and use a local copy. See section 5.2.1.

2. For MiKTEX, try updating the `miktex-misc` package. This may install the `lwarpmk` executable for you.

   Otherwise, continue with the following:

3. Locate the file `lwarpmk.lua`, which should be in the `scripts` directory of the TDS tree. On a TEX Live or MiKTEX system you may use

   Enter ⇒ `kpsewhich lwarpmk.lua`

   (If the file is not found, you may also generate a local copy and use it instead. See section 5.2.1.)

4. Create `lwarpmk`:

   Unix: Create a symbolic link and make it executable:

   (a) Locate the TEX Live binaries:

   Enter ⇒ `kpsewhich -var-value TEXMFROOT`

   This will be something like:

   `/usr/local/texlive/<year>`

   The binaries are then located in the `bin/<arch>` directory under the root:

   `/usr/local/texlive/<year>/bin/<architecture>/`

   In this directory you will find programs such as `pdflatex` and `makeindex`.

   (b) In the binaries directory, create a new symbolic link from the binaries directory to `lwarpmk.lua`:

   Enter ⇒ `ln -s <pathtolwarpmk.lua> lwarpmk`

   (c) Make the link executable:

   Enter ⇒ `chmod 0755 lwarpmk`

   Windows TEX Live: Create a new `lwarpmk.exe` file:

   (a) Locate the TEX Live binaries as shown above for Unix.

   (b) In the binaries directory, make a *copy* of `runscript.exe` and call it `lwarpmk.exe` This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

   Windows MiKTEX: Create a new `lwarpmk.bat` file:

   (a) Locate the binaries. These will be in a directory such as:

   `C:\Program Files\MiKTeX 2.9\miktex\bin\x64`

   In this directory you will find programs such as `pdflatex.exe` and `makeindex.exe`.

   (b) Create a new file named `lwarpmk.bat` containing:

   ```
   texlua "C:\Program Files\MiKTeX 2.9\scripts\lwarp\lwarp.texlua" %*
   ```

   This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

5.2.1 Using a local copy of lwarpmk

It is also possible to use a local version of `lwarpmk`:

1. When compiling the tutorial in section 6, use the `lwarpmk` option for the lwarp package:

   ```
   \usepackage[lwarpmk]{lwarp}
   ```

2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.

3. `lwarpmk.lua` may be used for this project:

   Unix:

   (a) Make `lwarpmk.lua` executable:

   Enter ⇒ `chmod 0755 lwarpmk.lua`

   (b) Compile documents with

   Enter ⇒ `./lwarpmk.lua html`

   Enter ⇒ `./lwarpmk.lua print`

   etc.

   (c) It may be useful to rename or link to a version without the `.lua` suffix.

   Windows:

   Compile documents with either of the following, depending on which command shell is being used:

   Enter ⇒ `texlua lwarpmk.lua html`

   Enter ⇒ `texlua lwarpmk.lua print`

   etc.

   Or:

   Enter ⇒ `lwarpmk html`

   Enter ⇒ `lwarpmk print`

   etc.

5.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 56.

Enter ⇒ `luatex --version`

Enter ⇒ `xindy --version`

Enter ⇒ `latexmk --version`

Enter ⇒ `perl --version`

Enter ⇒ `pdfcrop --version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate --version`

Enter ⇒ `pdftocairo -v`

To install `xindy`, `latexmk`, and `pdfcrop`:

The TEX utilities `xindy`, `latexmk`, and `pdfcrop` may be installed in TexLive with `tlmgr`, installed by MikTeX, provided by your operating system's package manager, or downloaded from the CTAN archive:

http://ctan.org/pkg/xindy
http://ctan.org/pkg/latexmk
http://ctan.org/pkg/pdfcrop

Prog `pdftotext`
Prog `pdfseparate`
Prog `pdftocairo`

To install the Poppler utilities to a Unix/Linux system:

The tools from the POPPLER project should be provided by your operating system's package manager.

To install the Poppler utilities to a MacOS machine:

1. Install Homebrew from https://brew.sh/:
   Enter ⇒
```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```
2. Install the Poppler utilities:
   Enter ⇒ `brew install poppler`

To install the Poppler utilities to a Windows machine:

1. See table 2 on page 56.
2. Download and extract the Poppler utilities `pdftotext`, `pdfseparate`, and `pdftocairo` to a directory, such as `Poppler`.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.
6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the `bin` directory of the Poppler utilities, such as:
   `C:\Users\<myname>\Desktop\Poppler\poppler-0.5_x86\poppler-0.5\bin`

   Be sure to include `\bin`.
9. Click "Ok" when done.

Prog `perl` **To install Perl to a Windows machine:**

1. Download and install a version of Perl, such as Straweberry Perl, to a directory without a space in its name, such as `C:\Strawberry`.
2. Edit the Path as seen above for the Poppler utilities.
3. Enter the `bin` directory of the Perl utility, such as:

   `C:\Strawberry\perl\bin`

   Be sure to include `\bin`.
4. Click "Ok" when done.

Any utilities installed by hand must be added to the `PATH`.

6 Tutorial

This section shows an example of how to create an lwarp document.

6.1 Starting a new project

1. Create a new project directory called `tutorial`.

File `tutorial.tex`

2. Inside the `tutorial` directory, create a new file called `tutorial.tex`. This may be done several ways:

   Copy from the documentation PDF:

   A listing is in fig. 1, which may be copied/pasted from the figure directly into your own editor, depending on the quality of the PDF viewer and editor, or:

   Copy from the lwarp documentation directory:

File `lwarp_tutorial.txt`

   Another copy may be found by entering into a command line:

   Enter ⇒ `texdoc -l lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

   This should be in the `doc/latex/lwarp/` directory along with this PDF documentation. Copy `lwarp_tutorial.txt` directly into your `tutorial` directory, renamed as `tutorial.tex`.

⚠ Bad formatting!

   When using Windows, use an editor other than Notepad, since Notepad does not accept the end-of-line from a Unix text file.

3. Compile the project:

   Enter ⇒ `pdflatex tutorial.tex`

   (several times)

   (`xelatex` or `lualatex` may be used as well.)

4. View the resulting `tutorial.pdf` with a PDF viewer.

A number of new files are created when `tutorial.tex` is compiled, as shown in table 3. These files are created by the `lwarp` package.

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created, the configuration files for `lwarpmk` are updated to record the operating system, LaTeX program (`pdflatex`, `xelatex`, or `lualatex`), the filenames of the source code and HTML output, and whether the additional helper program `latexmk` will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

```
% Save this as tutorial.tex for the lwarp package tutorial.

\documentclass{book}

\usepackage{iftex}

% --- LOAD FONT SELECTION AND ENCODING BEFORE LOADING LWARP ---

\ifPDFTeX
\usepackage{lmodern}            % pdflatex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\else
\usepackage{fontspec}           % XeLaTeX or LuaLaTeX
\fi

% --- LWARP IS LOADED NEXT ---
\usepackage[
%   HomeHTMLFilename=index,     % Filename of the homepage.
%   HTMLFilename={node-},       % Filename prefix of other pages.
%   IndexLanguage=english,      % Language for xindy index, glossary.
%   latexmk,                    % Use latexmk to compile.
%   OSWindows,                  % Force Windows. (Usually automatic.)
%   mathjax,                    % Use MathJax to display math.
]{lwarp}
% \boolfalse{FileSectionNames}  % If false, numbers the files.

% --- OTHER PACKAGES ARE LOADED AFTER LWARP ---
\usepackage{makeidx} \makeindex
\usepackage{xcolor}             % (Demonstration purposes only.)
\usepackage{hyperref,cleveref}  % LOAD THESE LAST!

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarp Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2}        % Include subsections in the \TOC.
\setcounter{secnumdepth}{2}     % Number down to subsections.
\setcounter{FileDepth}{1}       % Split \HTML\ files at sections
\booltrue{CombineHigherDepths}  % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1}    % Include subsections in the side\TOC
\HTMLTitle{Webpage Title}       % Overrides \title for the web page.
\HTMLAuthor{Some Author}        % Sets the HTML meta author tag.
\HTMLLanguage{en-US}            % Sets the HTML meta language.
\HTMLDescription{A description.}% Sets the HTML meta description.
```

```
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{lwarp_sagebrush.css}

\begin{document}

\maketitle                      % Or titlepage/titlingpage environment.

% An article abstract would go here.

\tableofcontents                % MUST BE BEFORE THE FIRST SECTION BREAK!
\listoffigures

\chapter{First chapter}

\section{A section}

This is some text which is indexed.\index{Some text.}

\subsection{A subsection}

See \cref{fig:withtext}.

\begin{figure}\begin{center}
\fbox{\textcolor{blue!50!green}{Text in a figure.}}
\caption{A figure with text\label{fig:withtext}}
\end{center}\end{figure}

\section{Some math}

Inline math: $r = r_0 + vt - \frac{1}{2}at^2$
followed by display math:
\begin{equation}
a^2 + b^2 = c^2
\end{equation}


\begin{warpprint}   % For print output ...
\cleardoublepage    % ... a common method to place index entry into TOC.
\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\end{warpprint}
\ForceHTMLPage      % HTML index will be on its own page.
\ForceHTMLTOC       % HTML index will have its own toc entry.
\printindex

\end{document}
```

Table 3: Files created along with the print version

`tutorial.pdf`: The PDF output from LaTeX. The print version of the document.

`tutorial_html.tex`: A small `.tex` file used to create a parallel HTML version of the document, which co-exists with usual the PDF version, and which will have its own auxiliary files. In this way, both PDF and HTML documents may co-exist side-by-side.

Auxiliary files: The usual LaTeX files `.aux`, `.log`, `.out`, `.toc`, `.lof`, `.idx`. When an HTML version of the document is created, `_html` versions of the auxiliary files will also be generated.

`lwarpmk.conf`: A configuration file for `lwarpmk`, which is used to automate the compilation of PDF or HTML versions of the document.

`tutorial.lwarpmkconf`: Another configuration file used by `lwarpmk`, which is only useful if you wish to have several projects residing in the same directory.

`.css files`: `lwarp.css`, `lwarp_formal.css`, `lwarp_sagebrush.css` These files are standard for lwarp, and are not meant to be modified by the user.

`sample_project.css`: An example of a user-customized css file, which may be used for project-specific changes to the lwarp defaults.

`lwarp.xdy`: Used by lwarp while creating an index. This file should not be modified by the user. A custom file may be used instead, if necessary.

`lwarp_mathjax.txt`: Inserted into the HTML files when `MathJax` is used to display math. This file should not be modified by the user.

`comment.cut`: A temporary file used by lwarp to conditionally process blocks of text. This file may be ignored.

When the `lwarpmk` option is given to the lwarp package:

`lwarpmk.lua`: A local copy of the `lwarpmk` utility.

On Unix-related operating systems this file must be made executable:

```
chmod u+x lwarpmk.lua
```

This may be useful to have to archive with a project for future use.

6.2 Compiling the print version with `lwarpmk`

The `lwarpmk` utility program is used to compile either the printed or the HTML version of the document.

`lwarpmk print` is used to recompile a printed version of the document.

1. Re-compile the print version:

   Enter ⇒ `lwarpmk print`

   `lwarpmk` prints an introduction then checks to see if the document must be recompiled. If it seems that the files are up-to-date, then `lwarpmk` informs you of that fact and then exits.

2. Make a small change in the original document, such as adding a space character.

3. Recompile again.

   Enter ⇒ `lwarpmk print`

   The document is recompiled when a change is seen in the source. Several compilations may be necessary to resolve cross-references.

4. Force a recompile to occur.

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk print`

   `lwarpmk again` updates the date code for the file, triggering a recompile the next time the document is made.[5]

5. Process the index.[6][7]

   Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

   Enter ⇒ `lwarpmk print`

7. To force a single recompile when needed, even if no changes were detected:

   Enter ⇒ `lwarpmk print1`

Note that the HTML customization commands are ignored while making the print version.

[5] Although, when using the utility `latexmk` (introduced later), the changed date is ignored and an actual change in contents must occur to cause a recompile.

[6] A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

[7] Also see section 8.4.9 for index options.

6.3 Compiling the HTML version with `lwarpmk`

`lwarpmk html` is used to recompile an HTML version of the document.

1. Compile the HTML version:

   Enter ⇒ `lwarpmk html`

   (a) `lwarpmk` uses LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.

   (b) `pdftotext` is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.

   (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`[8], `Some-math.html`, and the document's index in `_Index.html`.[9]

2. View the homepage in a web browser.

   Open the file `tutorial.html` in a web browser.

   math

   Note that math is still displayed as its plain-text LaTeX source until the images of the math expressions have been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 6.4 and 6.5.

3. Force a recompile:

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk html`

   Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile:[10][11]

   Enter ⇒ `lwarpmk htmlindex`

   Enter ⇒ `lwarpmk html`

   `_Index.html` is updated for the new LaTeX index.

5. Reload the web page to see the added index.

6. To force a single recompile when needed, even if no changes were detected:

   Enter ⇒ `lwarpmk html1`

[8] `First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

[9] `index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

[10] A `lwarpmk htmlglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

[11] Also see section 8.4.9 for index options.

6.4 Generating the SVG images

math as SVG images

By default lwarp represents math as SVG images with the LaTeX source included in `alt` attributes. In this way, the math displays as it was drawn by LaTeX, and the LaTeX source may be copied and pasted into some other document.

`picture` and Ti*k*z

lwarp uses the same mechanism for `picture` and Ti*k*z environments.

1. Create the SVG images:

   Enter ⇒ `lwarpmk limages`

   Enter ⇒ `lwarpmk html`

2. Move to the tutorial's math page and reload.
3. The math images are displayed using the same font and formatting as the printed version.
4. Copy/paste a math expression into a text editor to see the LaTeX source.

⚠ Adding/removing

When a math expression, `picture`, or Ti*k*z environment is added or removed, the SVG images must be re-created with `lwarpmk limages` to maintain the proper image file sequence numbers.

⚠ HTML instead of images

If HTML appears where an SVG image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time.

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as LaTeX would, but has the disadvantage of requiring an individual file for each math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time `$x$` is used, for example, yet another file is created. For a document with a large amount of math, see section 6.5 to use MathJax instead.

6.5 Using MathJax for math

math with MathJax

Math may also be represented using the MathJax Javascript project.

1. In the tutorial's source code, uncomment the `mathjax` package option for lwarp:

   ```
   mathjax, % Use MathJax to display math.
   ```

2. Recompile

   Enter ⇒ `lwarpmk html`

3. Reload the math page.

⚠ MathJax requirements

MathJax requires web access unless a local copy of MathJax is available, and it also requires that Javascript is enabled for the web page. The math is rendered by MathJax. Right-click on math to see several options for rendering, and for copying the LaTeX source.

While using MathJax has many advantages, it may not be able to represent complex expressions or spacing adjustments as well as LaTeX, and it may not support some math-related packages.

6.6 Changing the CSS style

For a formal css style, add to the preamble:

```
\usepackage{lwarp}
...
\CSSFilename{lwarp_formal.css}
...
\begin{document}
```

For a modern css style, `lwarp_sagebrush.css` is also provided:

```
\CSSFilename{lwarp_sagebrush.css}
```

See section 7.4 for more information about modifying the CSS styling of the document.

6.7 Customizing the HTML output

A number of settings may be made to control the HTML output, including filename generation, automatic compilation, math output, document splitting, meta data, and page headers and footers.

See section 7.3 for more information.

6.8 Using `latexmk`

`latexmk` is a LaTeX utility used to monitor changes in source files and recompile as needed.

1. In the tutorial's source code uncomment the `latexmk` option for the lwarp package:

   ```
   latexmk, % Use latexmk to compile.
   ```

2. Recompile the printed version of the document.

   Enter ⇒ `lwarpmk print`

   lwarp updates its own configuration files (`lwarpmk.conf` and `tutorial.lwarpmkconf`) whenever the printed version of the document is compiled. These configuration files remember that `lwarpmk` should use `latexmk` to compile the document.

3. Recompile the document.

   Enter ⇒ `lwarpmk print`

   and/or

   Enter ⇒ `lwarpmk html`

Changes are detected by comparing checksums rather than modification times, so `lwarpmk again` will not trigger a recompile, but `latexmk` has a much better awareness of changes than the `lwarpmk` utility does and it is likely to correctly know when to recompile. A recompile may be forced by making a small change to the source.

forced single-pass recompile

A single recompile may be forced with:

Enter ⇒ `lwarpmk print1`

and/or

Enter ⇒ `lwarpmk html1`

6.9 Using XeLaTeX or LuaLaTeX

XeLaTeX or LuaLaTeX may be used instead of LaTeX.

1. Remove the auxiliary files for the project:

   Enter ⇒ `lwarpmk cleanall`

2. Use `xelatex` or `lualatex` to recompile the printed version.

   Enter ⇒ `xelatex tutorial.tex`

   -or-

   Enter ⇒ `lualatex tutorial.tex`

   When the recompile occurs, the configuration files for `lwarpmk` are modified to remember which TeX engine was used. XeLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

3. To recompile the document:

   Enter ⇒ `lwarpmk print`

   -and-

   Enter ⇒ `lwarpmk html`

4. Also rememeber to update the indexes and recompile again.

6.10 Using a glossary

lwarp supports the glossaries package, although this tutorial does not supply an example.

Opt IndexLanguage

To assign a language to be used while processing the index and glossary, use the IndexLanguage option:

```
\usepackage[IndexLanguage=english]{lwarp}
```

To process the glossary for the print version:

Enter ⇒ `lwarpmk printglossary`

To process the glossary for the HTML version:

Enter ⇒ `lwarpmk htmlglossary`

In each case, the document will have to be recompiled afterwards.

6.11 Cleaning auxiliary files

To remove the auxiliary files `.aux`, `.toc`, `.lof`, `.lot`, `.idx`, `.ind`, `.log`, and `.gl*`:

Enter ⇒ `lwarpmk clean`

6.12 Cleaning auxiliary and output files

To remove the auxiliary files, and also remove the `.pdf` and `.html` files:

Enter ⇒ `lwarpmk cleanall`

6.13 Processing multiple projects in the same directory

It is possible to have several projects in the same directory. `lwarpmk` has an optional parameter which is the document to compile.

To create each project:

Enter ⇒ `pdflatex project_a`

Enter ⇒ `pdflatex project_b`

Each project is given its own configuration file:

`project_a.lwarpmkconf, project_b.lwarpmkconf`

To compile each project with `lwarkmk`:

Enter ⇒ `lwarpmk print project_a`

Enter ⇒ `lwarpmk html project_b`

6.14 Using the `make` utility

`lwarpmk` has an action which may be useful for integration with the common `make` utility:

`lwarpmk pdftohtml [project]`

`make` may be used to compile the code to PDF with HTML tags (`project_html.pdf`), then `lwarpmk` may be used to convert each target to HTML files.

7 Additional details

7.1 Font and UTF-8 support

lwarp uses `pdftotext` to convert PDF output into UTF-8-encoded text. This process requires that UTF-8 information be embedded in the PDF file, which usually prevents the use of older bit-mapped fonts.

pdflatex, T1, UTF8

While using `pdflatex`, fontspec is automatically loaded with T1 encoding, and inputenc is automatically loaded with UTF8 encoding, each of which is required for the conversion process. fontspec may be loaded with an additional encoding after lwarp.

vector fonts
Computer Modern

While using `pdflatex`, if no font-related package is specified, the default bit-mapped Computer Modern font is used, so simply add

```
usepackage{lmodern}
```

to the preamble to enable the related vector font instead, or use

```
\usepackage{dejavu}
```

or other other font packages, which may provide an increased coverage of Unicode mappings. Avoid bit-mapped fonts.

XeLaTeX and LuaLaTeX users must use the fontspec package. Do NOT use fontenc!

Place fontspec or fontenc and other font and UTF-8 related commands after the `\documentclass` command and before `\usepackage{lwarp}`:

1. `documentclass{article/book/report}` goes here, followed by any of:
2. Font and UTF-8 related commands:
   - For XeLaTeX or LuaLaTeX:
     - fontspec and font choices

       Pkg fontspec

       ligatures

       lwarp sets the following to turn off TeX ligatures during the generation of HTML tags, and turn off common ligatures in regular text, since older browsers may not display them correctly and newer browsers can automatically re-create them.

       ```
       \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
       \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
       \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
       ```
   - For `pdflatex`:
     - lmodern or other font-related packages

       Pkg lmodern

Pkg fontenc
Pkg inputenc
Pkg newunicodechar
File glyphtounicode

Pkg cmap
Pkg textcomp
Pkg microtype
ligatures

- fontenc
- inputenc
- newunicodechar
- `\input glyphtounicode.tex`
- `\input glyphtounicode-cmr.tex% from the pdfx package`
- `\pdfgentounicode=1`
- cmap
- textcomp
- microtype is automatically used by lwarp to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off TeX ligatures.

3. `\usepackage{lwarp}` (section 7.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

7.1.1 Indexes and UTF-8

lwarp uses the `xindy` program to processes indexes.

While using `xelatex` or `lualatex`, `xindy` is used for the index. Everything is handled in UTF-8 encoding, and should work as expected.

While using `pdflatex`, the `texindy` program is used with the `-C utf8` option, which is newly supported in recent distributions of LaTeX. This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. XeLaTeX or LuaLaTeX are recommended for non-Latin languages.

For an older distribution of LaTeX, it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the texindy call. See section 11.3.

Table 4: Package options

Option	Description
warppprint	Generate print output, and also generate configuration files.
warpHTML	Generate HTML output.
mathsvg	Show math using SVG images.
mathjax	Show math using MathJax.
OSWindows	Force compatibility with MS-Windows.
BaseJobname	The \jobname to use. Set to the \jobname of the printed version even while generating HTML.
HomeHTMLFilename	The filename of the home page.
HTMLFilename	A prefix for the filenames of the remaining web pages.
IndexLanguage	The xindy language option used for index and glossary generation.
latexmk	Boolean for lwarpmk to use latexmk for compiling documents.
lwarpmk	Generate a local copy of lwarpmk.lua.
xdyFilename	Set a custom filename for xindy.

7.2 lwarp package loading and options

lwarp supports book, report, and article classes, as well as the equivalent Koma-script classes and memoir.

Pkg lwarp Load the lwarp package immediately after the font and UTF-8 setup commands.

Opt warppprint
Opt warpHTML
Usually controlled by lwarpmk, and not set in the document. Select the warppprint option to generate print output (default), or the warpHTML option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual pdflatex, etc. When lwarp is loaded in print mode, it creates <project>_html.tex, which sets the warpHTML option before calling the user's source code <project>.tex. In this way, <project>.tex can \usepackage{lwarp} without any options to create a printed version, while <project>_html.tex will create an HTML version.

Opt mathsvg
Opt mathjax
For math display, select mathsvg (default), or mathjax. For more information about the math options, see section 8.5.

Opt `OSWindows` See section 7.5 if using Windows.

Opt `BaseJobname` Not intended for the user. Used internally by lwarp when creating the `*_html.tex` file used to compile the HTML version. See section 22.

Opt `HomeHTMLFilename` See section 7.3.

Opt `HTMLFilename` See section 7.3.

Opt `IndexLanguage` If using an index or glossary, see section 22.

Opt `latexmk` Has `lwarpmk` use `latexmk` to recompile the document several times if necessary. Otherwise, `lwarpmk` attempts to determing for itself whether to recompile. See section 7.3.

Opt `lwarpmk` If you wish to have lwarp generate a local copy of `lwarpmk.lua` for archival or local-installation purposes, compile the print version with the `lwarpmk` option set. See section 22.

Opt `xdyFilename` The default `xindy` filename is `lwarp.xdy`. If you wish to use a custom `.xdy` file for index generation, see section 22.

Table 5: HTML settings

Option	Description
SideTOCDepth	Sectioning depth of the sideTOC.
FileDepth	Sectioning depth of the file splits.
CombineHigherDepths	Combine higher section levels.
FileSectionNames	Use section names for file names, else use numbers.
FootnoteDepth	Sectioning depth of footnotes.
\abstractname	The name of the abstract.
\CSSFilename	The CSS for the following files.
\HTMLLanguage	The `html lang` tag.
\HTMLTitle	The HTML `title` meta tag, overriding `\title`.
\HTMLAuthor	The HTML `author` meta tag, overriding `\author`.
\HTMLDescription	The HTML `description` meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.

7.3 Customizing the HTML output

⚠ Placement!

Several settings may be used to customize the HTML output. Watch for the correct placement of each!

⚠ Changes!

Note that if changes are made, it is best to first:

1. Clear all the HTML, PDF, and auxiliary files:

   Enter ⇒ `lwarpmk clearall`

2. Recompile the print version in order to recreate the configuration files for `lwarpmk`:

   Enter ⇒ `lwarpmk print`

3. Finally, recompile the HTML version with the new settings:

   Enter ⇒ `lwarpmk html`

Options for the lwarp package:

Use the following as options for `\usepackage[<options>]{lwarp}`:

Opt `HomeHTMLFilename`
Default: `\BaseJobname`

`HomeHTMLFilename`: Filename of the homepage, without the "`.html`" suffix. Defaults to the `\BaseJobname`. A common setting is:

```
HomeHTMLFilename=index
```

causing the homepage to be the file `index.html`. Underscores are allowed in `HomeHTMLFilename` and `HTMLFilename` options, but may need to be escaped elsewhere, such as when appearing in a list:

filename underscores

```
\item [\href{file\_name.pdf}{text}] \
```

See section 7.3.1 for examples of naming and numbering HTML files.

Opt `HTMLFilename`
Default: `<empty>`

`HTMLFilename`: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty. See section 7.3.1 for examples of naming and numbering HTML files.

Opt `latexmk`
Default: `false`

`latexmk`: Controls whether lwarp uses `latexmk` to compile the document. This setting is written to `lwarpmk`'s configuration files.

Opt `mathsvg`
Default: `true`

`mathsvg`: Selects SVG display for math output. (The default.)

Opt `mathjax`
Default: `false`

`mathjax`: Selects MathJax for math output.

Placed in the preamble before `\begin{document}`:

Ctr `tocdepth`

`tocdepth`: Sectioning depth of the table of contents. See section 14 for a list of LaTeX stack depths.

Ctr `SideTOCDepth`
Default: 1

`SideTOCDepth`: Sectioning depth of the sideTOC. Defaults to 1, causing the sideTOC to show sections but not subsections.

sideTOC

Each subpage of the website has its own small table of contents on the side (the "sideTOC"). Its depth is set by `SideTOCDepth`. This sideTOC is only shown if the web page is wide enough. When using a narrow web browser window, "responsive web design" is used to show the sideTOC at the top of the page and a link back to "Home" at the bottom.

It is recommended to set:

```
SideTOCDepth = FileDepth
```

or

```
SideTOCDepth = FileDepth+1
```

⚠ inaccessible pages

If `SideTOCDepth < FileDepth`, web pages will be inaccessible via the sideTOC.

Ctr FileDepth
Default: -5

FileDepth: Sectioning depth of file splits. Defaults to -5, causing the entire HTML website to be one single file.

- To place the entire file into one HTML page, use:
  `\setcounter{FileDepth}{-5}`
- To split the HTML file at `\section` depth, use:
  `\setcounter{FileDepth}{1}`
- ⚠ To ensure that the HTML pages/files are accessible:
  Place a `\tableofcontents` somewhere before the first section break (therefore in the "home page"), and set
  `tocdepth >= FileDepth`

Bool CombineHigherDepths
Default: true

CombineHigherDepths: Combine a higher section with its first lower subsections, down to the `FileDepth`. Defaults to true. Set to false to simulate the concept of a chapter opening on its own page, for example.

The file splits are controlled by the counter `FileDepth` and the boolean `CombineHigherDepths`. Setting `FileDepth` to 0 splits the file at chapters, 1 at sections, etc. `CombineHigherDepths` controls whether to combine pages at levels higher than the chosen `FileDepth`, such as in this tutorial where the page which opens the chapter also contains the first section.

⚠ Inaccesible pages!

Be careful to set `tocdepth` and `SideTOCDepth` to allow access to each page of the website. Set `tocdepth` and `SideTOCDepth` to be greater than or equal to `FileDepth`.

⚠ Lost in an old page!

When making changes to the file structure, it is possible to end up with the web browser pointing to an old file which is no longer in use. When this occurs, changes to the web site will not appear in the browser, even if reloading the page, because that page is no longer in use. It is best to return to the home page, clean the files (`lwarpmk cleanall`), change `FileDepth` and/or `CombineHigherDepths`, then finally recompile and renavigate to the desired page using the new file structure.

Bool FileSectionNames
Default: true

FileSectionNames: If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the `HTMLFilename` option is used as a prefix. See section 7.3.1 for examples of naming and numbering HTML files.

Ctr FootnoteDepth
Default: 5

FootnoteDepth: Determines where to place pending footnotes. 5 places footnotes before each break down to the `\subparagraph` level. 1 places footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

\abstractname
Default: Abstract

\abstractname: The name of the abstract. This may also be over-written by the babel package. Defaults to "Abstract".

Placed before `\begin{document}`, or before any sectioning command which causes a file break:

`\CSSFilename`
Default: `lwarp.css`

`\CSSFilename`: {⟨*filename.css*⟩} Sets the CSS file to use for the following files. May be changed before each each sectioning command which would cause a file split.

The CSS styles of the web pages are set by the `\CSSFilename` command. If `\CSSFilename` is not used, a default plain style is used to mimic printed LaTeX output. `lwarp_sagebrush.css` is a semi-fancy colored style as shown in this tutorial. Change it to `lwarp_formal.css` for a more formal look, or comment out the `\CSSFilename` command to see the default. `\CSSFilename` may be used before each file break to set the CSS for individual pagess of the website.

`\HTMLLanguage`
Default: `en-US`

`\HTMLLanguage`: {⟨*langauge*⟩} The HTML file's `html lang` meta tag. Defaults to `en-US`.

`\HTMLTitle`
Default: `\thetitle`

`\HTMLTitle`: {⟨*title*⟩} Overrides `\title` for the HTML header's meta title. Defaults to `\thetitle`, which is set by `\title`, or empty otherwise. Unlike the author, `\thetitle` is set by `\title` even if not using the titling package.

`\HTMLAuthor`
Default: `\theauthor`

`\HTMLAuthor`: {⟨*author*⟩} The HTML header's meta author. Defaults to `\theauthor`, which is set by `\author` if using the titling package, but is empty otherwise. There are several ways to represent the author and affiliations, especially if using the authblk package, most of which do not result in a sensible `\theauthor`, so `\HTMLAuthor` is useful to create a list of authors without their affiliations.

`\HTMLDescription`
Default: `<empty>`

`\HTMLDescription`: {⟨*description*⟩} Sets the HTML `description` tag for the following files. May be changed before each each sectioning command which would cause a file split.

`\HTMLFirstPageTop`
Default: `<empty>`

`\HTMLFirstPageTop`: {⟨*contents*⟩} A user-definable custom action applied to the top of the home page. Useful for logos, etc. Defaults empty. Ignored in print output.

`\HTMLPageTop`
Default: `<empty>`

`\HTMLPageTop`: {⟨*contents*⟩} A user-definable custom action applied to the top of pages other than the home page. Useful for logos, etc. Defaults empty. `\LinkHome` may be used to place a link back to the homepage. Ignored in print output.

`\HTMLPageBottom`
Default: `<empty>`

`\HTMLPageBottom`: {⟨*contents*⟩} A user-definable custom action applied to the bottom of each web page. Useful for authors, copyright notices, contact information, etc. Defaults empty. `\LinkHome` may be used to place a link back to the homepage. Ignored in print output.

Placed in the home page before the first sectioning command which causes a file break:

\tableofcontents
⚠ TOC on the homepage!

\tableofcontents: Used to place a table of contents on the home page. This command must be used before the first file split, so that a way is available to navigate to other files from the homepage.

Links to each chapter/section are provided, as selected by `tocdepth`.

Placed in the document wherever necessary:

Env warpprint

warpprint: An environment which is only used while generating print output. Place here anything which does not apply to HTML and which may cause problems with lwarp. If lwarp knows about and emulates or supports a package then its related macros, lengths, counters, etc. probably won't have to be placed inside a `warpprint` environment, but unknown packages may cause problems which may be isolated from lwarp using this environment.

Env warpHTML

warpHTML: An environment which is only used while generating HTML output. This is useful for website logos and other items which have no purpose in printed output.

\warpprintonly

\warpprintonly: {⟨*contents*⟩} A macro version of the `warpprint` environment.

\warpHTMLonly

\warpHTMLonly: {⟨*contents*⟩} A macro version of the `warpHTML` environment.

7.3.1 Example HTML file naming

Examples of ways to name or number HTML files:

Numbered HTML nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`. [12]

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={node-}
]{lwarp}
\boolfalse{FileSectionNames}
```

[12] See `\SetHTMLFileNumber` to number in groups by chapter, for example.

Named HTML sections, no prefix:

Example: `index.html`, and `About.html`, `Products.html`

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={}
]{lwarp}
\booltrue{FileSectionNames}
```

Named HTML sections, with prefix:

Example: Homepage `mywebsite.html`, and additional pages such as `mywebsite-About.html`, `mywebsite-Products`, etc.

```
\usepackage[
    HomeHTMLFilename=mywebsite,
    HTMLFilename={mywebsite-}
]{lwarp}
\booltrue{FileSectionNames}
```

7.4 Customizing the CSS

`\CSSFilename`
Default: `lwarp.css`

`\CSSFilename` may be used to choose which `.css` file is used to display each page of the web site. Use `\CSSFilename` before `\begin{document}` to assign the style of the home page. If different parts of the website should have different styles, call `\CSSFilename` again before each section heading which creates a new file. This may be changed numerous times throughout the file, resulting in different HTML pages having different CSS files assigned:

```
...
\newCSS{myCSS.css}
\chapter{Another Chapter}
...
```

The styles provided by lwarp include:

`lwarp.css`: A default style if `\CSSFilename` is not used. This style is comparable to a plain LaTeX document. To set this style, you may use `\CSSFilename{lwarp.css}`, or no `\CSSFilename` call at all.

`lwarp_formal.css`: A formal style with a serif fonts and a traditional look.

lwarp_sagebrush.css: A style with muted colors, gradient backgrounds, additional borders, and rounded corners.

To see each style in use, change the \CSSFilename entry in the tutorial, `lwarpmk html` again, and then reload the tutorial webpage.

Custom css

A customized style may also be created. For each new project a file called `sample_project.css` is generated. This may be renamed to `<project>.css` then used by assigning `\CSSFilename{<project>.css}`.

⚠ Rename it!

Note that `sample_project.css` is overwritten whenever lwarp is loaded in print mode. It is therefore important to rename the file to something like `<project>.css` before using it, so that your own changes are not overwritten.

`<project>.css` has an entry which loads `lwarp.css`, and this entry may be changed to load `lwarp_formal.css` or `lwarp_sagebrush.css` if desired. Additional changes to the css may be made by making entries later in the `<project>.css` file.

File `lwarp.css`
File `project.css`
File `sample_project.css`

It is best to make a local project-specific css file such as `project.css`, containing only things which are different from `lwarp.css`. The file `project.css` should refer to `lwarp.css` as follows:

```
/* ( --- Start of project.css --- ) */
/* ( --- A sample project-specific CSS file for lwarp --- ) */

/* Load default lwarp settings: */
@import url("lwarp.css") ;
/* or lwarp_formal.css, lwarp_sagebrush.css */

/* Project-specific CSS setting follow here. */
/* . . . */

/* ( --- End of project.css --- ) */
```

Finally use `\CSSFilename{<project>.css}` in the document to activate the custom css.

7.5 Selecting the operating system

Prog `Unix`
Prog `Mac OS`
Prog `Linux`

lwarp tries to detect which operating system is being used. Unix / Mac OS / Linux is the default (collectively referred to as "Unix" in the configuration files), and MS-Windows is supported as well.

Prog `MS-Windows`
Prog `Windows`
Opt `OSWindows`

If Windows is not correctly detected, use the lwarp option `OSWindows`.

When detected or specified, the operating-system path separator used by lwarp is modified, the boolean `usingOSWindows` is set true. This boolean may be tested by the user for later use.

7.6 Selecting actions for print or HTML output

The following environments and macros are used to select actions which only apply to either traditional LaTeX print-formatted PDF generation, or to HTML generation.

For most of built-in LaTeX and many additional packages there is user-level source code support or emulation, so no special handling will be required. For those cases which lwarp does not handle by itself, the following environments and macros may be used to isolate sections of code for print-only or HTML-only.

These environments are also useful for creating a special version of the titlepage for print and another for HTML.

Env `warpHTML` Anything which is to be done only for HTML5 output is surrounded by a `warpHTML` environment:

```
\begin{warpHTML}
 ... something to be done only during HTML generation
\end{warpHTML}
```

Env `warpprint` Anything which is to be done only for print output is surrounded by a `warpprint` environment:

```
\begin{warpprint}
 ... something to be done only during traditional PDF generation
\end{warpprint}
```

Env `warpall` Anything which is to be done for any output may be surrounded by a `warpall` environment. Doing so is optional.

```
\begin{warpall}
 ... something to be done during print PDF or HTML output
\end{warpall}
```

Macros are also provided for print-only or HTML-only code:

`\warpprintonly` {⟨*actions*⟩}

Performs the given actions only when print output is being generated.

`\warpHTMLonly` {⟨*actions*⟩}

Performs the given actions only when HTML output is being generated.

7.7 Commands to be placed into the `warpprint` environment

Certain print-related commands should always be placed inside a `warpprint` environment, or may need other special handling. These are unrelated to HTML output, but are hard to isolate automatically. For example:

- Paragraph formatting: `\parindent` `\parskip`
- Manual page positions such as the textpos package, which is emulated but only in a limited way.

Some packages require additional setup commands. Where these packages are emulated for HTML, setup commands may work for the emulated HTML output as well as for print output. See the details for each package in this document for more information.

Also see section 12: Troubleshooting.

7.8 Title page

In the preamble, place an additional block of code to set the following:

```
\title{Document Title} % One line only
\author{Author One\affiliation{Affiliation One} \and
    Author Two\affiliation{Affiliation Two} }
\date{Optional date}
```

The title is used in the meta tags in the HTML files, unless overridden by `\HTMLTitle`, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 56.7.

`\maketitle` Use `\maketitle` just after the `\begin{document}`, as this will establish the title of the homepage. Optionally, use a `titlepage` environment instead.

Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The titlepage

will be set in a class titlepage, and \printtitle, etc. may be used inside this environment.

Env titlingpage Another form of custom title page, where \maketitle is allowed, and additional information may be included as well.

\title {⟨*title*⟩}

⚠ Avoid newlines in the \title; these will interfere with the file break and CSS detection. Use a \subtitle command instead (section 56.7). The title will appear in the document \maketitle as a heading <h1>. The HTML meta title tag will also have this title, unless \HTMLTitle is used to set the meta title to something else instead.

\author {⟨*author*⟩}

⚠ In \author, use \protect before formatting commands such as \textsc. In HTML, the author will appear in a of class author in the \maketitle. If the titling package is used, the author will also appear in a HTML meta tag, but \HTMLAuthor may be necessary to create a plain list of names if \author had affiliations added. \affiliation is a new addition to lwarp.

\date {⟨*date*⟩}

\date works as expected. In HTML, this will appear in a class titledate.

\thanks {⟨*text*⟩}

\thanks are allowed in the titlepage fields, and will be rendered as HTML notes at the bottom of the title page.

7.9 HTML page meta descriptions

\HTMLDescription {⟨*A description of the web page.*⟩} The default is no description.

Each page of HTML output should have its own HTML meta description, which usually shows up in web search results, is limited to around 150 characters in length, and should not include the ASCII double quote character (").

limitations

placement Use \HTMLDescription just before \begin{document} to set the description of the home page, and also just before each sectioning command such as \chapter or \section where a new file will be generated, depending on FileDepth. For example, if FileDepth is 1, use \HTMLDescription just before each \section command, and that description will be placed inside the HTML page for that \section. The same descrition will be used for all following HTML files as well, until reset by a new \HTMLDescription. It is best to use a unique description for each HTML file.

disabling To disable the generation of HTML description meta tags, use:

```
\HTMLDescription{}
```

7.10 HTML page meta title

\HTMLTitle {⟨*title*⟩} Sets the contents of the web page `<meta name="title">` element. Defaults to `\HTMLtitle{\thetitle}`. May be set empty to cancel the meta title tag.

7.11 HTML page meta author

\HTMLAuthor {⟨*author*⟩} Sets the contents of the web page `<meta name="author">` element. Defaults to `\HTMLAuthor{\theauthor}`. May be set empty to cancel the meta author tag.

`\author` may be used to create a list of authors and their affiliations, in several formats if using authblk, and these may not successfully parse properly into a sensible list for `\theauthor`. `\HTMLAuthor` may be used to set the meta tag to a simple list of names.

7.12 Modifying `xindy` index processing

Prog xindy
File lwarp.xdy

lwarpmk uses the file `lwarp.xdy` to process the index. This file is over-written by lwarp whenever a print version of the document is processed.

To customize index processing:

1. Copy `lwarp.xdy` to a new filename such as `projectname.xdy`
2. Make changes to `projectname.xdy`. Keep the line which says

   ```
   (markup-locref :open "\hyperindexref{" :close "}")
   ```

   This line creates the hyperlinks for the HTML index. During print output `\hyperindexref` becomes a null function.

Opt xdyFilename

3. In the document source use the `xdyFilename` option for lwarp:

   ```
   \usepackage[
       ... other options ...
       xdyFilename=projectname.xdy,
   ]{lwarp}
   ```

4. Recompile the print version, which causes lwarp to rewrite the `lwarpmk.conf` configuration file. This tells `lwarpmk` to use the custom `projectname.xdy` file instead of `lwarp.xdy`.

8 Special cases and limitations

Also see section 12: Troubleshooting.

Some commonly-used LaTeX expressions should be modified to allow for a smooth conversion to both HTML and print-formatted outputs:

8.1 Formatting

8.1.1 Text formatting

⚠ `\bfseries`, etc. `\textbf`, etc. are supported, but `\bfseries`, etc. are not yet supported.

⚠ HTML special chars `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

8.1.2 Horizontal space

`\hspace` `\hspace` is converted to an inline HTML span of the given width, except that `0` width is ignored, a width of `.16667em` is converted to an HTML thin breakable space (`U+2009`), and a `\fill` is converted to a `\qquad`.

`\,` `~` and `\,` are converted to HTML entities.

`\kern` `\hskip` `\kern` and `\hskip` are treated as a single normal space.

8.1.3 Text alignment

Use the environments `center`, `flushright`, `flushleft` instead of the macros `\centering`, `\raggedright`, `\raggedleft`.

8.1.4 Accents

Native LaTeX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

8.1.5 Textcomp

Pkg `textcomp` Some textcomp symbols do not have Unicode equivalents, and thus are not supported.

⚠ Missing symbols Many textcomp symbols are not supported by many fonts. Try using more complete fonts in the CSS, but expect to see gaps in coverage.

8.1.6 Superscripts and other non-math uses of math mode

Use `\textsuperscript{x}` instead of `$^{x}$`

8.1.7 Empty `\item` followed by a new line of text or a nested list:

Use a trailing backslash: `\item[label] \`

8.1.8 Filenames and URLs in lists or footnotes

filename underscore Escape underscores in the filenames:

```
\item[\href{file\_name.pdf}{text}]
```

8.1.9 relsize package

Pkg `relsize` For HTML only the inline macros are supported: `\textlarger`, `\textsmaller`, and `\textscale`. Each becomes an inline span of a modified `font-size`.

`\relsize`, `\larger`, `\smaller`, and `\relscale` are ignored.

While creating SVG math for HTML, the original definitions are temporarilty restored, and so should work as expected.

⚠ not small The HTML browser's setting for minumum font size may limit how small the output will be displayed.

8.2 Boxes and minipages

8.2.1 Marginpars

\marginpar [⟨*left*⟩] {⟨*right*⟩} \marginpar may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the \marginpar. Paragraph breaks are converted to <br /> tags.

\marginparBlock [⟨*left*⟩] {⟨*right*⟩} To include block-related macros, use \marginparBlock, which takes the same arguments but creates a instead of a . A line break will occur in the text where the \marginBlock occurs.

8.2.2 Save Boxes

LaTeX boxes are placed inline and do not allow line breaks, so boxes with long contents may overflow the line during HTML conversion. This is mostly a problem when the boxes contain objects which themselves hold large HTML tags, such as rotation commands with long contents. When this object overflows the line, some HTML code will be lost and the page will be corrupted.

8.2.3 Minipages

⚠ inline A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made inline-block.

placement Minipages and parboxes will be placed side-by-side in HTML unless you place a \newline between them.

side-by-side Side-by-side minipages may be separated by \quad, \qquad, \enskip, \hspace, \hfill, or a \rule. When inside a center environment, the result is similar in print and HTML. Paragraph tags are surpressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span There is limited support for minipages inside an HTML . An HTML cannot appear inside a . While in a , minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an "inline" format, without markup except for HTML breaks. Use \newline or \par for an HTML break.

size When using \linewidth, \textwidth, and \textheight, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages A minipage of width exactly \linewidth is automatically given no HTML width.

full-width minipages

A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML `width` attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment

Nested minipages adopt their parent's text alignment in HTML, whereas in regular LaTeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

8.2.4 Side-by-side minipages

Place side-by-side minipages inside a `center` environment, with horizontal space between them, such as `\quad`, `\qquad`, `\hspace`, or `\hfill`. The result is similar in print and HTML. Do not use space commands at the start or end of the line.

8.2.5 Framed minipages and other environments

`\fbox` can only be used around inline `<span>` items during HTML output, but HTML cannot place a block element such as a `<div>` for a minipage or a list inside of a `<span>`. Several options are provided for framing an object, depending on which kind of object and which packages are loaded:

`\fbox`
`\fboxBlock`
Env `fminipage`

For a framed object, options include:

To remove the frame in HTML output: Place the `\fbox` command and its closing brace inside `warpprint` environments. This will nullify the frame for HTML output.

For inline text:

To frame the contents inline with some formatting losses in HTML: This is the default action of `\fbox` when enclosing a minipage. During HTML output, `\fbox` nullifies the HTML tags for `minipage`, `\parbox`, and lists. The contents are included as inline text inside the `\fbox`'s `<span>` of class `framebox`. For lists, line breaks are converted to HTML breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra HTML markup.

For inline `minipage` and lists:

To frame the contents on their own line with improved formatting in HTML: A new command `\fboxBlock` is included, intended to be a direct replacement for `\fbox` for cases where the `\fbox` surrounds a minipage, table, or list. For print output, this behaves as `\fbox`. For HTML output, the contents are placed inside an HTML `<div>` with the class `framed`, resulting in the contents being placed on their own line with a frame surrounding them. The contents preserve their HTML formatting, so lists and minipages look nicer, and valid HTML is created for a `tabular`. While an `\fbox` containing a `tabular` is valid LaTeX code, the result in HTML is problematic since a table is a `<div>` not a `<span>`, so use

\fboxBlock around a tabular, or else place the tabular inside a minipage, or use fminipage, described next. Also see below regarding the "Misplaced alignment tab character &" error.

For display tabular, minipages, and lists:

To create a framed minipage in both print and HTML: A new environment fminipage is included. For print output, this is identical to minipage, except that it is also framed. For HTML output, this forms a of class framed, the contents preserve their HTML formatting, and valid HTML is created for a tabular. Also see below regarding the "Misplaced alignment tab character &" error.

colored boxes and frames:

To create colored frames and boxes: See section 294 for xcolor's \colorbox and \fcolorbox, and lwarp's additional \colorboxBlock and \fcolorboxBlock.

⚠ Misplaced alignment tab character &

To frame tables or verbatim environments: Place the contents inside a fminipage, or perhaps a \fboxBlock for a tabular. Also, if using \fboxblock with tabular, you will have to use \StartDefiningTabulars before the start of the macro which uses \fboxBlock and the tabular, and \EndDefiningTabulars afterwards. Also see the lwarp documentation for the fancybox package.

To frame equations: See section 142 for the fancybox package.

For fancy framed minipages: See packages boxedminipage, shadow, fancybox, framed, mdframed.

Custom environments: Use a custom environment to create a sidebar, containing a BlockClass environment with custom css formatting, and \warpprintonly{\hrule} command:

```
\begin{BlockClass}{frameminipage}% ignored in print output
  % use CSS to format div class ``framedminipage''
\warpprintonly{\hrule} % only appears in print output
Contents
\warpprintonly{\hrule} % only appears in print output
\end{BlockClass}
```

8.2.6 fancybox package

Pkg fancybox

framed equation example

fancybox's documentation has an example FramedEqn environment which combines math, \Sbox, a minipage, and an \fbox. This combination requires that the entire environment be enclosed inside a lateximage, which is done by adding \lateximage at the very start of FramedEqn's beginning code, and \endlateximage at the very end of the ending code. Unfortunately, the HTML alt attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives

`\fbox` works with fancybox. Also see lwarp's `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

framed table example

The fancybox documentation's example framed table using an `\fbox` containing a `tabular` does not work with lwarp, but the `FramedTable` environment does work if `\fbox` is replaced by `\fboxBlock`. This method loses HTML formatting. A better method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim

lwarp does not support the `verbatim` environment inside a span, box, or fancybox's `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The fancybox documentation's example `FramedVerb` may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed `\VerbBox`

fancybox's `\VerbBox` may be used inside `\fbox`.

indented alignment

`LVerbatim`, `\LVerbatimInput`, and `\LUseVerbatim` indent with horizontal space which may not line up exactly with what `pdftotext` detects. Some lines may be off slightly in their left edge.

8.2.7 mdframed package

Pkg `mdframed` support

Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for mdframed environments and frame titles.

⚠ loading

When used, lwarp loads mdframed in HTML with `framemethod=none`.

font

For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarp does not support `\bfseries` and friends, only one font selection may be made at a time.

theoremtitlefont

`theoremtitlefont` is not supported, since the following text is not in braces in the mdframed source.

footnotes

Footnotes are currently placed at the bottom of the HTML page.

ignored options

`userdefinedwidth` and `align` are currently ignored.

8.3 Cross-references

labels
⚠ underscores

Labels with special characters may be a problem. It is best to stick with alphanumeric, hyphen, underscore, and perhaps the colon (if not French).

`\nameref`
⚠ empty link

`\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

8.3.1 Page references

⚠ LaTeX page numbers

The printed page does not translate to the HTML page, so `\pageref` references are converted to parentheses containing `\pagerefPageFor`, which defaults to "see ", followed by a hyperlink to the appropriate object.

Ex:

```
\ref{sec:name} on page \pageref{sec:name}
```
in HTML becomes:
"Sec. 1.23 on page (**see** sec. 1.23)".

`\pagerefPageFor` may be redefined to "page for ", empty, etc. See page 360.

8.3.2 cleveref and varioref packages

Pkg `cleveref`
Pkg `varioref`
⚠ cleveref page numbers

cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

Ex:

```
\cpageref{tab:first,tab:second}
```
in HTML becomes:
"pages **for** table 4.1 and **for** table 4.2"

See `\cpagerefFor` at page 419 to redefine the message which is printed for page number references.

8.3.3 Hyperlinks, hyperref, and url

Pkg `hyperref`
Pkg `url`

lwarp emulates hyperref, including the creation of active hyperlinks, but does not require that hyperref be loaded by the document.

lwarp can also load url, but url should not be used at the same time as hyperref, since they both define the `\url` command. lwarp does not (yet) attempt to convert url links into hyperlinks during HTML output, nor does url create hyperlinks during print output.

⚠ backref

When generating HTML, lwarp's emulation of hyperref does not automatically load backref, so backref must be loaded explicitly.

8.3.4 Footnotes and page notes

lwarp uses native LaTeX footnote code, although with its own `\box` to avoid the LaTeX output routine. The usual functions work as-is.

⚠ pfnote numbers

While emulating pfnote, lwarp is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. lwarp therefore uses continuous footnote numbering even for pfnote.

8.4 Front and back matter

8.4.1 Starred chapters and sections

The following describes \ForceHTMLPage and \ForceHTMLTOC, which may be used for endnotes, glossaries, tocbibind, and the index. See the following sections where applicable. Continue here if interested in the reason for adding these commands to lwarp.

Some packages use \chapter* or \section* to introduce reference material such as notes or lists, often to be placed in the back matter of a book. These starred sections are placed inline instead of on their own HTML pages, and they are not given TOC entries.

lwarp provides a method to cause a starred section to be on its own HTML page, subject to FileDepth, and also a method to cause the starred section to have its own TOC entry during HTML output.

\ForceHTMLPage To place a starred section on its own HTML page, use \ForceHTMLPage just before the \chapter* or \section*. lwarp will create a new page for the starred sectional unit.

A starred sectional unit does not have a TOC entry unless one is placed manually. The typical method using \phantomsection and \addcontentsline works for inline text but fails when the new starred section is given its own webpage after the TOC entry is created. If the starred section has its own HTML page but no correct TOC entry pointing to that page, the page will be inaccessible unless some other link is created.

⚠ inaccessible HTML page

\ForceHTMLTOC To automatically force the HTML version of the document to have a TOC entry for a starred section, use \ForceHTMLTOC just before the \chapter* or \section*. The TOC will only be assigned for HTML output, not for print output, and it will appear in the main TOC and also the sideTOC per page.

For print output, \ForceHTMLTOC and \ForceHTMLPage have no effect.

8.4.2 abstract package

Pkg abstract

If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a file split, which would put a table of contents out of the home page if it is after the abstract.

⚠ missing TOC

8.4.3 titling and authblk

Pkg titling
Pkg authblk
package support
⚠ load order
\published and \subtitle

lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 56.7.

8.4.4 tocloft package

Opt tocloft titles
Pkg tocloft
⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its titles option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

8.4.5 appendix package

Pkg appendix
⚠ incorrect TOC link

During HTML conversion, the option toc without the option page results in a TOC link to whichever section was before the appendices environment. It is recommended to use both toc and also page at the same time.

8.4.6 pagenote package

Pkg pagenote

pagenote works as-is, but the page option is disabled.

8.4.7 endnotes package

Pkg endnotes
table of contents

To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\enoteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page

To additionally have the endnotes on their own HTML page, if FileDepth allows:

```
\ForceHTMLPage
\theendnotes
```

8.4.8 glossaries package

Pkg glossaries

xindy is required for glossaries.

The default `style=item` option for glossaries conflicts with lwarp, so the style is forced to `index` instead.

The page number list in the printed form would become `\namerefs` in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

placement and TOC options

The glossaries may be placed in a numbered or unnumbered section, given a TOC entry, and placed inline or on their own HTML page:

Numbered section, on its own HTML page:

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

Unnumbered section, inline with the current HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

Unnumbered section, on its own HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

Opt `IndexLanguage`

The lwarp package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.

Opt `lwarpmk printglossary`
Opt `lwarpmk htmlglossary`

`lwarpmk` has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary` to process the glossaries created by glossaries using `xindy`.

8.4.9 Index and the tocbibind package

Pkg `makeidx`
Pkg `tocbibind`
Opt `IndexLanguage`

The lwarp package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.

⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

placement and TOC options

An index may be placed inline with other HTML text, or on its own HTML page:

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

On its own HTML page, with a manual TOC entry:

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

Inline, with an automatic TOC entry:

Pkg tocbibind

The tocbibind package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

On its own HTML page, with an automatic TOC entry:

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt tocbibind numindex

numbered index section

Use the tocbibind numindex option to generate a numbered index. Without this option, the index heading has no number.

See section 66 for lwarp's core index and glossary code, and section 279 for tocbibind.

8.5 Math

8.5.1 Rendering tradeoffs

Math rendering

Math may be rendered as SVG graphics or using the MathJax JavaScript display engine.

SVG files In its current implementation, rendering math as images creates a new SVG file for each expression. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and check-summing may be used to remove the need for duplicate files.

SVG inline Another approach could be to in-line the SVG files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

8.5.2 SVG option

SVG math option For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[13], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

SVG math copy/paste For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the LaTeX code for the math expression. For $\mathcal{AMS}$ environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred $\mathcal{AMS}$ environment, although the `\tag` macro will still appear inside the LaTeX math expression.

8.5.3 MathJax option

MathJax math option The popular MathJax alternative (`mathjax.org`) may be used to display math.

Prog `MathJax`

[13] See section 295 regarding fonts and fractions.

When MathJax is enabled, math is rendered twice:

1. As regular LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of LaTeX, and
2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

MathJax limitations

Prog `MathJax`

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike LaTeX. `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

  https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY

subequations

- MathJax itself does not support subequations. This may be improved by parsing the LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a lateximage, and therefore also inside a Ti*k*z or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

⚠ siunitx inside an equation

- Usage of siunitx inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

  https://github.com/burnpanck/MathJax-siunitx

  Also see section 8.5.5.

tabbing

- A `tabbing` environment is emulated using an HTML `<pre>`.While MathJax is enabled inside `tabbing`, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, bigdelim, units, and nicefrac, along with occasionally-used macros such as `\footnote` and `\relax`.

custom MathJax macros

- MathJax does not automatically support custom LaTeX macros, but they may be created by the user inside a math expression:

```
\begin{document}
(...)
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
    % New macros for MathJax are
    % placed inside a math expression:
    \(
        \newcommand{\expval}[1]{\langle#1\rangle}
        \newcommand{\abs}[1]{\lvert#1\rvert}
    \)
}{}
\end{warpHTML}
```

8.5.4 ntheorem package

Pkg ntheorem

⚠ Font control

This conversion is not total. Font control is via css, and the custom LaTeX font settings are ignored.

⚠ Equation numbering

ntheorem has a bug with equation numbering in $\mathcal{AMS}$ environments when the option `thref` is used. lwarp does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarp's HTML output, but not with the print output. It is recommended to use cleveref instead of ntheorem's `thref` option.

8.5.5 siunitx package

Pkg siunitx

⚠ per-mode

Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ math mode required

Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the siunitx extension for MathJax is not currently hosted at any public CDN, thus siunitx is not usable with MathJax unless a local copy of this extension is created first.

8.5.6 units and nicefrac packages

Pkg units

Pkg nicefrac

units and nicefrac work as-is with lwarp, but MathJax does not have an extension for units or nicefrac. These packages do work with lwarp's option `svgmath`.

8.5.7 newtxmath package

Pkg newtxmath

⚠ loading sequence

The proper load order is:

```
...
\usepackage{lwarp}
...
\usepackage{amsthm}
\usepackage{newtxmath}
...
```

8.6 Graphics

Pkg graphics

Pkg graphicx

⚠ .pdf image files

⚠ no file extension

For `\includegraphics` with `.pdf` files, the user should provide a `.pdf` image file, and also a `.svg`, `.png`, or `.jpg` version of the same image. These should be referred to without a file extension:

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

For print output, lwarp will automatically choose the `.pdf` if available, other some other format otherwise. For HTML, one of the other formats is used instead.

Prog pdftocairo

To convert a PDF image to SVG, use the utility `pdftocairo`:

```
Enter ⇒   pdftocairo -svg filename.pdf
```

If a `.pdf` file is referred to with its file extension, a link to the `.pdf` file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

Pkg epstopdf

For `.eps` files, use epstopdf to provide a PDF version, and also provide a SVG version as well.

other image files

For `.png`, `.jpg`, or `.gif` image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

⚠ graphics vs. graphicx

⚠ viewports

If using the older graphics syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer graphicx syntax. Note that viewports are not supported by warp; the entire image will be shown.

units

For `\includegraphics`, avoid `px` and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use `ex` or `em`. For fixed-sized images, use `cm`, `mm`, `in`, `pt`, or `pc`. Use the keys `width=.5\linewidth`, or

similar for \textwidth or \textheight to give fixed-sized images proportional to a 6 by 9 inch text area.

options

\includegraphics accepts width and height, origin, rotate and scale, plus a new class key.

HTML class

With HTML output, \includegraphics accepts an optional class=xyz keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

\rotatebox

\rotatebox accepts the optional origin key.

⚠ browser support

\rotatebox, \scalebox, and \reflectbox depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

8.6.1 grffile package

Pkg grffile

⚠ matching PDF and SVG

grffile is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

8.6.2 color package

Pkg color

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

⚠ missing colors

It should be sufficient for the user's document to load color then load xcolor as well.

8.6.3 xcolor package

Pkg xcolor

\colorboxBlock and \fcolorboxBlock

\colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility, and they are identical to \colorbox and \fcolorbox in print mode. In HTML mode they place their contents into a instead of a . These s are set to display: inline-block so adjacent \colorboxBlocks appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for \colorboxBlock and \fcolorboxBlock are created by lwarp's core if xcolor is loaded.

background: none

\fcolorbox and \fcolorboxBlock allow a background color of none, in which case only the frame is drawn, which can be useful for HTML.

color support

Color definitions, models, and mixing are fully supported without any changes re-

quired.

tables Colored tables are ignored so far. Use CSS to style tables.

colored text and boxes `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

`\color` and `\pagecolor` `\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

8.6.4 epstopdf package

Pkg `epstopdf` When using epstopdf to convert images to PDF, use the `pdftocairo` utility to also provide an SVG version as well. In the document, refer to the image filename without a suffix. The PDF version will be used in print output, and the SVG version will be used for HTML.

8.6.5 overpic package

Pkg `overpic` ⚠ scaling The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

8.7 Tabbing

The `tabbing` environment works, except that SVG math and `lateximages` do not yet work inside the environment.

8.8 Tabular

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, siunitx `S` columns, or the packages multirow, longtable, supertabular, or xtab.

Defining environments:

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include `tabular` and instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands. These are are ignored in print mode.

```
\StartDefiningTabulars
<define macros or environments using tabular and &
here>
\EndDefiningTabulars
```

⚠ floatrow

This includes before and after defining any macro which used \ttabbox from floatrow.

⚠ tabular inside another environment

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

Cell contents:

⚠ paragraphs

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.

⚠ \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
... &  \multirow{2}{.5in}{text}  & ...
... &  \mrowcell                 & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 208.2 for \multicolumnrow.

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

⚠ empty cells

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}  & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use \TabularMacro just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:

⚠ * column specification

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

\multirow

- In \multirow cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the \multirow which do not appear in the HTML version.

⚠ \newcolumntype

- \newcolumntype is ignored; unknown column types are set to l.

Rules:

vertical rules

- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.

full-width rules

- \toprule, \midrule, \bottomrule, and \hline ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use \cmidrule followed by \bottomrule, it may be necessary to use:

```
\cmidrule{2-3} \\[-2ex]
\bottomrule
```

The optional -2ex is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly misplaced \noalign

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a "misplaced \noalign" error occurs, change

  ```
  This & That \endhead
  ```

  to

  ```
  \warpprintonly{This & That \endhead}
  ```

  and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevent to HTML output.

Other:

- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

longtable headings

- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}.

⚠ S columns

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarp's tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

  ```
  \warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
  \warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
  ```

8.8.1 longtable package

Pkg longtable ⚠

Longtable \endhead, \endfoot, and \endlastfoot rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid "Misplaced \noalign." when using \begin{warpprint}.

Keep the \endfirsthead row, which is still relevent to HTML output.

⚠ \kill is ignored, place a \kill line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside \warpingprintonly.

8.8.2 supertabular and xtab packages

Pkg supertabular
Pkg xtab
⚠ misplaced alignment
alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.8.

8.8.3 bigdelim package

Pkg bigdelim
⚠ use \mrowcell

\ldelim and \rdelim use \multirow, so \mrowcell must be used in the proper number of empty cells in the same column below \ldelim or \rdelim, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

	a	b
left {	c	d
	e	f
	g	h

8.9 Floats

8.9.1 float, trivfloat, and/or algorithmicx together

Pkg float
Pkg trivfloat
Pkg algorithmx
⚠ package conflicts

If using \newfloat, trivfloat, and/or algorithmicx together, see section 285.1.

8.9.2 caption and subcaption packages

Pkg caption
Pkg subcaption

To ensure proper float numbering, set caption positions such as:

```
\captionsetup[table]{position=top}
\captionsetup[figure]{position=bottom}
```

Similarly for `subtable`, `subfigure`, and `longtable`.

8.9.3 subfig package

Pkg `subfig`

⚠ `lof/lotdepth`

At present, the package options for `lofdepth` and `lotdepth` are not working. These counters must be set separately after the package has been loaded.

horizontal spacing

In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

8.9.4 floatrow package

Pkg `floatrow`

⚠ misplaced alignment

alignment tab character &

⚠ subfig package

Use `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 8.8.

When combined with the subfig package, while inside a `subfloatrow` `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.

⚠ `\FBwidth, \FBheight`

The emulation of floatrow does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:

- Use fixed lengths. lwarp will scale the HTML lengths appropriately.
- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

8.9.5 keyfloat package

Pkg `keyfloat`

⚠ `keywrap`

If placing a `\keyfig`[H] inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

8.10 Koma-Script

Cls `komascript`

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

⚠ Not fully tested! Please send bug reports!

Some features have not yet been tested. Please contact the author with any bug reports.

8.11 Memoir

Cls `memoir`

⚠ options clash

While emulating memoir, lwarp pre-loads a number of packages (section 301.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading lwarp:

```
\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}
```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

lwarp loads pagenote to perform memoir's pagenote functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the memoir enhancements. It is currently recommended to load and use fancyvrb instead.

The memoir glossary system is not yet supported by `lwarpmk`. The glossaries package may be used instead, but does require the glossary entries be changed from the memoir syntax to the glossaries syntax.

8.12 Miscellaneous

8.12.1 verse and memoir

Pkg verse
Cls memoir
\attrib

The documentation for the verse and memoir packages suggest defining an \attrib command, which may already exist in current documents, but it will only work for print output. lwarp provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len \leftskip
Len \leftmargini
Len \TMLvleftskip
Len \TMLleftmargini

These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths \HTMLvleftskip and \HTMLleftmargini are provided to control the margins in HTML output. These new lengths may be set by the user before any verse environment, and persist until they are manually changed again. One reason to change \HTMLleftmargini is if there is a wide \flagverse in use, such as the word "Chorus", in which case the value of \HTMLleftmargini should be set to a wide enough length to contain "Chorus". The default is wide enough for a stanza number.

⚠ spacing

Horizontal spacing relies on pdftotext's ability to discern the layout (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargini or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

8.12.2 newclude package

Pkg newclude
⚠ loading

newclude modifies \label in a non-adaptive way, so newclude must be loaded before lwarp is loaded:

```
\documentclass{article}
…<font setup>
\usepackage{newclude}
\usepackage[warpHTML]{lwarp}
…
```

8.12.3 babel package

Pkg babel

\CaptionSeparator

When French is used, the caption separator is changed to a dash. The following may be used to restore it to a colon:

```
\renewcommand*{\CaptionSeparator}{:~}
```

punctuation spaces

⚠ customized spacing

Also when French is used, lwarp creates fixed-width space around punctuation by patching \FBcolonspace, \FBthinspace, \FBguillspace, \FBmedkern, \FBthickkern, \FBtextellipsis, and the tilde. If the user's document also changes these parameters, the user's changes should be placed inside a warpprint environment so that the user's changes do not affect the HTML output.

8.12.4 todonotes and luatodonotes packages

Pkg todonotes

Pkg luatodonotes

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

8.12.5 fixme

Pkg fixme

⚠ external layouts

External layouts (\fxloadlayouts) are not supported.

User control is provided for setting the HTML styling of the "faces". The defaults are as follows, and may be changed in the preamble after fixme is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

8.12.6 xparse

Pkg xparse

To remove from the log any warnings about redeclaring objects, place the following before lwarp is loaded:

```
\usepackage[log-declarations=false]{xparse}
```

9 EPUB conversion

lwarp does not produce EPUB documents, but it may be told to modify its HTML output to greatly assist in the conversion. An external program may then be used to finish the conversion to EPUB.

<meta> author

To assign the author's name for regular lwarp HTML files, and also for the EPUB, use `\HTMLAuthor {⟨name⟩}`. This assigns the name to the `<meta>` author element. It may be set empty, and it defaults to `\theauthor`.

A special boolean is provided to simplify the process of converting lwarp HTML output to EPUB:

FormatEPUB

Bool FormatEPUB
Default: false

FormatEPUB changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

to the project's source preamble after `\usepackage{lwarp}`. The EPUB version of the document cannot co-exist with the regular HTML version, so

Enter ⇒ `lwarpmk cleanall`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk limages`

to recompile with the FormatEPUB boolean turned on. Several changes are then made to the HTML output:

- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.

Calibre

The resulting files will be ready to be loaded into an EPUB conversion program, such as the open-source program Calibre (https://calibre-ebook.com/).

⚠ search order

The EPUB conversion program must know what order the files are included. For lwarp projects, set the EPUB conversion software to do a breadth-first search of the files. For Calibre, this option is found in

Preferences → Plugins → File type plugins → HTML to Zip

Check the box Add linked files in breadth first order.

⚠ section breaks

The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 7. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`, whereas a `book` class document would break at `\chapter`, which is HTML heading level `<h3>`. For Calibre, this option is found in

> Preferences → Conversion (Common Options) → Structure Detection → Detect chapters at (XPath expression)

Select the "magic wand" to the right of this entry box, and set the first entry

> Match HTML tags with tag name:

to "h4". (Or "h3" for document classes with `\chapters`.) The Detect chapters at field should then show

> //h:h4 — or — //h:h3

This option is also available on the main tool bar at the Convert books button.

Once these settings have been made, the lwarp-generated HTML files may be loaded by Calibre, and then converted to an EPUB.

MathJax support

MathJax may be used in EPUB documents. Some e-readers include MathJax, but any given reader may or may not have a recent version, and may or may not include extensions such as support for siunitx.

lwarp adds some modifications to MathML to support equations numbered by chapter. These modifications may not be compatible with the e-reader's version of MathJax, so lwarp requests that a known version be loaded instead. In some cases chapter numbering of equations still doesn't work.

Until math support in EPUB documents is improved, it is recommended to use SVG images instead of MathJax, especially for equations numbered by chapter, or where siunitx support is important.

10 Word-processor conversion

lwarp may be told to modify its HTML output to make it easier to import the HTML document into a word processor. At the time of this writing, it seems that LibreOffice works best at preserving table layout, but it still has some limitations, such as an inability to automatically assign figure and table frames and captions according to user-selected HTML classes. lwarp provides some assistance in locating these frame boundaries, as shown below.

10.1 Activating word-processor conversion

A special boolean is provided to simplify the process of converting lwarp HTML output to EPUB:

FormatWP

Bool `FormatWP`
Default: `false`

Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Additionally, honors the booleans `WPMarkFloats`, `WPMarkMinipages`, `WPMarkTOC`, and `WPMarkLOFT`.

To help modify lwarp HTML output for easier import to a word processor, add

```
\booltrue{FormatWP}
```

to the project's source preamble after lwarp is loaded. The following changes are formatting adjustments then made to the HTML output:

- If using a class without chapters, `\section` and lower are shifted up in level for the HTML heading tags. The CSS has not been changed, so the section heading formats will not match the normal HTML output, but when imported to LibreOffice Writer the higher section headings will import as **Heading 1** for the title, **Heading 2** for `\section`, etc.
- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and `<div>`s, but they are no longer useful when the document has been imported into a word processor.

- An additional `<div>` with an `id` encapsulates each float and minipage, which on import into LibreOffice Writer causes a thin frame to appear around the text block for each.
- Float captions are given an explicit italic formatting.
- Tabular rule borders are made explicit for LibreOffice Writer. LibreOffice displays a light border around each cell while editing, even those which have no border when printed, and lwarp also uses a light border for thin rules, so it will be best to judge the results using the print preview instead of while editing in LibreOffice.
- `\includegraphics` and SVG math width and height are made explicit for LibreOffice.
- `\hspace` is approximated by a number of `\quad`s, and rules are approximated by a number of underscores.
- Explicit HTML styles are given to:
  - `\textsc`, etc.
  - `\underline`, soul and ulem markup.
  - `center`, `flushleft`, `flushright`.
  - `\marginpar`, keyfloat, sidenotes, floatflt, and wrapfig.
  - fancybox `\shadowbox`, etc.
  - The LaTeX and TeX logos.
- Honors several booleans:

  `WPMarkFloats`: Marks the begin and end of floats.

  `WPMarkMinipages`: Marks the begin and end of minipages.

  `WPMarkTOC`: Marks the location of the Table of Contents.

  `WPMarkLOFT`: Marks the locations of the List of Figures/Tables.

  `WPMarkMath`: Prints LaTeX math instead of using images.

  `WPTitleHeading`: Adjusts title and section headings.

  Several of these may be used to add markers to the HTML text which help determine where to adjust the word processor document after import.

10.2 Additional modifications

WPMarkFloats

Adds

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.

Bool `WPMarkFloats`
Default: `false`

WPMarkMinipages

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

Bool `WPMarkMinipages`
Default: `false`

WPMarkTOC

While formatting for word processors, adds

```
=== table of contents ===
```

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set `false`, the actual TOC is printed instead.

Bool `WPMarkTOC`
Default: `true`

WPMarkLOFT

While formatting for word processors, adds

```
=== list of figures ===
=== list of tables ===
```

and/or

Bool WPMarkLOFT
Default: false

where each of these lists would have been. This helps identify where to insert the actual lists.

If set `false`, the actual lists are printed instead.

WPMarkMath

While formatting for word processors, prints math as LaTeX code instead of creating SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

Bool WPMarkMath
Default: false
Prog TeXMaths
siunitx

When using the siunitx package, enter

```
\usepackage{siunitx}
```

in the TeXMaths preamble. Equation numbering is problematic for AMS math environments.

WPTitleHeading

While formatting for word processors, `true` sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to `false` to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Bool WPTitleHeading
Default: false
section headings

See table 6 on page 123.

10.3 Recommendations

TOC, LOF, LOT

For use with LibreOffice Writer, it is recommended to:

1. Set \booltrue{FormatWP}.
2. Set \booltrue{WPMarkTOC} and \boolfalse{WPMarkLOFT}.
3. Use lwarp to generate the HTML document.
4. Copy/paste from the HTML document into an empty LibreOffice Writer document.
5. Manually insert a LibreOffice TOC in the LibreOffice document.

Table 6: Section HTML headings for word-processor conversion

	HTML headings*			
	With `\chapter`		Without `\chapter`	
	`WPTitleHeading`		`WPTitleHeading`	
Section	`true`	`false`	`true`	`false`
Title	`<h1>`	plain	`<h1>`	plain
`\part`	`<h2>`	`<h1>`	`<h2>`	`<h1>`
`\chapter`	`<h3>`	`<h2>`	—	—
`\section`	`<h4>`	`<h3>`	`<h3>`	`<h2>`
`\subsection`	`<h5>`	`<h4>`	`<h4>`	`<h3>`
`\paragraph`	`<h6>`	`<h5>`	`<h5>`	`<h4>`
`\subparagraph`	span	`<h6>`	`<h6>`	`<h5>`

* For default depths when not `FormatWP`, see table 7 on page 132.

6. Manually add frames around each float, adding a caption which is cut/pasted from each float's simulated caption.
7. Manually create cross references.

This process yields a document with an actual LibreOffice Table of Contents, but a simulated List of Figures and List of Tables.

siunitx

For siunitx, remember to adjust the preamble as mentioned above.

LO view border options

LibreOffice has options in the View menu to turn on/off the display of thin borders around table cells and text objects.

10.4 Limitations

Floats and captions are not explicitly converted to LibreOffice floats with their own captions. Floats are surrounded by a thin frame in the LibreOffice editor, and may be marked with `WPMarkFloats`, but are not given a proper LibreOffice object frame. Captions are given an explicit italic formatting, but not a proper LibreOffice paragraph style.

Cross references are not actual LibreOffice linked cross references.

The List of Figures and List of Tables are not linked. The pasted pseudo LOF and LOT match the numbering of the LaTeX and HTML versions.

Equation numbering is not automatic, but the equation numbers in SVG math will match the LaTeX and HTML output. SVG math is recommended when using the AMS environments, which may have multiple numbered equations per object.

As of when last checked, LibreOffice ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
  - `\textcircled`
  - booktabs trim
- `\hspace` and rules, also used by algorithmic.
- Coloring of text decorations, used by soul and ulem.
- Overline text decoration, used by romanbar.

Libreoffice also has limitations with frames and backgrounds:

- Multiple lines in an object are framed individually instead of as a whole.
- Nested frames are not handled correctly.
- Images inside boxes are not framed correctly.
- Spans with background colors and frames are not displayed correctly.

11 Modifying lwarp

To quickly find a package in `lwarp.dtx`, search for `*packagename`.

Purely text-based packages probably will work as-is when generating HTML.

Look to existing code for ideas on how to expand into new code.

An environment may be converted to a `lateximage` then displayed with an image of the resulting LaTeX output. See section 77 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 43.8.

11.1 Creating an lwarp version of a package

When creating HTML, lwarp redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `lwarp-<packagename>.sty` version exists. If so, the lwarp version is used instead. This modular system allows users to create their own versions of packages for lwarp to use for HTML, simply by creating a new package with a `lwarp-` prefix. If placed in the local directory along with the source code, it will be seen by that project alone. If placed alongside the other `lwarp-` packages where TeX can see it, then the user's new package will be seen by any documents using lwarp. (Remember `mktexlsr` or `texhash`.)

An `lwarp-<packagename>.sty` package is only used during HTML generation. Its purpose is to pretend to be the original package, while modify anything necessary to create a successful HTML conversion. For many packages it is sufficient to simply provide nullified macros, lengths, counters, etc. for anything which the original package does, while passing the raw text on to be typeset. See the pre-existing `lwarp-` packages for examples.

Anything the user might expect of the original package must be replaced or emulated by the new `lwarp-` package, including package options, user-adjustable counters, lengths, and booleans, and conditional behaviors. In many of these packages, most of the new definitions have a "local" prefix according to the package name, and `@` characters inside the name, which hides these names from the user. In most cases these macros will not need to be emulated for HTML output. Only the "user-facing" macros need to be nullified or emulated.

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If "Drop"ped, the original print-version package is ignored, and only the lwarp- version is used. Use this where the original print version is useless for HTML. If "Pass"ed, the original package is loaded first, with the user-supplied options, then the lwarp- version continues loading as well. See section 218 (ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a lwarp- version.

11.1.1 Adding a package to the lwarp.dtx file

When adding a package to lwarp.dtx for permanent inclusing in lwarp, provide the lwarp-<packagename> code in lwarp.dtx, add its entry into lwarp.ins, and also remember to add

```
\LWR@loadafter{<packagename>}
```

to lwarp.dtx in section 23.1. This causes lwarp to stop with an error if packagename is loaded before lwarp.

11.2 Testing lwarp

When changes have been made, test the print output before testing the HTML. The print output compiles faster, and any errors in the printed version will be easier to figure out than the HTML version.

Remember that the configuration files are only rewritten when compiling the printed version of the document.

Sometimes it is worth checking the <project>_html.pdf file, which is the PDF containing HTML tags. Also, <project>_html.html has the text conversion of these tags, before the file is split into individual HTML files.

It is also worth checking the browser's tools for verifying the correctness of HTML and css code.

11.3 Modifying lwarpmk

Prog lwarpmk
File lwarpmk.lua

In most installations, lwarpmk.lua is an executable file located somewhere the operating system knows about, and it is called by typing "lwarpmk" into a terminal.

A project-local copy of `lwarpmk.lua` may be generated, modified, and then used to compile documents:

1. Add the `lwarpmk` option to the lwarp package.
2. Recompile the printed version of the document. The `lwarpmk` option causes lwarp to create a local copy of `lwarpmk.lua`
3. The `lwarpmk` option may now be removed from the lwarp package.
4. Copy and rename `lwarpmk.lua` to a new file such as `mymake.lua`.
5. Modify `mymake.lua` as desired.
6. If necessary, make `mymake.lua` executable.
7. Use `mymake.lua` instead of `lwarpmk.lua`.

To adjust the command-line arguments for compiling the document, look in `mymake.lua` for "`latexname`".

To adjust the command-line arguments for processing the index, look for "`xindy`".

12 Troubleshooting

12.1 Using the `lwarp.sty` package

Also see:
Section 7.7: Commands to be placed into the `warpprint` environment
Section 8: Special cases and limitations

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 7.1 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 7.3.

Tabular problems: See section 8.8.

Obscure error messages:

Print first: Be sure that a print version of the document compiles and that your document's LaTeX code is correct, before attempting to generate an HTML version.

Options clash: If using memoir, see section 8.11.

"Missing $ inserted.": If using a filename or URL in a footnote or `\item`, escape underscores with `\_`.

"Label(s) may have changed. Rerun to get cross-references right.": This warning may repeat endlessly if a math expression is used in a caption. Simple math expressions such as $X=1$ may be replaced with

```
\textit{X}\,=\,1
```

"Leaders not followed by proper glue": This can be caused by a missing `l@<floattype>` or `l@<sectiontype>` definition. See lwarp's definitions for examples.

"Improper `\prevdepth`": (Associated with `\LWR@orignewpage`.) Something tried to use `\ensuremath` where lwarp then tried to create a `lateximage`. If you can locate what used `\ensuremath`, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
```

inside a group first.

Missing sections:

- See section 7.3 regarding the `FileDepth` and `SideTOCDepth` counters, and the use of `\tableofcontents` in the home page.

Missing HTML files:

- See the warning regarding changes to the HTML settings at section 7.3.

Missing / incorrect cross-references:

- Use `lwarpmk again` followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.

labels

- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, underscore, and perhaps the colon (if not French).

⚠ underscores

`\nameref`

⚠ empty link

`\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

⚠ cleveref page numbers

- cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

  Ex:

  ```
  \cpageref{tab:first,tab:second}
  ```

  in HTML becomes:

  "pages **for** table 4.1 and **for** table 4.2"

  See `\cpagerefFor` at page 419 to redefine the message which is printed for page number references.

Em-dashes or En-dashes in listing captions and titles:

Use XeLaTeX or LuaLaTeX.

Floats out of sequence:

Mixed "Here" and floating: Floats [H]ere and regular floats may become out of order. `\clearpage` if necessary.

Caption setup: With `\captionsetup` set the positions for the captions `above` or `below` to match their use in the source code.

Print document contains HTML tags:

- Be sure that the document selects `\usepackage[warpprint]{lwarp}` instead of `[warpHTML]`.

Images are appearing in strange places:

- `lwarpmk limages` to refresh the `lateximage` images.

SVG images:

⚠ Adding/removing

When a math expression, `picture`, or Ti*k*z environment is added or removed, the SVG images must be re-created with `lwarpmk limages` to maintain the proper image file sequence numbers.

⚠ HTML instead of images

If HTML appears where an SVG image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time.

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as LaTeX would, but has the disadvantage of requiring an individual file for each math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time `$x$` is used, for example, yet another file is created. For a document with a large amount of math, see section 6.5 to use MathJax instead.

Plain-looking document:

- The document's CSS stylesheet may not be available, or may be linked incorrectly. Verify any `\CSSFilename` statements point to a valid CSS file.

Broken fragments of HTML:

- Check the PDF file used to create HTML to see if the tags overflowed the margin. (This is why such large page size and margins are used.)

Changes do not seem to be taking effect:

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 7.3.
- Verify that the proper CSS is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

Un-matched conditional compiles:

- Verify the proper `begin/end` of `warpprint`, `warpHTML`, and `warpall` environments.

12.1.1 Debug tracing output

`\tracinglwarp`

When `\tracinglwarp` is used, lwarp will add extra tracing messages to the `.log` file. The last several messages may help track down errors.

Place `\tracinglwarp` just after `\usepackage{lwarp}` to activate tracing.

12.2 Compiling the `lwarp.dtx` file

`lwarp_tutorial.tex`: Copy or link `lwarp_tutorial.txt` from the TDS `doc` directory to the `source` directory, or wherever you wish to compile the documentation. This file is included verbatim into the documentation, but is in the `doc` directory so that it may be found by `texdoc` and copied by the user.

Illogical error messages caused by an out-of-sync `lwarp.sty` file:

1. Delete the `lwarp.sty` file.
2. `pdflatex lwarp.ins` to generate a new `lwarp.sty` file.
3. `pdflatex lwarp.dtx` to recompile the `lwarp.pdf` documentation.

Un-nested environments:

Be sure to properly nest:

- `\begin{macrocode}` and `\end{macrocode}`
- `\begin{macro}` and `\end{macro}`
- `\begin{environment}` and `\end{environment}`

File 1 `lwarp.sty`

13 Implementation

This package is perhaps best described as a large collection of smaller individual technical challenges, in many cases solved through a number of ~~crude hacks~~ clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

Judgement calls were made, and are often commented. Improvements are possible. The author is open to ideas and suggestions.

Packages were patched for re-use where they provided significant functionality. Examples include xcolor with its color models and conversion to HTML color output, and siunitx which provides many number and unit-formatting options, almost all of which are available in pure-text form, and thus easily used by `pdftotext`.

Packages were emulated where their primary purpose was visual formatting which is not relevent to HTML output. For example, packages related to sectioning are already patched by numerous other packages, creating a difficult number of combinations to try to support, and yet in HTML output all of the formatting is thrown away, so these packages are merely emulated.

Table 7: Section depths and HTML headings

Section	LaTeX depth	HTML headings *
title of the entire website		`<h1>`
none	-5	new for this package
book	-2	**not yet used**
part	-1	`<h2>`
chapter	0	`<h3>`
section	1	`<h4>`
subsection	2	`<h5>`
subsubsection	3	`<h6>`
paragraph	4	`<span class = "paragraph">`
subparagraph	5	`<span class = "subparagraph">`
listitem	7	new for this package, used for list items

* If `FormatWP` is true, section headings may be adjusted, depending on `WPTitleHeading`. See table 6 on page 123.

Packages with graphical output are allowed as-is, but must be nested inside a `lateximage` environment to preserve the graphics.

Testing has primarily been done with the Iceweasel/Firefox browser.

14 Section depths and HTML headings

Stacks are created to track depth inside the LaTeX document structure. This depth is translated to HTML headings as shown in table 7. "Depth" here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the LaTeX document structure.

When starting a new section, the program first must close out any existing sections and lists of a deeper level to keep the HTML tags nested correctly.

Support for the memoir package will require the addition of a `book` level, which may push the HTML headings down a step, and also cause `subsubsection` to become a `<div>` due to a limit of six HTML headings.

It is possible to use HTML5 `<section>` and `<h1>` for all levels, but this may not be well-recognized by older browsers.

Fixed levels for parts and chapters allow the CSS to remain fixed as well.

15 Source Code

This is where the documented source code for lwarp begins, continuing through the following sections all the way to the change log and index at the end of this document.

The following sections document the actual implementation of the lwarp package.

line numbers The small numbers at the left end of a line refer to line numbers in the `lwarp.sty` file.

subjects Blue-colored tags in the left margin aid in quickly identifying the subject of each paragraph.

objects Black-colored tags in the left marign are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are command macros. Each of these also appears in the index as individual entries, and are also listed together under “files”, “packages”, “environments”, “booleans”, and “counters”.

index entries

⚠ warnings Special warnings are marked with a warning icon.

for HTML output:
for PRINT output:
for HTML & PRINT:

Green-colored tags in the left margin show which sections of source code apply to the generation of HTML, print, or both forms of output.

— lwarp source code begins on the following page —

16 Detecting the TEX Engine — pdflatex, lualatex, xelatex

```
\RequirePackage{iftex}

\ifLuaTeX
\RequirePackage{luatex85}% until the geometry package is updated
\fi
```

17 pdfLaTeX T1 and UTF8 encoding

When using pdfLATEX, lwarp required T1 and UTF8 encoding.

XELATEX and LuaLATEX are both UTF8 by nature.

```
\ifPDFTeX
\RequirePackage[T1]{fontenc}
\RequirePackage[utf8]{inputenc}
\fi
```

18 Unicode input characters

for HTML & PRINT:

If using `pdflatex`, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

A commonly-used multiply symbol is declared to be `\texttimes`.

The first arguments of `\newunicodechar` below are text ligatures in the source code, even though they are not printed in the following listing.

```

\RequirePackage{newunicodechar}

\newunicodechar{×}{\texttimes}

\ifPDFTeX
\newunicodechar{ff}{ff}% the first arguments are ligatures
\newunicodechar{fi}{fi}
\newunicodechar{fl}{fl}
\newunicodechar{ffi}{ffi}
\newunicodechar{ffl}{ffl}
\newunicodechar{--}{---}
\newunicodechar{-}{--}
```

In PDFTEX, preserve upright quotes in verbatim text:

```
\RequirePackage{upquote}
\else
\fi
```

19 Miscellaneous tools

`\LWR@providelength` {⟨`\lengthname`⟩} Provides the length if it isn't defined yet.

Used to provide source compatibility for lengths which will be ignored, but might or might not be already provided by other packages.

```
\newcommand*\LWR@providelength[1]{%
    \ifdeflength{#1}{}{\newlength{#1}}%
}
```

20 Early package requirements

Pkg `etoolbox` Provides `\ifbool` and other functions.

Pkg `xpatch` Patches macros with optional arguments.

```
\RequirePackage{etoolbox}[2011/01/03]% v2.6 for \BeforeBeginEnvironment, etc.
\RequirePackage{xpatch}
```

Pkg `ifplatform` Provides `\ifwindows` to try to automatically detect Windows OS.

```
\RequirePackage{ifplatform}% sense op-system platform
```

Pkg `letltxmacro` Used to redefine `\textbf` and friends.

```
\RequirePackage{letltxmacro}
```

21 Operating-System portability

Prog Unix
Prog Mac OS
Prog Linux
Prog MS-Windows
Prog Windows
Opt OSWindows

lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as "UNIX" in the configuration files), and MS-WINDOWS is supported as well.

If WINDOWS is not correctly detected, use the lwarp option `OSWindows`.

When detected or specified, the operating-system path separator used by lwarp is modified, the boolean `usingOSWindows` is set true. This boolean may be tested by the user for later use.

21.1 Common portability code

Bool usingOSWindows

Set if the `OSWindows` option is used.

```
\newbool{usingOSWindows}
\boolfalse{usingOSWindows}
```

21.2 Unix, Linux, and Mac OS

\OSPathSymbol

Symbol used to separate directories in a path.

```
\newcommand*{\OSPathSymbol}{/}
```

21.3 MS-Windows

For MS-Windows:

\LWR@setOSWindows

Set defaults for the MS-Windows operating system. lwarp attempts to auto-detect the operatings system, and the `OSWindows` option may also be used to force MS-Windows compatibility.

```
\newcommand*{\LWR@setOSWindows}
{
\booltrue{usingOSWindows}
\renewcommand*{\OSPathSymbol}{\@backslashchar}
}
```

Test for windows during compile. The user may also specify `OSWindows` package option in case this test fails.

```
\ifwindows
\LWR@setOSWindows
\fi
```

22 Package options

Pkg `kvoptions` Allows key/value package options.

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool `warpingprint`

Bool `warpingHTML`

Bool `mathjax`

Set to true/false depending on the package option selections for print/HTML/EPUB output and mathsvg/mathjax:

```
\newbool{warpingprint}
\newbool{warpingHTML}
\newbool{mathjax}
```

defaults The default is print output, and SVG math if the user chose HTML output.

```
\booltrue{warpingprint}%
\boolfalse{warpingHTML}%
\boolfalse{mathjax}%
```

Opt `warpprint` If the `warpprint` option is given, boolean `warpingprint` is true and boolean `warpingHTML` is false, and may be used for `\ifbool` tests.

```
\DeclareVoidOption{warpprint}{%
\PackageInfo{lwarp}{Using option 'warpprint'}
\booltrue{warpingprint}%
\boolfalse{warpingHTML}%
}
```

Env `warpHTML` Anything in the `warpHTML` environment will be generated for HTML output only.

Opt `warpHTML` If the `warpHTML` option is given, boolean `warpingHTML` is true and boolean `warpingprint` is false, and may be used for `\ifbool` tests.

```
\DeclareVoidOption{warpHTML}{%
```

```
\PackageInfo{lwarp}{Using option 'warpHTML'}%
\booltrue{warpingHTML}%
\boolfalse{warpingprint}%
}
```

Opt mathsvg Option mathsvg selects SVG math display: If the mathsvg option is given, boolean mathjax is false, and may be used for \ifbool tests.

```
\DeclareVoidOption{mathsvg}{%
\PackageInfo{lwarp}{Using option 'mathsvg'}
\boolfalse{mathjax}%
}
```

Opt mathjax Option mathjax selects MathJax math display: If the mathjax option is given, boolean mathjax is true, may be used for \ifbool tests.

```
\DeclareVoidOption{mathjax}{%
\PackageInfo{lwarp}{Using option 'mathjax'}
\booltrue{mathjax}%
}
```

Opt BaseJobname Option BaseJobname sets the \BaseJobname for this document.

This is the \jobname of the printed version, even if currently compiling the HTML version. I.e. this is the \jobname without _html appended. This is used to set \HomeHTMLFilename if the user did not provide one.

```
\DeclareStringOption[\jobname]{BaseJobname}
```

Opt IndexLanguage Sets the language to be assigned in lwarpmk's configuration files. This is then used by lwarpmk while processing the index and glossary.

```
\DeclareStringOption[english]{IndexLanguage}
```

Opt xdyFilename Selects a custom .xdy file. The default is lwarp.xdy. A customized file should be based on lwarp.xdy, and must retain the line

```
(markup-locref :open "\hyperindexref{" :close "}")
```

```
\DeclareStringOption[lwarp.xdy]{xdyFilename}
```

Opt lwarpmk Tells lwarp to generate a local copy of lwarpmk called lwarpmk.lua. Useful for archiving for future use. This file may be made executable and acts just like lwarpmk.

If lwarpmk option, creates a local copy of lwarpmk.lua:

```
\newbool{LWR@creatinglwarpmk}
\boolfalse{LWR@creatinglwarpmk}

\DeclareVoidOption{lwarpmk}{
\PackageInfo{lwarp}{Using option 'lwarpmk'}
\booltrue{LWR@creatinglwarpmk}
}
```

Opt `OSWindows` Tells lwarp to use MS-Windows compatibility. Auto-detection of the operating system is attempted, and this option is only necessary if the auto-detection fails. See the automatically-generated `lwarpmk.conf` file to find out whether the operating system was detected correctly.

```
\DeclareVoidOption{OSWindows}{
\PackageInfo{lwarp}{Using option 'OSWindows'}
\LWR@setOSWindows
}
```

Opt `HomeHTMLFilename` Default: `\lwarp` The filename of the homepage. The default is the jobname. This option is stored into `\LWR@HomeHTMLFilename`, and later transferred into `\HomeHTMLFilename` for internal use.

```
\DeclareStringOption[]{HomeHTMLFilename}
```

Opt `HTMLFilename` Default: `<empty>` The filename prefix of web pages after the homepage. The default is empty, no prefix. This option is stored into `\LWR@HTMLFilename`, and later transferred into `\HTMLFilename` for internal use.

```
\DeclareStringOption[]{HTMLFilename}
```

Opt `latexmk` Option `latexmk` tells `lwarpmk` to use `latexmk` when compiling documents.

```
\DeclareBoolOption[false]{latexmk}
```

Execute options Execute the package options, with the defaults which have been set just above:

```
\ProcessKeyvalOptions*\relax
```

Assign the `\BaseJobname` if the user hasn't provided one:

```
\providecommand*{\BaseJobname}{\LWR@BaseJobname}
```

Defaults unless already over-ridden by the user:

```
\ifcsempty{LWR@HomeHTMLFilename}{
\newcommand*{\HomeHTMLFilename}{\BaseJobname}
```

```
}{
\csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
}

\csedef{HTMLFilename}{\LWR@HTMLFilename}
```

22.1 Conditional compilation

`\warpprintonly` {⟨*contents*⟩}

Only process the contents if producing printed output.

```
\newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}
```

`\warpHTMLonly` {⟨*contents*⟩}

Only process the contents if producing HTML output.

```
\newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}
```

Pkg `comment` Provides conditional code blocks.

```
\RequirePackage{comment}
```

Use `comment_print.cut` for print mode, and `comment_html.cut` for HTML mode. This helps `latexmk` to more reliably know whether to recompile.

```
\ifbool{warpingHTML}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
}{}

\ifbool{warpingprint}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
}{}

\excludecomment{testing}
```

Env `warpall` Anything in the `warpall` environment will be generated for print or HTML outputs.

```
\includecomment{warpall}
```

Env `warpprint` Anything in the `warpprint` environment will be generated for print output only.

Env `warpHTML`

For HTML output:

```
\ifbool{warpingHTML}{%
\includecomment{warpHTML}
}
{\excludecomment{warpHTML}}%

\ifbool{warpingprint}
{\includecomment{warpprint}}
{\excludecomment{warpprint}}
```

Optionally generate a local copy of `lwarpmk`. Default to no.

```
\ifbool{LWR@creatinglwarpmk}
{\includecomment{LWR@createlwarpmk}}
{\excludecomment{LWR@createlwarpmk}}
```

23 Package load order

Several packages should only be loaded before lwarp, and most others should only be loaded after.

Packages which should only be loaded before lwarp have their own

```
lwarp-<packagename>.sty
```

which use `\LWR@loadbefore` to trigger an error if they are loaded after lwarp. Examples include fontspec, inputenc, fontenc, and newunicodechar.

Most packages should be loaded after lwarp. This is enfoced by a large number of `\LWR@loadafter` statements, below.

Some packages are emulated by memoir, and so these are tested by `\LWR@notmemoirloadafter`, which does not cause an error if memoir is used.

23.1 Tests of package load order

`\LWR@loadafter` {⟨*packagename*⟩} Error if this package was loaded before lwarp.

```
\newcommand*{\LWR@loadafter}[1]{%
\@ifpackageloaded{#1}
{
\PackageError{lwarp}
```

```
{Package #1, or one which uses #1, must be loaded after lwarp}
{Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
Package #1 may also be loaded by something else, which must also be moved
after lwarp.}
}
{}
}
```

`\LWR@notmemoirloadafter` {⟨*packagename*⟩} Error if not memoir class and this package was loaded before lwarp.

memoir emulates many packages, and pretends that they have already been loaded.

```
\@ifclassloaded{memoir}
{\newcommand*{\LWR@notmemoirloadafter}[1]{}}
{\LetLtxMacro\LWR@notmemoirloadafter\LWR@loadafter}
```

`\LWR@loadbefore` {⟨*packagename*⟩} Error if this package is after lwarp.

```
\newcommand*{\LWR@loadbefore}[1]{%
\@ifpackageloaded{#1}
{}
{
\PackageError{lwarp}
{Package #1 must be loaded before lwarp}
{Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
}
}
```

`\LWR@loadnever` {⟨*badpackagename*⟩} {⟨*replacementpkgname*⟩}

The first packages is not supported, so tell the user to use the second instead.

```
\newcommand*{\LWR@loadnever}[2]{%
\PackageError{lwarp}
{Package #1 is not supported by lwarp's HTML conversion.
Package(s) #2 may be useful instead}
{Package #1 might conflict with lwarp in some way,
or is superceded by another package.
For a possible alternative, see package(s) #2.}
}
```

23.2 Enforcing package loading after lwarp

Packages which should only be loaded after lwarp are tested here to trip an error of they have already been loaded.

The following packages must be loaded after lwarp:

```
\LWR@loadafter{a4}
\LWR@loadafter{a4wide}
\LWR@loadafter{a5comb}
\LWR@notmemoirloadafter{abstract}
\LWR@loadafter{adjmulticol}
\LWR@loadafter{addlines}
\LWR@loadafter{afterpage}
\LWR@loadafter{algorithmicx}
\LWR@loadafter{alltt}
\LWR@loadafter{amsmath}
\LWR@loadafter{amsthm}
\LWR@loadafter{anonchap}
\LWR@loadafter{anysize}
\LWR@notmemoirloadafter{appendix}
\LWR@loadafter{arabicfront}
\LWR@notmemoirloadafter{array}
\LWR@loadafter{atbegshi}
\LWR@loadafter{authblk}
\LWR@loadafter{backref}
\LWR@loadafter{balance}
\LWR@loadafter{bigdelim}
\LWR@loadafter{bigstrut}
\LWR@loadafter{bookmark}
\LWR@notmemoirloadafter{booktabs}
\LWR@loadafter{boxedminipage}
\LWR@loadafter{boxedminipage2e}
\LWR@loadafter{breakurl}
\LWR@loadafter{cancel}
\LWR@notmemoirloadafter{ccaption}
\LWR@loadafter{changebar}
\LWR@notmemoirloadafter{changepage}
\LWR@notmemoirloadafter{chngpage}
\LWR@loadafter{chappg}
\LWR@loadafter{cite}
\LWR@loadafter{color}
\LWR@loadafter{continue}
\LWR@notmemoirloadafter{crop}
\LWR@loadafter{cuted}
\LWR@loadafter{cutwin}
\LWR@loadafter{dblfnote}
\LWR@notmemoirloadafter{dcolumn}
\LWR@loadafter{draftwatermark}
```

```
\LWR@loadafter{easy-todo}
\LWR@loadafter{ebook}
\LWR@loadafter{ellipsis}
\LWR@loadafter{emptypage}
\LWR@loadafter{endfloat}
\LWR@loadafter{endnotes}
\LWR@notmemoirloadafter{enumerate}
\LWR@loadafter{enumitem}
\LWR@notmemoirloadafter{epigraph}
\LWR@loadafter{eso-pic}
\LWR@loadafter{everypage}
\LWR@loadafter{everyshi}
\LWR@loadafter{extramarks}
\LWR@loadafter{fancybox}
\LWR@loadafter{fancyhdr}
\LWR@loadafter{fancyref}
\LWR@loadafter{fancyvrb}
\LWR@loadafter{figcaps}
\LWR@loadafter{fix2col}
\LWR@loadafter{fixme}
\LWR@loadafter{fixmetodonotes}
\LWR@loadafter{flafter}
\LWR@loadafter{float}
\LWR@loadafter{floatflt}
\LWR@loadafter{floatpag}
\LWR@loadafter{floatrow}
\LWR@loadafter{fltrace}
\LWR@loadafter{flushend}
\LWR@loadafter{fncychap}
\LWR@loadafter{fnpos}
% fontenc must be loaded before lwarp
% fontspec must be loaded before lwarp
\LWR@loadafter{footmisc}
\LWR@loadafter{footnote}
\LWR@loadafter{footnotehyper}
\LWR@loadafter{footnpag}
\LWR@loadafter{framed}
\LWR@loadafter{ftnright}
\LWR@loadafter{fullpage}
\LWR@loadafter{fullwidth}
\LWR@loadafter{fwlw}
\LWR@loadafter{geometry}
\LWR@loadafter{glossaries}
% \LWR@loadafter{graphics}% pre-loaded by xunicode
% \LWR@loadafter{graphicx}% pre-loaded by xunicode
\LWR@loadafter{grffile}
\LWR@loadafter{hang}
\LWR@loadafter{hanging}
\LWR@loadafter{hyperref}
\LWR@loadafter{hyperxmp}
```

```
\LWR@loadafter{idxlayout}
\LWR@loadafter{ifoddpage}
\LWR@loadafter{indentfirst}
% inputenc must be loaded before lwarp
\LWR@loadafter{keyfloat}
\LWR@loadafter{layout}
\LWR@loadafter{letterspace}
\LWR@loadafter{lettrine}
\LWR@loadafter{lips}
\LWR@loadafter{listings}
\LWR@loadafter{longtable}
\LWR@loadafter{lscape}
\LWR@loadafter{ltcaption}
\LWR@loadafter{ltxtable}
\LWR@loadafter{luatodonotes}
\LWR@loadafter{marginfit}
\LWR@loadafter{marginfix}
\LWR@loadafter{marginnote}
\LWR@loadafter{mcaption}
\LWR@loadafter{mdframed}
\LWR@loadafter{memhfixc}
\LWR@loadafter{metalogo}
\LWR@loadafter{microtype}
\LWR@loadafter{midfloat}
\LWR@loadafter{midpage}
\LWR@notmemoirloadafter{moreverb}
% morewrites must be loaded before lwarp
\LWR@notmemoirloadafter{mparhack}
%\LWR@loadafter{multicol}% loaded by ltxdoc
\LWR@loadafter{multirow}
\LWR@loadafter{nameref}
\LWR@loadafter{natbib}
\LWR@notmemoirloadafter{needspace}
% newclude must be loaded before lwarp
\LWR@loadafter{newtxmath}
% newunicodechar must be loaded before lwarp
\LWR@notmemoirloadafter{nextpage}
\LWR@loadafter{nonumonpart}
\LWR@loadafter{nopageno}
\LWR@loadafter{nowidow}
\LWR@loadafter{ntheorem}
\LWR@loadafter{overpic}
\LWR@notmemoirloadafter{pagenote}
\LWR@loadafter{pagesel}
\LWR@loadafter{paralist}
\LWR@notmemoirloadafter{parskip}
\LWR@loadafter{pdflscape}
\LWR@loadafter{pdfsync}
\LWR@loadafter{pfnote}
\LWR@loadafter{placeins}
```

```
\LWR@loadafter{prelim2e}
\LWR@loadafter{prettyref}
\LWR@loadafter{preview}
\LWR@loadafter{quotchap}
\LWR@loadafter{ragged2e}
\LWR@loadafter{realscripts}
\LWR@loadafter{relsize}
\LWR@loadafter{romanbar}
\LWR@loadafter{romanbarpagenumber}
\LWR@loadafter{rotating}
\LWR@loadafter{rotfloat}
\LWR@loadafter{savetrees}
% \LWR@loadafter{scalefnt}% loaded by babel-french
\LWR@loadafter{scrextend}
\LWR@loadafter{scrhack}
\LWR@loadafter{scrlayer}
\LWR@loadafter{scrlayer-notecolumn}
\LWR@loadafter{scrlayer-scrpage}
\LWR@loadafter{section}
\LWR@loadafter{sectsty}
\LWR@notmemoirloadafter{setspace}
\LWR@loadafter{shadow}
\LWR@notmemoirloadafter{showidx}
\LWR@loadafter{showkeys}
\LWR@loadafter{sidecap}
\LWR@loadafter{sidenotes}
\LWR@loadafter{siunitx}
\LWR@loadafter{soul}
\LWR@loadafter{soulpos}
\LWR@loadafter{soulutf8}
\LWR@loadafter{stabular}
\LWR@loadafter{subfig}
\LWR@loadafter{subfigure}
\LWR@loadafter{supertabular}
\LWR@loadafter{tabls}
\LWR@notmemoirloadafter{tabularx}
\LWR@loadafter{tabulary}
\LWR@loadafter{textarea}
% \LWR@loadafter{textcomp}% maybe before lwarp with font packages
\LWR@loadafter{textfit}
\LWR@loadafter{textpos}
\LWR@loadafter{theorem}
\LWR@loadafter{threeparttable}
\LWR@loadafter{tikz}
\LWR@loadafter{titleps}
\LWR@loadafter{titlesec}
\LWR@loadafter{titletoc}
\LWR@notmemoirloadafter{titling}
% \LWR@loadafter{tocbasic}% preloaded by koma-script classes
\LWR@notmemoirloadafter{tocbibind}
```

```
\LWR@notmemoirloadafter{tocloft}
\LWR@loadafter{tocstyle}
\LWR@loadafter{todo}
\LWR@loadafter{todonotes}
\LWR@loadafter{transparent}
\LWR@loadafter{trivfloat}
\LWR@loadafter{turnthepage}

% \LWR@loadafter{typearea}% preloaded by koma-script classes
\LWR@loadafter{ulem}
\LWR@loadafter{upref}
\LWR@loadafter{varioref}
\LWR@notmemoirloadafter{verse}
\LWR@loadafter{wallpaper}
\LWR@loadafter{watermark}
\LWR@loadafter{wrapfig}
\LWR@loadafter{xcolor}
\LWR@loadafter{xfrac}
\LWR@loadafter{xltxtra}
\LWR@loadafter{xmpincl}
\LWR@loadafter{xtab}
\LWR@loadafter{zwpagelayout}
```

24 Required packages

These packages are automatically loaded by lwarp when generating HTML output. Some of them are also automatically loaded when generating print output, but some are not.

In the document preamble, create a `warpprint` environment, and place inside it any of the following packages which are required and which are labeled as "Print: OK to Load in a `warpprint` environment". Those packages which are labeled as "Print: Pre-Loaded" need not be placed into the document preamble.

for HTML & PRINT:

```
\begin{warpall}
```

See: http://tex.stackexchange.com/a/47579.

Detects XeTeX and LuaLaTeX:

```
\RequirePackage{iftex}
\newif\ifxetexorluatex
\ifXeTeX
    \xetexorluatextrue
\else
    \ifLuaTeX
```

```
        \xetexorluatextrue
    \else
        \xetexorluatexfalse
    \fi
\fi

\end{warpall}
```

for HTML output:

```
\begin{warpHTML}

\ifxetexorluatex
% ^^A     \usepackage[no-math]{fontspec}
```

The monospaced font is used for HTML tags, so turn off its TeX ligatures and common ligatures:

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
\else
```

pdflatex only: Only pre-loaded if pdflatex is being used.

Pkg microtype

ligatures Older browsers don't display ligatures. Turn off letter ligatures, keeping LaTeX dash and quote ligatures, which may fail on older browers but at least won't corrupt written words.

```
\RequirePackage {microtype}

\microtypesetup{
    protrusion=false,
    expansion=false,
    tracking=false,
    kerning=false,
    spacing=false}

\DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

\fi

\end{warpHTML}
```

Pkg geometry Tactics to avoid unwanted page breaks and margin overflow:

- Uses a very long and wide page to minimize page breaks and margin overflow.

- Uses a scriptsize font.
- Uses extra space at the margin to avoid HTML tag overflow off the page.
- Forces a new PDF page before some environments.
- Forces line break between major pieces of long tags.

for HTML output:

```
\begin{warpHTML}
\RequirePackage[paperheight=190in,paperwidth=20in,%
left=2in,right=12in,%
top=1in,bottom=1in,%
]{geometry}
\@twosidefalse
\@mparswitchfalse
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg `xparse`

LaTeX3 command argument parsing

```
\RequirePackage{xparse}
```

Pkg `afterpackage`

Used to patch titling to add `\AddSubtitlePublished`. Provided by scrlfile or afterpackage.

```
\@ifundefined{AfterPackage}%
{
\RequirePackage{afterpackage}
}{}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg `expl3`

LaTeX3 programming

```
\RequirePackage{expl3}
```

Pkg `gettitlestring`

Used to emulate `\nameref`.

```
\RequirePackage{gettitlestring}
```

Pkg everyhook

everyhook is used to patch paragraph handling.

```
\RequirePackage{everyhook}
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg filecontents

Used to write helper files, done in print mode.

Patched to work with morewrites, per https://tex.stackexchange.com/questions/312830/does-morewrites-not-support-filecontents-and-can-i-write-body-of-environment-us/312910

```
\RequirePackage{filecontents}

\@ifpackagelater{filecontents}{2011/10/09}%
{}
{
\newwrite\fcwrite
\let\LWR@origfilec@ntents\filec@ntents
\def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg xifthen

```
\RequirePackage{xifthen}
```

Pkg xstring

```
\RequirePackage{xstring}
```

Pkg xstring

```
\RequirePackage{verbatim}
```

Pkg makeidx

```
\RequirePackage{makeidx}
\makeindex
```

Pkg calc

```
\RequirePackage{calc}
```

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

```
\RequirePackage{refcount}
```

Pkg newfloat

```
\RequirePackage{newfloat}
```

Pkg caption

```
\RequirePackage{caption}
```

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg environ

Used to encapsulate math environments for re-use in HTML <alt> text.

```
\RequirePackage{environ}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg zref

Used for cross-references.

```
\RequirePackage{zref}
```

Pkg amsmath

Equation numbers are placed to the left for HTML.

newtxmath automatically loads amsmath, so the options leqno and fleqn are passed beforehand to be picked up both here and by newtxmath if it is used.

```
\PassOptionsToPackage{leqno}{amsmath}
\RequirePackage{amsmath}
```

Used to convert lengths for image width/height options.

```
\RequirePackage{printlen}
```

```
\end{warpHTML}
```

25 Loading packages

for HTML output:

```
\begin{warpHTML}
```

Remember the original \RequirePackage:

```
\LetLtxMacro{\LWR@origRequirePackage}{\RequirePackage}
```

\LWR@requirepackagenames Stores the list of required package names.

```
\newcommand*{\LWR@requirepackagenames}{}
```

\LWR@findword [⟨*1: separator*⟩] {⟨*2: list*⟩} {⟨*3: index*⟩} [⟨*4: destination*⟩]

Note that argument 4 is passed directly to \StrBetween.

```
\newcommand*\LWR@findword[3][,]{%
    \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
}
```

\LWR@lookforpackagename {⟨*index*⟩} If this is a package name, re-direct it to the lwarp version by renaming it lwarp- followed by the original name.

```
\newcommand*{\LWR@lookforpackagename}[1]{%
```

Find the n'th package name from the list:

```
\LWR@findword{\LWR@requirepackagenames}{#1}[\LWR@strresult]%
```

Remove blanks. The original name with blanks is in LWR@strresult and the final name with no blanks goes into LWR@strresulttwo.

```
\LWR@traceinfo{LWR@lookforpackagename with blanks: !\LWR@strresult!}%
\StrSubstitute[100]{\LWR@strresult}{ }{}[\LWR@strresulttwo]%
\LWR@traceinfo{LWR@lookforpackagename without blanks: !\LWR@strresulttwo!}%
```

See if the package name was found:

```
\IfStrEq{\LWR@strresulttwo}{}%
{}% no filename
{% yes filename
```

If found, and if an lwarp-equivalent name exists, use `lwarp-*` instead.

```
\IfFileExists{lwarp-\LWR@strresulttwo.sty}%
{% lwarp-* file found
        \StrSubstitute%
            {\LWR@requirepackagenames}%
            {\LWR@strresult}%
            {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
}%
{


        \StrSubstitute%
            {\LWR@requirepackagenames}%
            {\LWR@strresult}%
            {\LWR@strresulttwo}[\LWR@requirepackagenames]%
}% no lwarp-* file
}% yes filename
}
```

`\RequirePackage` `[`⟨*1: options*⟩`] {`⟨*2: package names*⟩`} [`⟨*3: version*⟩`]`

For each of many package names in a comma-separated list, if an lwarp version of a package exists, select it instead of the LaTeX version.

```
\RenewDocumentCommand{\RequirePackage}{o m o}{%
```

Redirect up to nine names:

```
\renewcommand*{\LWR@requirepackagenames}{#2}
\LWR@lookforpackagename{1}
\LWR@lookforpackagename{2}
\LWR@lookforpackagename{3}
\LWR@lookforpackagename{4}
\LWR@lookforpackagename{5}
\LWR@lookforpackagename{6}
\LWR@lookforpackagename{7}
\LWR@lookforpackagename{8}
\LWR@lookforpackagename{9}
```

`\RequirePackage` depending on the options and version:

```
\IfValueTF{#1}
{% options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
    {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
}
{% no options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
    {\LWR@origRequirePackage{\LWR@requirepackagenames}}
}
}
\LetLtxMacro{\usepackage}{\RequirePackage}
```

`\LWR@ProvidesPackagePass` {⟨*pkgname*⟩} [⟨*version*⟩]

Uses the original package, including options.

```
\NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
\PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
\IfValueTF{#2}
{\ProvidesPackage{lwarp-#1}[#2]}
{\ProvidesPackage{lwarp-#1}}
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
\ProcessOptions\relax

\IfValueTF{#2}
{\LWR@origRequirePackage{#1}[#2]}
{\LWR@origRequirePackage{#1}}
}
```

`\LWR@ProvidesPackageDrop` {⟨*pkgname*⟩} [⟨*version*⟩]

Ignores the original package and uses lwarp's version instead. Drops/discards all options.

```
\NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
\PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
\IfValueTF{#2}
{\ProvidesPackage{lwarp-#1}[#2]}
{\ProvidesPackage{lwarp-#1}}
\DeclareOption*{}
\ProcessOptions\relax
}


\end{warpHTML}
```

26 File handles

Defines file handles for writes.

for HTML & PRINT:
```
\begin{warpall}
```

`\LWR@quickfile` For quick temporary use only. This is reused in several places.

```
\newwrite\LWR@quickfile%
```

```
\end{warpall}
```

for HTML output:
```
\begin{warpHTML}
```

`\LWR@lateximagesfile` For `lateximages.txt`.

```
\newwrite\LWR@lateximagesfile
```

```
\end{warpHTML}
```

27 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. lwarp creates `<filename>_html_inc.tex` whose contents are:
   `\input <filename>.tex`
2. `<filename>_html_inc.tex` is then `\included` instead of `<filename>.tex`.
3. `<filename>_html_inc.aux` is automatically generated and used by LaTeX.

for HTML output:
```
\begin{warpHTML}
```

`\include` `{`⟨*filename*⟩`}`

`\@include` `{`⟨*filename*⟩`}` Modified to load `_html_inc` files.

```
\def\@include#1 {%
\immediate\openout\LWR@quickfile #1_html_inc.tex% new
\immediate\write\LWR@quickfile{\string\input{#1.tex}}% new
```

```
\immediate\closeout\LWR@quickfile% new
\LWR@origclearpage% \changed
\if@filesw
    \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
\fi
\@tempswatrue
\if@partsw
    \@tempswafalse
    \edef\reserved@b{#1}%
    \@for\reserved@a:=\@partlist\do
    {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
\fi
\if@tempswa
    \let\@auxout\@partaux
    \if@filesw
        \immediate\openout\@partaux #1_html_inc.aux % changed
        \immediate\write\@partaux{\relax}%
    \fi
    \@input@{#1_html_inc.tex}% changed
    \LWR@origclearpage% changed
    \@writeckpt{#1}%
    \if@filesw
        \immediate\closeout\@partaux
    \fi
\else
    \deadcycles\z@
    \@nameuse{cp@#1}%
\fi
\let\@auxout\@mainaux%
}

\end{warpHTML}
```

28 Copying a file

for HTML output:

```
\begin{warpHTML}
```

`\LWR@copyfile` {⟨*source filename*⟩} {⟨*destination filename*⟩}

Used to copy the `.toc` file to `.sidetoc` to re-print the TOC in the sideTOC navigation pane.

```
\newwrite\LWR@copyoutfile  % open the file to write to
\newread\LWR@copyinfile      % open the file to read from

\newcommand*{\LWR@copyfile}[2]{%
```

```
\LWR@traceinfo{LWR@copyfile: copying #1 to #2}

\immediate\openout\LWR@copyoutfile=#2
\openin\LWR@copyinfile=#1
\begingroup\endlinechar=-1
\makeatletter

\LWR@traceinfo{LWR@copyfile: about to loop}

\loop\unless\ifeof\LWR@copyinfile
  \LWR@traceinfo{LWR@copyfile: one line}
  \read\LWR@copyinfile to\LWR@fileline  % Read one line and store it into \LWR@fileline
%   \LWR@fileline\par                   % print the content into the pdf
% print the content:
  \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
\repeat
\immediate\closeout\LWR@copyoutfile
\LWR@traceinfo{LWR@copyfile: done}
\endgroup
}

\end{warpHTML}
```

29 Debugging messages

for HTML & PRINT:

```
\begin{warpall}
```

Bool `LWR@tracinglwarp` True if tracing is turned on.

```
\newbool{LWR@tracinglwarp}
```

`\tracinglwarp` Turns on the debug tracing messages.

```
\newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}
```

`\LWR@traceinfo` `{`⟨*text*⟩`}` If tracing is turned on, writes the text to the `.log` file.

```
\newcommand{\LWR@traceinfo}[1]{%
\ifbool{LWR@tracinglwarp}%
{%
    \typeout{*** lwarp: #1}%
    % \PackageInfo{lwarp}{#1 : }%
}%
{}%
}
```

Bool HTMLDebugComments Add comments in HTML about closing s, sections, etc.
Default: false

```
\newbool{HTMLDebugComments}
\boolfalse{HTMLDebugComments}
```

If \tracinglwarp, show where preamble hooks occur:

```
\AfterEndPreamble{
\LWR@traceinfo{AfterEndPreamble}
}

\AtBeginDocument{
\LWR@traceinfo{AtBeginDocument}
}

\end{warpall}
```

30 HTML-conversion output modifications

These booleans modify the HTML output in various ways to improve conversion to EPUB or word processor imports.

for HTML & PRINT:

```
\begin{warpall}
```

30.1 User-level controls

Bool FormatEPUB Changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
Default: false

```
\newbool{FormatEPUB}
\boolfalse{FormatEPUB}
```

Bool FormatWP Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.
Default: false

```
\newbool{FormatWP}
\boolfalse{FormatWP}
```

Bool WPMarkFloats Adds
Default: false

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions. [14]

```
\newbool{WPMarkFloats}
\boolfalse{WPMarkFloats}
```

Bool `WPMarkMinipages` Default: `false`

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

```
\newbool{WPMarkMinipages}
\boolfalse{WPMarkMinipages}
```

Bool `WPMarkTOC` Default: `true`

While formatting for word processors, adds

```
=== table of contents ===
```

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set `false`, the actual TOC is printed instead.

```
\newbool{WPMarkTOC}
\booltrue{WPMarkTOC}
```

Bool `WPMarkLOFT` Default: `false`

While formatting for word processors, adds

```
=== list of figures ===    and/or
=== list of tables ===
```

where each of these lists would have been. This helps identify where to insert the actual lists.

[14] Perhaps some day word processors will have HTML import options for identifying `<figure>` and `<figcaption>` tags for figures and tables.

If set `false`, the actual lists are printed instead.

```
\newbool{WPMarkLOFT}
\boolfalse{WPMarkLOFT}
```

Bool `WPMarkMath` While formatting for word processors, prints math as LaTeX code instead of creating SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.
Default: `false`

```
\newbool{WPMarkMath}
\boolfalse{WPMarkMath}
```

Bool `WPTitleHeading` While formatting for word processors, `true` sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to `false` to cause the title to be plain text, and the section headings to begin at **Heading 1**.
Default: `false`

See table 6 on page 123.

```
\newbool{WPTitleHeading}
\boolfalse{WPTitleHeading}
```

```
\end{warpall}
```

30.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If `WPTitleHeading` is true, adjust so that `part` is **Heading 1**.

If `WPTitleHeading` is false, use `<h1>` for the title, and set `part` to **Heading 2**.

for HTML output:

```
\begin{warpHTML}
```

```
\AtBeginDocument{
\ifbool{FormatWP}{
\@ifundefined{chapter}{
\ifbool{WPTitleHeading}{% part and section starting at h2
\renewcommand*{\LWR@tagtitle}{h1}
\renewcommand*{\LWR@tagtitleend}{/h1}
\renewcommand*{\LWR@tagpart}{h2}
\renewcommand*{\LWR@tagpartend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
```

```
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% WPTitleHeading
{% not WPTitleHeading, part and section starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagsection}{h2}
\renewcommand*{\LWR@tagsectionend}{/h2}
\renewcommand*{\LWR@tagsubsection}{h3}
\renewcommand*{\LWR@tagsubsectionend}{/h3}
\renewcommand*{\LWR@tagsubsubsection}{h4}
\renewcommand*{\LWR@tagsubsubsectionend}{/h4}
\renewcommand*{\LWR@tagparagraph}{h5}
\renewcommand*{\LWR@tagparagraphend}{/h5}
\renewcommand*{\LWR@tagsubparagraph}{h6}
\renewcommand*{\LWR@tagsubparagraphend}{/h6}
}% not WPTitleHeading
}% chapter undefined
{% chapter defined
\ifbool{WPTitleHeading}{}
{% not WPTitleHeading, part and chapter starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagchapter}{h2}
\renewcommand*{\LWR@tagchapterend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% not WPTitleHeading
}% chapter defined
}{}% FormatWP
}% AtBeginDocument

\end{warpHTML}
```

31 Remembering original formatting macros

for HTML output:

```
\begin{warpHTML}
```

Remember original definitions of formatting commands. Will be changed to HTML commands for most uses. Will be temporarily restored to original meaning inside any lateximage environment. Also nullify unused commands.

```
\LetLtxMacro{\LWR@origtextrm}{\textrm}
\LetLtxMacro{\LWR@origtextsf}{\textsf}
\LetLtxMacro{\LWR@origtexttt}{\texttt}
\LetLtxMacro{\LWR@origtextnormal}{\textnormal}
\LetLtxMacro{\LWR@origtextbf}{\textbf}
\LetLtxMacro{\LWR@origtextmd}{\textmd}
\LetLtxMacro{\LWR@origtextit}{\textit}
\LetLtxMacro{\LWR@origtextsl}{\textsl}
\LetLtxMacro{\LWR@origtextsc}{\textsc}
\LetLtxMacro{\LWR@origtextup}{\textup}
\LetLtxMacro{\LWR@origemph}{\emph}

\LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
\LetLtxMacro{\LWR@origsffamily}{\sffamily}
\LetLtxMacro{\LWR@origttfamily}{\ttfamily}
\LetLtxMacro{\LWR@origbfseries}{\bfseries}
\LetLtxMacro{\LWR@origmdseries}{\mdseries}
\LetLtxMacro{\LWR@origupshape}{\upshape}
\LetLtxMacro{\LWR@origslshape}{\slshape}
\LetLtxMacro{\LWR@origscshape}{\scshape}
\LetLtxMacro{\LWR@origitshape}{\itshape}
\LetLtxMacro{\LWR@origem}{\em}
\LetLtxMacro{\LWR@orignormalfont}{\normalfont}

\let\LWR@origraggedright\raggedright
\let\LWR@origonecolumn\onecolumn

\let\LWR@origsp\sp
\let\LWR@origsb\sb
\LetLtxMacro\LWR@origtextsuperscript\textsuperscript
\LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript
\AtBeginDocument{
\LetLtxMacro\LWR@origtextsubscript\textsubscript
\LetLtxMacro\LWR@orig@textsubscript\@textsubscript
}
\LetLtxMacro\LWR@origunderline\underline

\let\LWR@origscriptsize\scriptsize

\let\LWR@orignewpage\newpage
```

```
727
728 \let\LWR@origpagestyle\pagestyle
729 \let\LWR@origthispagestyle\thispagestyle
730 \LetLtxMacro\LWR@origpagenumbering\pagenumbering
731
732 \LetLtxMacro{\LWR@origminipage}{\minipage}
733 \let\LWR@origendminipage\endminipage
734 \LetLtxMacro{\LWR@origparbox}{\parbox}
735
736 \let\LWR@orignewline\newline
737
738 \LetLtxMacro\LWR@origitem\item
739
740
741 \AtBeginDocument{% in case packages change definition
742 \LetLtxMacro{\LWR@origitemize}{\itemize}
743 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
744 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
745 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
746 \LetLtxMacro{\LWR@origdescription}{\description}
747 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
748 }
749
750 \let\LWR@origpar\par
751
752 \LetLtxMacro{\LWR@origfootnote}{\footnote}
753 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
754
755 \let\LWR@origclearpage\clearpage


756 \end{warpHTML}
```

32 Accents

Native LaTeX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

for HTML output:

```
757 \begin{warpHTML}
```

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
758 \AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only \t needs to be redefined:

```
\LetLtxMacro{\LWR@origt}{\t}
```

For pdfLaTeX, additional work is required:

```
\ifPDFTeX
\LetLtxMacro{\LWR@origequalaccent}{\=}
\LetLtxMacro{\LWR@origdotaccent}{\.}
\LetLtxMacro{\LWR@origu}{\u}
\LetLtxMacro{\LWR@origv}{\v}
\LetLtxMacro{\LWR@origc}{\c}
\LetLtxMacro{\LWR@origd}{\d}
\LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdfLaTeX, Unicode diacritical marks are used:

```
\renewcommand*{\=}[1]{#1\HTMLunicode{0305}}
\renewcommand*{\.}[1]{#1\HTMLunicode{0307}}
\renewcommand*{\u}[1]{#1\HTMLunicode{0306}}
\renewcommand*{\v}[1]{#1\HTMLunicode{030C}}
\renewcommand*{\c}[1]{#1\HTMLunicode{0327}}
\renewcommand*{\d}[1]{#1\HTMLunicode{0323}}
\renewcommand*{\b}[1]{#1\HTMLunicode{0331}}
\fi
```

For all engines, a Unicode diacritical tie is used:

```
\def\LWR@t#1#2{#1\HTMLunicode{0361}#2}
\renewcommand*{\t}[1]{\LWR@t#1}
```

\LWR@restoreorigaccents Called from \restoreoriginalformatting when a lateximage is begun.

```
\ifPDFTeX
\newcommand*{\LWR@restoreorigaccents}{%
\LetLtxMacro{\=}{\LWR@origequalaccent}%
\LetLtxMacro{\.}{\LWR@origdotaccent}%
\LetLtxMacro{\u}{\LWR@origu}%
\LetLtxMacro{\v}{\LWR@origu}%
\LetLtxMacro{\t}{\LWR@origt}%
\LetLtxMacro{\c}{\LWR@origc}%
\LetLtxMacro{\d}{\LWR@origd}%
\LetLtxMacro{\b}{\LWR@origb}%
}
\else% XeLaTeX, LuaLaTeX:
\newcommand*{\LWR@restoreorigaccents}{%
```

```
\LetLtxMacro{\t}{\LWR@origt}%
}
\fi
}% AtBeginDocument

\end{warpHTML}
```

33 Configuration Files

```
\begin{warpprint}
\typeout{lwarp: generating configuration files}
\end{warpprint}
```

33.1 project_html.tex

File `project_html.tex` Used to allow an HTML version of the document to exist alongside the print version.

Only write `\jobname_html.tex` if generating the print version.

```
\begin{warpprint}
\immediate\openout\LWR@quickfile=\jobname_html.tex
\immediate\write\LWR@quickfile{%
\detokenize{\PassOptionsToPackage}%
{warpHTML,BaseJobname=\jobname}{lwarp}%
}
\immediate\write\LWR@quickfile{%
\detokenize{\input}\string{\jobname.tex\string }%
}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

33.2 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the lwarp package when executing
`pdflatex <project.tex>`,
or similar for `xelatex` or `lualatex`, in print-document generation mode, which is the default unless the `warpHTML` option is given. `lwarpmk.conf` is then used by the utility `lwarpmk`.

An example `lwarpmk.conf`:

```
opsystem = "Unix"   -- or "Windows"
latexname = "pdflatex"   -- or "lualatex" or "xelatex"
```

```
sourcename = "projectname"   -- your .tex source
homehtmlfilename = "index"   -- or "projectname"
htmlfilename = ""   -- or "projectname" if numbered HTML files
```

for PRINT output:

```
\begin{warpprint}
\ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
\immediate\openout\LWR@quickfile=lwarpmk.conf
\ifbool{usingOSWindows}{
\immediate\write\LWR@quickfile{opsystem = "Windows"}
}{
\immediate\write\LWR@quickfile{opsystem = "Unix"}
}
\ifPDFTeX
\immediate\write\LWR@quickfile{latexname = "pdflatex"}
\fi
\ifXeTeX
\immediate\write\LWR@quickfile{latexname = "xelatex"}
\fi
\ifLuaTeX
\immediate\write\LWR@quickfile{latexname = "lualatex"}
\fi
\immediate\write\LWR@quickfile{sourcename = "\jobname"}
\immediate\write\LWR@quickfile{%
homehtmlfilename = "\HomeHTMLFilename"%
}
\immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
\immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
\immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
\immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

33.3 project.lwarpmkconf

File `project.lwarpmkconf` A project-specific configuration file for `lwarpmk`.

```
\begin{warpprint}
\ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
\immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
\ifbool{usingOSWindows}{
\immediate\write\LWR@quickfile{opsystem = "Windows"}
}{
\immediate\write\LWR@quickfile{opsystem = "Unix"}
}
\ifPDFTeX
\immediate\write\LWR@quickfile{latexname = "pdflatex"}
```

```
\fi
\ifXeTeX
\immediate\write\LWR@quickfile{latexname = "xelatex"}
\fi
\ifLuaTeX
\immediate\write\LWR@quickfile{latexname = "lualatex"}
\fi
\immediate\write\LWR@quickfile{sourcename = "\jobname"}
\immediate\write\LWR@quickfile{%
homehtmlfilename = "\HomeHTMLFilename"%
}
\immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
\immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
\immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
\immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

33.4 lwarp.css

File `lwarp.css` This is the base CSS layer used by lwarp.

This must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp.css}
/*
  CSS stylesheet for the LaTeX lwarp package
  Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
*/


/* a fix for older browsers: */
header, section, footer, aside, nav, main,
    article, figure { display: block; }


A:link {color:#000080 ; text-decoration: none ; }
A:visited {color:#800000 ; }
A:hover {color:#000080 ; text-decoration: underline ;}
A:active {color:#800000 ; }

a.tocpart {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
a.tocchapter {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
```

```
a.tocsection {display: inline-block ; margin-left: 1em ;
    text-indent: -.5em ; font-weight: bold ; }
a.tocsubsection {display: inline-block ; margin-left: 2em ;
    text-indent: -.5em ; }
a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
    text-indent: -.5em ; }
a.tocparagraph {display: inline-block ; margin-left: 4em ;
    text-indent: -.5em ; }
a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
    text-indent: -.5em ; }
a.tocfigure {margin-left: 0em}
a.tocsubfigure {margin-left: 2em}
a.toctable {margin-left: 0em}
a.tocsubtable {margin-left: 2em}
a.toctheorem {margin-left: 0em}
a.toclstlisting {margin-left: 0em}

body {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
        "Lucida Bright", Georgia, serif;
    background: #FAF7F4 ;
    color: black ;
    margin:0em ;
    padding:0em ;
    font-size: 100% ;
    line-height: 1.2 ;
}

p {margin: 1.5ex 0em 1.5ex 0em ;}
table p {margin: .5ex 0em .5ex 0em ;}

/* Holds a section number to add space between it and the name */
span.sectionnumber { margin-right: 0em }

/* Inserted in front of index lines */
span.indexitem {margin-left: 0em}
span.indexsubitem {margin-left: 2em}
span.indexsubsubitem {margin-left: 4em}

div.hidden, span.hidden { display: none ; }

kbd {
    font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
    font-size: 100% ;
}

pre { padding: 3pt ; }

```

```
span.strong { font-weight: bold; }

span.textmd { font-weight: normal; }

span.textsc { font-variant: small-caps; }

span.textsl { font-style: oblique; }

span.textup { font-variant: normal; }

span.textrm {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
    "Lucida Bright", Georgia, serif;
}

span.textsf {
     font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
}

span.textcircled { border: 1px solid black ; border-radius: 1ex ; }

span.underline {
    text-decoration: underline ;
    text-decoration-skip ;
}


/* For realscripts */
.supsubscript {
    display: inline-block;
    text-align:left ;
}

.supsubscript sup,
.supsubscript sub {
    position: relative;
    display: block;
    font-size: .5em;
    line-height: 1;
}

.supsubscript sup {
    top: .5em;
}

.supsubscript sub {
    top: .5em;
}

```

```
span.attribution {
  margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
}

span.citetitle {
  margin-left: 1em ; font-size: 80% ; font-style: oblique;
}

span.poemtitle {
  font-size: 120% ; font-weight: bold;
}

pre.tabbing {
    font-family: "Linux Libertine Mono O", "Lucida Console",
        "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
        "Liberation Mono", "FreeMono", "Andale Mono",
        "Nimbus Mono L", "Courier New", monospace;
}

blockquote {
    margin-left: 0px ;
    margin-right: 0px ;
}

/* quotchap is for the quotchap package */
div.quotchap {
  font-style: oblique ;
  overflow-x: auto ;
  margin-left: 2em ;
  margin-right: 2em ;
}

blockquote p, div.quotchap p {
  line-height: 1.5;
    text-align: left ;
    font-size: .85em ;
    margin-left: 3em ;
    margin-right: 3em ;
}

/* qauthor is for the quotchap package */
div.qauthor {
  display: block ;
  text-align: right ;
  margin-left: auto ;
  margin-right: 2em ;
  font-size: 80% ;
  font-variant: small-caps;
}

```

```
1036 div.qauthor p {
1037   text-align: right ;
1038 }
1039
1040 blockquotation {
1041     margin-left: 0px ;
1042     margin-right: 0px ;
1043 }
1044
1045 blockquotation p {
1046   line-height: 1.5;
1047     text-align: left ;
1048     font-size: .85em ;
1049     margin-left: 3em ;
1050     margin-right: 3em ;
1051 }
1052
1053 div.epigraph, div.dictum {
1054   line-height: 1.2;
1055     text-align: left ;
1056     padding: 3ex 1em 0ex 1em ;
1057 /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1058     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1059 /*    margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1060     font-size: .85em ;
1061     max-width: 27em ;
1062 }
1063
1064
1065
1066 div.epigraphsource, div.dictumauthor {
1067     text-align:right ;
1068     margin-left:auto ;
1069 /*     max-width: 50% ; */
1070     border-top: 1px solid #A0A0A0 ;
1071     padding-bottom: 3ex ;
1072     line-height: 1.2;
1073 }
1074
1075 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1076 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1077 div.dictumauthor { font-style:italic }
1078
1079
1080 /* lettrine package: */
1081 span.lettrine { font-size: 3ex ; float: left ; }
1082 span.lettrinetext { font-variant: small-caps ; }
1083
1084 /* ulem and soul packages: */
1085 span.uline {
```

```
1086     text-decoration: underline ;
1087     text-decoration-skip ;
1088 }
1089
1090 span.uuline {
1091     text-decoration: underline ;
1092     text-decoration-skip ;
1093     text-decoration-style: double ;
1094 }
1095
1096 span.uwave {
1097     text-decoration: underline ;
1098     text-decoration-skip ;
1099     text-decoration-style: wavy ;
1100 }
1101
1102 span.sout {
1103     text-decoration: line-through ;
1104 }
1105
1106 span.xout {
1107     text-decoration: line-through ;
1108 }
1109
1110 span.dashuline {
1111     text-decoration: underline ;
1112     text-decoration-skip ;
1113     text-decoration-style: dashed ;
1114 }
1115
1116 span.dotuline {
1117     text-decoration: underline ;
1118     text-decoration-skip ;
1119     text-decoration-style: dotted ;
1120 }
1121
1122 span.letterspacing { letter-spacing: .2ex ; }
1123
1124 span.capsspacing {
1125     font-variant: small-caps ;
1126     letter-spacing: .1ex ;
1127 }
1128
1129 span.highlight { background: #F8E800 ; }
1130
1131
1132
1133
1134 html body {
1135   margin: 0 ;
```

```
1136   line-height: 1.2;
1137 }
1138
1139
1140 body div {
1141   margin: 0ex;
1142 }
1143
1144
1145 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1146 {
1147     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1148         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1149         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
1150         "Times New Roman", serif;
1151     font-style: normal ;
1152     font-weight: bold ;
1153     text-align: left ;
1154 }
1155
1156 h1 {    /* title of the entire website, used on each page */
1157     text-align: center ;
1158     font-size: 2.5em ;
1159     padding: .4ex 0em 0ex 0em ;
1160 }
1161 h2 { font-size: 2.25em }
1162 h3 { font-size: 2em }
1163 h4 { font-size: 1.75em }
1164 h5 { font-size: 1.5em }
1165 h6 { font-size: 1.25em }
1166 span.paragraph {font-size: 1em ; font-variant: normal ;
1167     margin-right: 1em ; }
1168 span.subparagraph {font-size: 1em ; font-variant: normal ;
1169     margin-right: 1em ;  }
1170
1171 div.minisec {
1172     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1173         Geneva, Verdana, sans-serif ;
1174     font-style: normal ;
1175     font-weight: bold ;
1176     text-align: left ;
1177 }
1178
1179 /* Title of the file */
1180 h1 {
1181   margin: 0ex 0em 0ex 0em ;
1182   line-height: 1.3;
1183   text-align: center ;
1184 }
1185
```

```
/* Part */
h2 {
  margin: 1ex 0em 1ex 0em ;
  line-height: 1.3;
  text-align: center ;
}

/* Chapter */
h3 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Section */
h4 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Section */
h5 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Sub-Section */
h6 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}


div.titlepage {
  text-align: center ;
}

.footnotes {
    font-size: .85em ;
    margin: 3ex 1em 0ex 1em ;
    padding-bottom: 1ex ;
    border-top: 1px solid silver ;
}

.marginpar, .marginparblock {
    max-width:50%;
    float:right;
    text-align:left;
    margin: 1ex 0.5em 1ex 1em ;
    padding: 1ex 0.5em 1ex 0.5em ;
    font-size: 85% ;
```

```
1236     border-top: 1px solid silver ;
1237     border-bottom: 1px solid silver ;
1238     overflow-x: auto;
1239 }
1240
1241 .marginpar br { margin-bottom: 2ex ; }
1242
1243 div.marginblock, div.marginparblock {
1244     max-width:50%;
1245     float:right;
1246     text-align:left;
1247     margin: 1ex 0.5em 1ex 1em ;
1248     padding: 1ex 0.5em 1ex 0.5em ;
1249     overflow-x: auto;
1250 }
1251
1252 div.marginblock div.minipage,
1253 div.marginparblock div.minipage {
1254     display: block ;
1255     margin: 0pt auto 0pt auto ;
1256 }
1257
1258 div.marginblock div.minipage p ,
1259 div.marginparblock div.minipage p
1260     { font-size: 85%}
1261
1262 div.marginblock br ,
1263 div.marginparblock br
1264     { margin-bottom: 2ex ; }
1265
1266
1267 section.textbody div.footnotes{
1268     margin: 3ex 0em 0ex 0em ;
1269     border-bottom: 2px solid silver ;
1270 }
1271
1272 .footnoteheader {
1273     border-top: 2px solid silver ;
1274     margin-top: 3ex ;
1275     padding-top: 1ex ;
1276     font-weight: bold ;
1277 }
1278
1279 .mpfootnotes {
1280     text-align: left ;
1281     font-size: .85em ;
1282     margin-left: 1em ;
1283     border-top: 1px solid silver ;
1284 }
1285
```

```
/* Remove footnote top border in the title page. */
div.titlepage div.mpfootnotes {
    border-top: none ;
}



ol {
  margin: 1ex 1em 1ex 0em;
  line-height: 1.2;
}

ul, body dir, body menu {
  margin: 3ex 1em 3ex 0em;
  line-height: 1.2;
}

li { margin: 0ex 0em 1ex 0em; }

html {
  margin: 0;
  padding: 0;
}

.programlisting {
  font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
  margin: 1ex 0ex 1ex 0ex ;
  padding: .5ex 0pt .5ex 0pt ;
  overflow-x: auto;
}

section.textbody>pre.programlisting {
border-top: 1px solid silver ;
border-bottom: 1px solid silver ;
}


.inlineprogramlisting {
  font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
        "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
        "Courier New", monospace;
  overflow-x: auto;
}

span.listinglabel {
    display: inline-block ;
    font-size: 70% ;
    width: 4em ;
```

```
1336     text-align: right ;
1337     margin-right: 2em ;
1338 }
1339
1340 div.abstract {
1341   margin: 2em 5% 2em 5% ;
1342   padding: 1ex 1em 1ex 1em ;
1343 /*  font-weight: bold ; */
1344   font-size: 90% ;
1345     text-align: left ;
1346 }
1347
1348 div.abstract dl {line-height:1.5;}
1349 div.abstract dt {color:#304070;}
1350
1351 div.abstracttitle{
1352     font-family: "URW Classico", Optima, "Linux Biolinum O",
1353         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1354         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1355     font-weight:bold;
1356     font-size:1.25em;
1357     text-align: center ;
1358 }
1359
1360 span.abstractrunintitle{
1361     font-family: "URW Classico", Optima, "Linux Biolinum O",
1362         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1363         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1364     font-weight:bold;
1365 }
1366
1367
1368 .verbatim {
1369     overflow-x: auto ;
1370 }
1371
1372 .alltt {
1373     overflow-x: auto ;
1374 }
1375
1376
1377 .bverbatim {
1378     margin: 1ex 0pt 1ex 0pt ;
1379     padding: .5ex 0pt .5ex 0pt ;
1380     overflow-x: auto ;
1381 }
1382
1383 .lverbatim {
1384     margin: 1ex 0pt 1ex 0pt ;
1385     padding: .5ex 0pt .5ex 0pt ;
```

```
    overflow-x: auto ;
}

.fancyvrb {
    font-size:.85em ;
    margin: 3ex 0pt 3ex 0pt
}

.fancyvrblabel {
    font-weight:bold;
    text-align: center ;
}


.verse {
    font-family: "Linux Libertine Mono O", "Lucida Console",
        "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
        "Liberation Mono", "FreeMono", "Andale Mono",
        "Nimbus Mono L", "Courier New", monospace;
    margin-left: 1em ;
}


div.singlespace { line-height: 1.2 ; }
div.onehalfspace { line-height: 1.5 ; }
div.doublespace { line-height: 2 ; }


/* Word processor format output: */
div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}




/* Minipage environments, vertically aligned to top, center, bottom: */
.minipage, .fminipage, .fcolorminipage {
    /* display: inline-block ; */
        /* Mini pages which follow each other will be tiled. */
    margin: .25em .25em .25em .25em;
    padding: .25em .25em .25em .25em;
    display: inline-flex;
    flex-direction: column ;
    overflow: auto;
}

/* Paragraphs in the flexbox did not collapse their margins. */
/* Have not yet researched this. */
.minipage p {margin: .75ex 0em .75ex 0em ;}
```

```
1436
1437 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1438 .fcolorbox .minipage, .fcolorboxBlock .minipage
1439     {border: none ; background: none;}
1440
1441 .fbox, .fboxBlock { border: 1px solid black ; }
1442
1443 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1444 .fminipage, .fcolorminipage
1445     {display: inline-block}
1446
1447 .shadowbox, .shabox {
1448    border: 1px solid black;
1449     box-shadow: 3px 3px 3px #808080 ;
1450      border-radius: 0px ;
1451     padding: .4ex .3em .4ex .3em ;
1452     margin: 0pt .3ex 0pt .3ex ;
1453   display: inline-block ;
1454 }
1455
1456 .doublebox {
1457    border: 3px double black;
1458      border-radius: 0px ;
1459     padding: .4ex .3em .4ex .3em ;
1460     margin: 0pt .3ex 0pt .3ex ;
1461   display: inline-block ;
1462 }
1463
1464 .ovalbox, .Ovalbox {
1465    border: 1px solid black;
1466      border-radius: 1ex ;
1467     padding: .4ex .3em .4ex .3em ;
1468     margin: 0pt .3ex 0pt .3ex ;
1469   display: inline-block ;
1470 }
1471
1472 .Ovalbox { border-width: 2px ; }
1473
1474 .framebox {
1475    border: 1px solid black;
1476      border-radius: 0px ;
1477     padding: .3ex .2em 0ex .2em ;
1478     margin: 0pt .1ex 0pt .1ex ;
1479   display: inline-block ;
1480 }
1481
1482
1483 .mdframed {
1484 /*     padding: 0ex ; */
1485 /*    border: 1px solid blafck; */
```

```
/*      border-radius: 0px ; */
    padding: 0ex ;
    margin: 3ex 5% 3ex 5% ;
/*   display: inline-block ; */
}

.mdframed p { padding: 0ex .5em 0ex .5em ; }

.mdframed dl { padding: 0ex .5em 0ex .5em ; }

.mdframedtitle {
    padding: .5em ;
    display: block ;
    font-size: 130% ;
    margin-bottom: 1ex ;
}

.mdframedsubtitle {
    padding: 0ex .5em 0ex .5em ;
    display: block ;
    font-size: 115% ;
}

.mdframedsubsubtitle {
    padding: 0ex .5em 0ex .5em ;
    display: block ;
}

.mdtheorem {
    padding: 0ex .5em 0ex .5em ;
    margin: 3ex 5% 3ex 5% ;
/*   display: inline-block ; */
}


/* framed package */
.framed, pre.boxedverbatim, fcolorbox {
    margin: 3ex 0em 3ex 0em ;
   border: 1px solid black;
     border-radius: 0px ;
    padding: .3ex 1em 0ex 1em ;
  display: block ;
}

.shaded {
    margin: 3ex 0em 3ex 0em ;
    padding: .3ex 1em .3ex 1em ;
    display: block ;
}

```

```
.snugframed {
    margin: 3ex 0em 3ex 0em ;
   border: 1px solid black;
     border-radius: 0px ;
  display: block ;
}

.framedleftbar {
    margin: 3ex 0em 3ex 0em ;
   border-left: 3pt solid black;
     border-radius: 0px ;
    padding: .3ex .2em .3ex 1em ;
  display: block ;
}

.framedtitle {
    margin: 0em ;
    padding: 0em ;
    font-size: 130%
}

.framedtitle p { padding: .3em }



dl {
  margin: 1ex 2em 1ex 0em;
  line-height: 1.3;
}

dl dt {
    margin-top: 1ex;
    margin-left: 1em ;
    font-weight: bold;
}

dl dd p { margin-top: 0em; }


nav {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    margin-bottom: 4ex ;
}

nav p {
    line-height: 1.2 ;
    margin-top:.5ex ;
    margin-bottom:.5ex;
```

```
    font-size: .9em ;
}



img, img.hyperimage, img.borderimage {
    max-width: 600px;
    border: 1px solid silver;
    box-shadow: 3px 3px 3px #808080 ;
    padding: .5% ;
    margin: .5% ;
    background: none ;
}

img.inlineimage{
    padding: 0px ;
    box-shadow: none ;
    border: none ;
    background: none ;
    margin: 0px ;
    display: inline-block ;
    border-radius: 0px ;
}

img.logoimage{
    max-width: 300px ;
    box-shadow: 3px 3px 3px #808080 ;
    border: 1px solid black ;
    background:none ;
    padding:0 ;
    margin:.5ex ;
    border-radius: 10px ;
}


.section {
/*
    To have each section float relative to each other:
*/
/*
    display: block ;
    float: left ;
    position: relative ;
    background: white ;
    border: 1px solid silver ;
    padding: .5em ;
*/
    margin: 0ex .5em 0ex .5em ;
    padding: 0 ;
}
```

```
1636
1637
1638 figure {
1639     margin: 3ex auto 3ex auto ;
1640     padding: 1ex 1em 1ex 1em ;
1641     overflow-x: auto ;
1642 }
1643
1644
1645 /* To automatically center images in figures: */
1646 /*
1647 figure img.inlineimage {
1648     margin: 0ex auto 0ex auto ;
1649     display: block ;
1650 }
1651 */
1652
1653 /* To automatically center minipages in figures: */
1654 /*
1655 figure div.minipage, figure div.minipage div.minipage {
1656     margin: 1ex auto 1ex auto ;
1657     display: block ;
1658 }
1659 */
1660
1661 figure div.minipage p { font-size: 85% ; }
1662
1663 figure.subfigure, figure.subtable  {
1664     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1665 }
1666
1667 figcaption .minipage { margin:0 ; padding: 0 }
1668
1669 div.minipage figure { border: none ; box-shadow: none ; }
1670
1671 div.floatrow { text-align: center; }
1672
1673 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1674
1675 div.floatfoot { font-size: .85em ;
1676     border-top: 1px solid silver ; line-height: 1.2 ; }
1677
1678 figcaption , .lstlistingtitle {
1679     font-size: .85em ;
1680     text-align: center ;
1681     font-weight: bold ;
1682     margin-top: 1ex ;
1683     margin-bottom: 1ex ;
1684 }
1685
```

```
figure.subfigure figcaption, figure.subtable figcaption  {
    border-bottom: none ; background: none ;
}

div.nonfloatcaption {
    margin: 1ex auto 1ex auto ;
    font-size: .85em ;
    text-align: center ;
    font-weight: bold ;
}

/* For a \RawCaption inside a minipage inside a figure's floatrow: */
figure div.floatrow div.minipage figcaption {
    border: none ;
    background: none ;
}


table {
    margin: 1ex auto 1ex auto ;
    border-collapse: separate ;
    border-spacing: 0px ;
    line-height: 1.3 ;
    }

tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \hline */

tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \toprule, \bottomrule */

td {padding: 1ex .5em 1ex .5em ;}

table td.tdl { text-align: left ; vertical-align: middle ; }
table td.tdc { text-align: center ; vertical-align: middle ; }
table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
table td.tdbang { text-align: center ; vertical-align: middle ; }
table td.tdr { text-align: right ; vertical-align: middle ; }
table td.tdp { text-align: left ; vertical-align: bottom ; }
table td.tdm { text-align: left ; vertical-align: middle ; }
table td.tdb { text-align: left ; vertical-align: top ; }
table td.tdP { text-align: center ; vertical-align: bottom ; }
table td.tdM { text-align: center ; vertical-align: middle ; }
table td.tdB { text-align: center ; vertical-align: top ; }

table td.tvertbarl { border-left: 1px solid black }
table td.tvertbarr { border-right: 1px solid black }


/* for cmidrules: */
```

```
table td.tdrule {
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulel {
    border-top-left-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdruler {
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulelr {
    border-top-left-radius:.5em ;
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}


/* Margins of paragraphs inside table cells: */
td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }

td.tdp , td.tdprule , td.tdP , td.tdPrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdm , td.tdmrule , td.tdM , td.tdMrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdb , td.tdbrule , td.tdB , td.tdBrule
    { padding: 0ex .5em 0ex .5em ; }


/* table notes: */
.tnotes {
    margin: 0ex 5% 1ex 5% ;
    padding: 0.5ex 1em 0.5ex 1em;
    font-size:.85em;
    text-align: left ;
}

.tnotes dl dt p {margin-bottom:0px;}

.tnoteitemheader {margin-right: 1em;}


```

```
/* for bigdelim */
.ldelim, .rdelim { font-size: 200% }


/* center, flushleft, flushright environments */
div.center{text-align:center;}
div.center table {margin-left:auto;margin-right:auto;}
div.flushleft{text-align:left;}
div.flushleft table {margin-left:0em ; margin-right:auto;}
div.flushright{text-align:right;}
div.flushright table {margin-left:auto ; margin-right: 0em ;}


/* Fancybox */
div.Btrivlist table tr td { padding: .2ex 0em ; }


/* program listing callouts: */
span.callout {
     font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    border-radius: .5em;
    background-color:black;
    color:white;
    padding:0px .25em 0px .25em;
    margin: 0 ;
    font-weight: bold;
    font-size:.72em ;
}

div.programlisting pre.verbatim span.callout{
    font-size: .85em ;
}









div.published
{
    text-align: center ;
    font-variant: normal ;
    font-style: italic ;
    font-size: 1em ;
    margin: 3ex 0em 3ex 0em ;
}

div.subtitle
{
    text-align: center ;
```

```
1836     font-variant: normal ;
1837     font-style: italic ;
1838     font-size: 1.25em ;
1839     margin: 3ex 0em 3ex 0em ;
1840 }
1841
1842 div.subtitle p { margin: 1ex ; }
1843
1844 div.author
1845 {
1846     font-variant: normal ;
1847     font-style: normal ;
1848     font-size: 1em ;
1849     margin: 3ex 0em 3ex 0em ;
1850 }
1851
1852 div.oneauthor {
1853     display: inline-block ;
1854     margin: 3ex 1em 0ex 1em ;
1855 }
1856
1857 /*
1858 div.author table {
1859     margin: 3ex auto 0ex auto ;
1860     background: none ;
1861 }
1862
1863 div.author table tbody tr td { padding: .25ex ; }
1864 */
1865
1866 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1867
1868 div.titledate {
1869     text-align: center ;
1870     font-size: .85em ;
1871     font-style: italic;
1872     margin: 6ex 0em 6ex 0em ;
1873 }
1874
1875
1876 nav.topnavigation{
1877     text-align: left ;
1878     padding: 0.5ex 1em 0.5ex 1em ;
1879 /*     margin: 2ex 0em 3ex 0em ; */
1880     margin: 0 ;
1881     border-bottom: 1px solid silver ;
1882     border-top: 1px solid silver ;
1883     clear:right ;
1884 }
1885
```

```
nav.botnavigation{
    text-align: left ;
    padding: 0.5ex 1em 0.5ex 1em ;
/*     margin: 3ex 0em 2ex 0em ; */
    margin: 0 ;
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    clear:right ;
}


header{
    line-height: 1.2 ;
    font-size: 1em ;
/*     border-bottom: 2px solid silver ; */
    margin: 0px ;
    padding: 0ex 1em 0ex 1em ;
    text-align:center ;
}

header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}


footer{
    font-size: .85em ;
    line-height: 1.2 ;
    margin-top: 1ex ;
    border-top: 2px solid silver ;
    padding: 2ex 1em 2ex 1em ;
    clear:right ;
    text-align:left ;
}


a.linkhome { font-weight:bold ; font-size: 1em ;}


div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }

img.lateximage{
    padding: 0px 0px 0px 0px ;
    box-shadow: none ;
    border: none ;
    background: none ;
    margin: 0px 0px -.15ex 0px ;
        /* pdfcrop leaves a slight margin, adjust to baseline */
    max-width: 100% ;
    border-radius: 0ex ;
    border: none ;
}
```

```
1936
1937
1938
1939 nav.sidetoc {
1940     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1941         "Lucida Bright", Georgia, serif;
1942     float:right ;
1943     width: 20%;
1944     border-left: 1px solid silver;
1945     border-top: 1px solid silver;
1946     border-bottom: 1px solid silver;
1947 /*     border-top: 2px solid #808080 ; */
1948     background: #FAF7F4 ;
1949     padding: 2ex 0em 2ex 1em ;
1950     margin: 0ex 0em 2ex 1em ;
1951     font-size:.9em ;
1952     border-radius: 20px 0px 0px 20px ;
1953     }
1954
1955 div.sidetoccontents {
1956 /*     border-top: 1px solid silver ; */
1957     overflow-y: auto ;
1958     width: 100% ;
1959     text-align: left ;
1960 }
1961
1962
1963 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1964     text-indent: 0 ; }
1965
1966 nav.sidetoc p a {color:black ; font-size: .7em ;}
1967
1968 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1969     border-bottom: 1px solid silver ;    }
1970
1971 nav.sidetoc a:hover {text-decoration: underline ; }
1972
1973
1974
1975 section.textbody { margin: 0ex 1em 0ex 1em ;}
1976
1977
1978 div.multicolsheading { -webkit-column-span: all;
1979     -moz-column-span: all; column-span: all; }
1980 div.multicols { -webkit-columns: 3 380px ;
1981     -moz-columns: 3 380px ; columns: 3 380px ; }
1982 div.multicols p {margin-top: 0ex}
1983
1984
1985
```

```
/* Used to support algorithmicx: */
span.floatright { float: right ; }



/* Native LaTeX theorems: */

.theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
.theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }


/* theorem, amsthm, and ntheorem packages */

span.theoremheader,
span.theoremheaderplain,
span.theoremheaderdefinition,
span.theoremheaderbreak,
span.theoremheadermarginbreak,
span.theoremheaderchangebreak,
span.theoremheaderchange,
span.theoremheadermargin
{
    font-style:normal ; font-weight: bold ; margin-right: 1em ;
}

span.amsthmnameplain,
span.amsthmnamedefinition,
span.amsthmnumberplain,
span.amsthmnumberdefinition
{
    font-style:normal ; font-weight: bold ;
}


span.amsthmnameremark,
span.amsthmnumberremark
{font-style:italic ; font-weight: normal ; }


span.amsthmnoteplain,
span.amsthmnotedefinition
{font-style:normal ;}


span.theoremheaderremark,
span.theoremheaderproof,
span.amsthmproofname
{font-style:italic ; font-weight: normal ; margin-right: 1em ; }

```

```
span.theoremheadersc
{
    font-style:normal ;
    font-variant: small-caps ;
    font-weight: normal ;
    margin-right: 1em ;
}

.theoremendmark {float:right}

div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
div.theorembodybreak, div.theorembodynonumberbreak,
div.theorembodymarginbreak,
div.theorembodychangebreak,
div.theorembodychange,
div.theorembodymargin
{
    font-style:italic;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
div.theorembodyplainupright, nonumberplainuprightsc,
div.amsthmbodydefinition, div.amsthmbodyremark,
div.amsthmproof
{
    font-style: normal ;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

span.amsthmnoteremark {}



/*
For CSS LaTeX and related logos:
Based on:
http://edward.oconnor.cx/2007/08/tex-poshlet
http://nitens.org/taraborelli/texlogo
*/

.latexlogofont {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: normal ;
}

.latexlogo {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
```

```
2086     letter-spacing: .03em ;
2087     font-size: 1.1em;
2088 }
2089
2090 .latexlogo sup {
2091   text-transform: uppercase;
2092   letter-spacing: .03em ;
2093   font-size: 0.85em;
2094   vertical-align: 0.15em;
2095   margin-left: -0.36em;
2096   margin-right: -0.15em;
2097 }
2098
2099 .latexlogo sub {
2100   text-transform: uppercase;
2101   vertical-align: -0.5ex;
2102   margin-left: -0.1667em;
2103   margin-right: -0.125em;
2104   font-size: 1em;
2105 }
2106
2107 .xetexlogo {
2108     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2109         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2110     letter-spacing: .03em ;
2111     font-size: 1.1em;
2112 }
2113
2114 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2115 .xetexlogo sub {
2116   text-transform: uppercase;
2117   vertical-align: -0.5ex;
2118   margin-left: -0.0667em;
2119   margin-right: -0.2em;
2120   font-size: 1em;
2121   letter-spacing: .03em ;
2122 }
2123
2124 /* A large gap between Xe and LaTeX v.s. TeX: */
2125 .xelatexlogo sub {
2126   text-transform: uppercase;
2127   vertical-align: -0.5ex;
2128   margin-left: -0.0667em;
2129   margin-right: -.05em;
2130   font-size: 1em;
2131   letter-spacing: .03em ;
2132 }
2133
2134 .amslogo {
2135     font-family: "TeXGyreChorus","URW Chancery L",
```

```
2136         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
2137         "Linux Libertine O", "Nimbus Roman No 9 L",  "FreeSerif",
2138         "Hoefler Text", Times, "Times New Roman", serif;
2139   font-style: italic;
2140 }
2141
2142 .lyxlogo {
2143     font-family: "URW Classico", Optima, "Linux Biolinum O",
2144         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2145         Verdana, sans-serif ;
2146 }
2147
2148
2149
2150
2151 /* Only display top and bottom navigation if a small screen: */
2152 /* Hide the sidetoc if a small screen: */
2153 nav.topnavigation { display:none; }
2154 nav.botnavigation { display:none; }
2155
2156 @media screen and (max-width: 45em) {
2157 /*     nav.sidetoc {display:none;} */
2158     nav.sidetoc {
2159         float: none ;
2160         width: 100% ;
2161         margin: 5ex 0px 5ex 0px ;
2162         padding: 0 ;
2163         border-radius: 0 ;
2164         border-bottom: 1px solid black ;
2165         border-top: 1px solid black ;
2166         box-shadow: none ;
2167     }
2168 /*     nav.topnavigation { display:block } */
2169     nav.botnavigation { display:block }
2170     .marginpar {
2171         max-width: 100%;
2172         float: none;
2173         display:block ;
2174         margin: 1ex 1em 1ex 1em ;
2175     }
2176 }
2177
2178 @media print {
2179     body {
2180         font-family: "Linux Libertine O",
2181         "DejaVu Serif", "Bitstream Vera Serif",
2182         "Liberation Serif", "Nimbus Roman No 9 L",
2183         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2184     }
2185     nav.sidetoc { display:none; }
```

```
    nav.topnavigation { display: none; }
    nav.botnavigation { display: none; }
}

@media handheld {
    nav.sidetoc { display:none; }
    nav.topnavigation { display:block }
    nav.botnavigation { display:block }
}

@media projection {
    nav.sidetoc { display:none; }
    nav.topnavigation { display:block }
    nav.botnavigation { display:block }
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

33.5 lwarp_sagebrush.css

File `lwarp_sagebrush.css` An optional CSS which may be used for a semi-modern appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_sagebrush.css}
@import url("lwarp.css") ;


A:link {color:#105030 ; text-decoration: none ; }
A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}



h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif",
        "Nimbus Roman No 9 L",  "FreeSerif",
        "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: small-caps ;
    font-weight: normal ;
    color: #304070 ;
```

```
2225     text-shadow: 2px 2px 3px #808080;
2226 }
2227
2228 h1 {    /* title of the entire website, used on each page */
2229     font-variant: small-caps ;
2230     color: #304070 ;
2231     text-shadow: 2px 2px 3px #808080;
2232     background-color: #F7F7F0 ;
2233     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2234 }
2235
2236 h1 {
2237   border-bottom: 1px solid #304070;
2238   border-top: 2px solid #304070;
2239 }
2240
2241 h2 {
2242   border-bottom: 1px solid #304070;
2243   border-top: 2px solid #304070;
2244     background-color: #F7F7F0 ;
2245     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2246 }
2247
2248
2249
2250 div.abstract {
2251     background: #f5f5eb ;
2252     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2253
2254   border: 1px solid silver;
2255     border-radius: 1em ;
2256 }
2257
2258 div.abstract dl {line-height:1.5;}
2259 div.abstract dt {color:#304070;}
2260
2261 div.abstracttitle{
2262     font-family: "URW Classico", Optima, "Linux Biolinum O",
2263         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2264         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2265     font-weight:bold;
2266     font-variant: small-caps ;
2267     font-size:1.5em;
2268     border-bottom: 1px solid silver ;
2269     color: #304070 ;
2270     text-align: center ;
2271     text-shadow: 1px 1px 2px #808080;
2272 }
2273
2274 span.abstractrunintitle{
```

```
2275     font-family: "URW Classico", Optima, "Linux Biolinum O",
2276         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2277         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2278     font-weight:bold;
2279 }
2280
2281
2282 div.epigraph, div.dictum {
2283     background: #f5f5eb ;
2284     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2285
2286     border: 1px solid silver ;
2287     border-radius: 1ex ;
2288     box-shadow: 3px 3px 3px #808080 ;
2289 }
2290
2291
2292 .example {
2293     background-color: #f5f5eb ;
2294     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2295
2296 }
2297
2298 div.exampletitle{
2299     font-family: "URW Classico", Optima, "Linux Biolinum O",
2300         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2301         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2302     font-weight:bold;
2303     font-variant: small-caps ;
2304     border-bottom: 1px solid silver ;
2305     color: #304070 ;
2306     text-align: center ;
2307     text-shadow: 1px 1px 2px #808080;
2308 }
2309
2310
2311 .sidebar {
2312     background-color: #f5f5eb ;
2313     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2314
2315 }
2316
2317 div.sidebartitle{
2318     font-family: "URW Classico", Optima, "Linux Biolinum O",
2319         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2320         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2321     font-weight:bold;
2322     font-variant: small-caps ;
2323     border-bottom: 1px solid silver ;
2324     color: #304070 ;
```

```
2325     text-align: center ;
2326     text-shadow: 1px 1px 2px #808080;
2327 }
2328
2329
2330 .fancyvrblabel {
2331     font-family: "URW Classico", Optima, "Linux Biolinum O",
2332         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2333         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2334     font-weight:bold;
2335     font-variant: small-caps ;
2336     font-size: 1.5em ;
2337     color: #304070 ;
2338     text-align: center ;
2339     text-shadow: 1px 1px 2px #808080;
2340 }
2341
2342
2343
2344 div.minipage {
2345     background-color: #eeeee7 ;
2346     border: 1px solid silver ;
2347     border-radius: 1ex ;
2348 }
2349
2350 section.textbody > div.minipage {
2351     box-shadow: 3px 3px 3px #808080 ;
2352 }
2353
2354 div.fboxBlock div.minipage { box-shadow: none ; }
2355
2356 .framed .minipage , .framedleftbar .minipage {
2357     border: none ;
2358     background: none ;
2359     padding: 0ex ;
2360     margin: 0ex ;
2361 }
2362
2363 figure.figure .minipage, figcaption .minipage { border: none; }
2364
2365 div.marginblock div.minipage ,
2366 div.marginparblock div.minipage
2367     { border: none; }
2368
2369 figure , div.marginblock {
2370     background-color: #eeeee7 ;
2371     border: 1px solid silver ;
2372     border-radius: 1ex ;
2373     box-shadow: 3px 3px 3px #808080 ;
2374 }
```

```
2375
2376 figure figure {
2377     border: 1px solid silver ;
2378     margin: 0em ;
2379     box-shadow: none ;
2380 }
2381
2382 /*
2383 figcaption {
2384     border-top: 1px solid silver ;
2385     border-bottom: 1px solid silver ;
2386     background-color: #e8e8e8 ;
2387 }
2388 */
2389
2390
2391 div.table {
2392     box-shadow: 3px 3px 3px #808080 ;
2393 }
2394
2395 /*
2396 .tnotes {
2397     background: #e8e8e8;
2398     border: 1px solid silver;
2399 }
2400 */
2401
2402
2403 nav.topnavigation{
2404     background-color: #b0b8b0 ;
2405     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2406 }
2407
2408 nav.botnavigation{
2409     background-color: #b0b8b0 ;
2410     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2411 }
2412
2413
2414
2415 header{
2416     background-color: #F7F7F0 ;
2417     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2418 }
2419
2420 footer{
2421     background-color: #F7F7F0 ;
2422     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2423 }
2424
```

```
2425
2426
2427 nav.sidetoc {
2428     background-color: #F7F7F0 ;
2429     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2430     box-shadow: 3px 3px 3px #808080 ;
2431     border-radius: 0px 0px 0px 20px ;
2432     }
2433
2434 div.sidetoctitle {color: #304070 ; }
2435
2436 nav.sidetoc a:hover {
2437     color:#006000 ;
2438     text-decoration: none ;
2439     text-shadow:0px 0px 2px #a0a0a0;
2440 }
2441
2442
2443 @media screen and (max-width: 45em) {
2444     nav.sidetoc { border-radius: 0 ; }
2445 }
2446
2447
2448 \end{filecontents*}
2449 % \end{Verbatim}% for syntax highlighting
2450 \end{warpprint}
```

33.6 lwarp_formal.css

File `lwarp_formal.css` An optional CSS which may be used for a more formal appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
2451 \begin{warpprint}
2452 \begin{filecontents*}{lwarp_formal.css}
2453 @import url("lwarp.css") ;
2454
2455
2456
2457 A:link {color:#802020 ; text-decoration:none; }
2458 A:visited {color:#802020 ; text-shadow:none ;}
2459 A:hover {color:#400000 ; text-shadow:none ;}
2460 A:active {color:#C00000 ; text-shadow:none ;}
2461
2462
2463 body {
```

```
2464     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2465         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2466         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2467         "Times New Roman", serif;
2468     background: #fffcf5;
2469 }
2470
2471 span.textrm {
2472     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2473         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2474         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2475         "Times New Roman", serif;
2476 }
2477
2478 span.textsf {
2479      font-family: "DejaVu Sans", "Bitstream Vera Sans",
2480         Geneva, Verdana, sans-serif ;
2481 }
2482
2483
2484
2485 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2486 {
2487     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2488         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2489         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2490         "Times New Roman", serif;
2491     color: #800000 ;
2492     text-shadow: none ;
2493 }
2494
2495 h1, h2 {
2496     background-color: #fffcf5 ;
2497     background-image: none ;
2498     border-bottom: 1px solid #808080;
2499     border-top: 2px solid #808080;
2500 }
2501
2502 div.abstracttitle {
2503     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2504         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2505         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2506         "Times New Roman", serif;
2507     color: black ;
2508     text-shadow: none ;
2509 }
2510
2511 span.abstractrunintitle {
2512     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2513         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```
2514        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2515        "Times New Roman", serif;
2516    color: black ;
2517    text-shadow: none ;
2518 }
2519
2520 div.abstract { font-size: 100% }
2521
2522 .sidebar {
2523    background: #fffcf5;
2524    background-image: none ;
2525  margin: 2em 5% 2em 5%;
2526  padding: 0.5em 1em;
2527  border: none ;
2528  border-top : 1px solid silver;
2529  border-bottom : 1px solid silver;
2530  font-size: 90% ;
2531 }
2532
2533 div.sidebartitle{
2534    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2535        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2536        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2537        "Times New Roman", serif;
2538    color: #800000 ;
2539    text-shadow: none ;
2540    border: none ;
2541 }
2542
2543 .example {
2544    background: #fffcf5;
2545    background-image: none ;
2546  margin: 2em 5% 2em 5%;
2547  padding: 0.5em 1em;
2548  border: none ;
2549  border-top : 1px solid silver;
2550  border-bottom : 1px solid silver;
2551 }
2552
2553 div.exampletitle{
2554    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2555        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2556        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2557        "Times New Roman", serif;
2558    color: #800000 ;
2559    text-shadow: none ;
2560    border: none ;
2561 }
2562
2563 div.fancyvrblabel{
```

```
2564     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2565         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2566         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2567         "Times New Roman", serif;
2568     color: #800000 ;
2569     text-shadow: none ;
2570     border: none ;
2571 }
2572
2573
2574
2575 .verse {
2576     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2577         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2578         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
2579         "Times New Roman", serif;
2580 }
2581
2582
2583 figure {
2584     margin: 3ex 5% 3ex 5% ;
2585     padding: 1ex 1em 1ex 1em ;
2586     background-color: #fffcf5 ;
2587     overflow-x: auto ;
2588     border: none ;
2589 /*     border-top: 1px solid silver; */
2590 /*     border-bottom: 1px solid silver; */
2591 }
2592
2593
2594 figcaption , .lstlisting {
2595     border: none ;
2596 /*     border-top: 1px solid silver ; */
2597 /*     border-bottom: 1px solid silver ; */
2598     background-color: #fffcf5 ;
2599 }
2600
2601 .tnotes {
2602     background: #fffcf5 ;
2603 }
2604
2605 .theorem {
2606         background: none ;
2607 }
2608
2609 .minipage {
2610     background-color: #fffcf5 ;
2611     border: none ;
2612 }
2613
```

```
div.floatrow figure { border: none ; }

figure figure { border: none ; }


nav.toc, nav.lof, nav.lot, nav.lol {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
}

nav.sidetoc {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
    border-radius: 0px 0px 0px 20px ;
}

div.sidetoctitle{
    color: #800000 ;
}

header{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
    text-align:center ;
}

footer{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
    padding: 2ex 1em 2ex 1em ;
    clear:right ;
    text-align:left ;
}

nav.botnavigation {
    background: #dedcd5 ;
    border-top: 1px solid black ;
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

33.7 sample_project.css

File `sample_project.css` The project-specific CSS file. Use with `\CSSFilename`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{sample_project.css}
/* ( --- Start of project.css --- ) */
/* ( --- A sample project-specific CSS file for lwarp --- ) */

/* Load default lwarp settings: */
@import url("lwarp.css") ;
/* or lwarp_formal.css, lwarp_sagebrush.css */

/* Project-specific CSS setting follow here. */
/* . . . */

/* ( --- End of project.css --- ) */
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

33.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for lwarp.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp.xdy}
(require "tex/inputenc/latin.xdy")
(merge-rule "\\PS *" "Postscript")
(require "texindy.xdy")
(require "page-ranges.xdy")
(require "book-order.xdy")
(require "page-ranges.xdy")
(markup-locref :open "\hyperindexref{" :close "}")
(define-location-class "arabic-page-numbers"
    ("arabic-numbers") :min-range-length 1)
(define-location-class-order ("roman-page-numbers"
                  "arabic-page-numbers"
                  "alpha-page-numbers"
                  "Roman-page-numbers"
```

```
                  "Alpha-page-numbers"
                  "see"
                  "seealso"))
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

33.9 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by lwarp when using MathJax.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_mathjax.txt}
<!-- https://groups.google.com/forum/#!topic/
                                mathjax-users/jUtewUcE2bY -->
<script type="text/x-mathjax-config">
MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
    var seteqsectionDefault = {name: "", num: 0};
    var seteqsections = {}, seteqsection = seteqsectionDefault;
    var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
    var AMS = MathJax.Extension["TeX/AMSmath"];
    TEX.Definitions.Add({
    macros: {
        seteqsection: "mySection",
        seteqnumber: "mySetEqNumber"
    }
    });

    PARSE.Augment({
    mySection: function (name) {
        seteqsection.num = AMS.number;
        var n = this.GetArgument(name);
        if (n === "") {
        seteqsection = seteqsectionDefault;
        } else {
        if (!seteqsections["_"+n])
            seteqsections["_"+n] = {name:n, num:0};
        seteqsection = seteqsections["_"+n];
        }
        AMS.number = seteqsection.num;
    },
    mySetEqNumber: function (name) {
        var n = this.GetArgument(name);
        if (!n || !n.match(/^ *[0-9]+ *$/))
```

```
2730            n = ""; else n = parseInt(n)-1;
2731        <!-- $ syntax highlighting -->
2732        if (n === "" || n < 1)
2733            TEX.Error
2734            ("Argument to "+name+" should be a positive integer");
2735        AMS.number = n;
2736    }
2737    });
2738    MathJax.Hub.Config({
2739    TeX: {
2740        equationNumbers: {
2741        formatTag: function (n)
2742            {return "("+(seteqsection.name+"."+n).replace(/^\./,"")+")"},
2743        formatID: function (n) {
2744            n = (seteqsection.name+'.'+n).replace
2745                (/[:"'<>&]/g,"").replace(/^\./,"");
2746            return 'mjx-eqn-' + n;
2747        }
2748        }
2749    }
2750    });
2751 });
2752 </script>
2753
2754 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2755 <script type="text/x-mathjax-config">
2756  MathJax.Ajax.config.path["Contrib"] =
2757    "https://cdn.mathjax.org/mathjax/contrib";
2758 </script>
2759
2760 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2761
2762 <script type="text/x-mathjax-config">
2763 MathJax.Hub.Config({
2764   extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
2765   jax: ["input/TeX","output/HTML-CSS"],
2766   tex2jax: {
2767        inlineMath: [["$","$"],["\\(","\\)"]] ,
2768        processClass: "tabbing|verse"
2769    },
2770   TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
2771 });
2772 MathJax.Ajax.config.path['siunitx']  = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2773 </script>
2774
2775 <script type="text/x-mathjax-config">
2776 MathJax.Hub.Config({
2777    TeX: {
2778    equationNumbers: {
2779        autoNumber: "AMS"
```

```
    }
    }
});
</script>

<!-- Alternative CDN provider: -->
<script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
</script>

<!-- No longer supported after April 30, 2017: -->
<!--
<script
  src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
</script>
-->

\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

33.10 lwarpmk option

The following is only generated if the `lwarpmk` option was given to lwarp.

```
\begin{LWR@createlwarpmk}
```

Prog `lwarpmk` Creates a local copy of `lwarpmk`:

```
\begin{filecontents*}{lwarpmk.lua}
#!/usr/bin/env texlua

-- Copyright 2016-2018 Brian Dunn

-- Print the usage of the lwarpmk command:

printversion = "v0.46"

function printhelp ()
print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
end

function printusage ()
print ( [[

lwarpmk print [project]: Compile the print version if necessary.
lwarpmk print1 [project]: Forced single compile of the print version.
```

```
lwarpmk printindex [project]: Process the index for the print version.
lwarpmk printglossary [project]: Process the glossary for the print version.
lwarpmk html [project]: Compile the HTML version if necessary.
lwarpmk html1 [project]: Forced single compile of the HTML version.
lwarpmk htmlindex [project]: Process the index for the html version.
lwarpmk htmlglossary [project]: Process the glossary for the html version.
lwarpmk again [project]: Touch the source code to trigger recompiles.
lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
lwarpmk pdftohtml [project]:
    For use with latexmk or a Makefile:
    Convert project_html.pdf to project_html.html and
    individual HTML files.
lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
lwarpmk -h: Print this help message.
lwarpmk --help: Print this help message.

]] )
printconf ()
end

-- Print the format of the configuration file lwarpmk.conf:

function printconf ()
print ( [[
An example lwarpmk.conf or <project>.lwarpmkconf project file:
--
opsystem = "Unix"   (or "Windows")
latexname = "pdflatex"  (or "lualatex", or "xelatex")
sourcename = "projectname"  (the source-code filename w/o .tex)
homehtmlfilename = "index"  (or perhaps the project name)
htmlfilename = ""  (or "projectname" - filename prefix)
latexmk = "false"  (or "true" to use latexmk to build PDFs)
languge = "english"  (use a language supported by xindy)
xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
--
Filenames must contain only letters, numbers, underscore, or dash.
Values must be in "quotes".

]] ) ;
end


-- Split one large sourcefile into a number of files,
-- starting with destfile.
-- The file is split at each occurance of <!--|Start file|newfilename|*

function splitfile (destfile,sourcefile)
print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
local sfile = io.open(sourcefile)
```

```
io.output(destfile)
for line in sfile:lines() do
i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
io.output(newfilename) ;
else -- not a splitpoint
io.write (line .. "\n") ;
end
end -- do
io.close(sfile)
end -- function

-- Incorrect value, so print an error and exit.

function cvalueerror ( line, linenum , cvalue )
    print ( linenum .. " : " .. line ) ;
    print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
    printconf () ;
    os.exit(1) ;
end

-- Load settings from the project's "lwarpmk.conf" file:

function loadconf ()
-- Default configuration filename:
local conffile = "lwarpmk.conf"
-- Optional configuration filename:
if arg[2] ~= nil then conffile = arg[2]..".lwarpmkconf" end
-- Default language:
language = "english"
-- Default xdyfile:
xdyfile = "lwarp.xdy"
-- Verify the file exists:
if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
print("lwarpmk: " .. conffile .." does not exist.")
print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
printhelp () ;
os.exit(1) -- exit the entire lwarpmk script
else -- file exists
-- Read the file:
print ("lwarpmk: Reading " .. conffile ..".")
local cfile = io.open(conffile)
-- Scan each line:
local linenum = 0
for line in cfile:lines() do -- scan lines
linenum = linenum + 1
i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_%.]*)\"") ;
-- Error if incorrect enclosing characters:
if ( i == nil ) then
print ( linenum .. " : " .. line ) ;
```

```
print ( "lwarpmk: Incorrect entry in " .. conffile ..".\n" ) ;
printconf () ;
os.exit(1) ;
end
if ( cvarname == "opsystem" ) then
    -- Verify choice of opsystem:
    if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
        opsystem = cvalue
    else
        cvalueerror ( line, linenum , cvalue )
    end
elseif ( cvarname == "latexname" ) then
    -- Verify choice of LaTeX compiler:
    if (
        (cvalue == "pdflatex") or
        (cvalue == "xelatex") or
        (cvalue == "lualatex")
    ) then
        latexname = cvalue
    else
        cvalueerror ( line, linenum , cvalue )
    end
elseif ( cvarname == "sourcename" ) then sourcename = cvalue
elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
elseif ( cvarname == "latexmk" ) then latexmk = cvalue
elseif ( cvarname == "language" ) then language = cvalue
elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
else
print ( linenum .. " : " .. line ) ;
print ("lwarpmk: Incorrect variable name \"" .. cvarname .. "\" in " .. conffile ..".\n" ) ;
printconf () ;
os.exit(1) ;
end
end -- do scan lines
io.close(cfile)
end -- file exists
-- Select some operating-system commands:
if opsystem=="Unix" then  -- For Unix / Linux / Mac OS:
rmname = "rm"
mvname = "mv"
touchnamepre = "touch"
touchnamepost = ""
dirslash = "/"
opquote= "\'"
elseif opsystem=="Windows" then -- For Windows
rmname = "DEL"
mvname = "MOVE"
touchnamepre = "COPY /b"
touchnamepost = "+,,"
```

```
dirslash = "\\"
opquote= "\""
else print ( "lwarpmk: Select Unix or Windows for opsystem" )
end --- for Windows

-- set xindycmd according to pdflatex vs xelatex/lualatex:
if ( latexname == "pdflatex" ) then
xindycmd = "texindy  -C utf8"
glossarycmd = "xindy -C utf8"
else
xindycmd = "xindy  -M texindy  -C utf8"
glossarycmd = "xindy -C utf8"
end

end -- loadconf


function refreshdate ()
os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
end


-- Scan the LaTeX log file for the phrase "Rerun to get",
-- indicating that the file should be compiled again.
-- Return true if found.

function reruntoget (filesource)
local fsource = io.open(filesource)
for line in fsource:lines() do
if ( string.find(line,"Rerun to get") ~= nil ) then
    io.close(fsource)
    return true
end
end
io.close(fsource)
return false
end


-- Compile one time, return true if should compile again.
-- fsuffix is "" for print, "_html" for HTML output.

function onetime (fsuffix)
print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
err = os.execute(
--    "echo " ..
    latexname .. " " .. sourcename..fsuffix )
if ( err ~= 0 ) then print ( "lwarpmk: Compile error.") ; os.exit(1) ; end
return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
end
```

```
3019
3020
3021 -- Compile up to five times.
3022 -- fsuffix is "" for print, "_html" for HTML output
3023
3024 function manytimes (fsuffix)
3025 if onetime(fsuffix) == true then
3026 if onetime(fsuffix) == true then
3027 if onetime(fsuffix) == true then
3028 if onetime(fsuffix) == true then
3029 if onetime(fsuffix) == true then
3030 end end end end end
3031 end
3032
3033 -- Exit if the given file does not exist.
3034
3035 function verifyfileexists (filename)
3036 if (lfs.attributes ( filename , "modification" ) == nil ) then
3037 print ( "lwarpmk: " .. filename .. " not found." ) ;
3038 os.exit (1) ;
3039 end
3040 end
3041
3042
3043 -- Convert <project>_html.pdf into HTML files:
3044
3045 function pdftohtml ()
3046     -- Convert to text:
3047     print ("lwarpmk: Converting " .. sourcename
3048         .."_html.pdf to " .. sourcename .. "_html.html")
3049     os.execute("pdftotext  -enc UTF-8  -nopgbrk  -layout "
3050         .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3051     -- Split the result into individual HTML files:
3052     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3053 end
3054
3055
3056 -- Remove auxiliary files:
3057
3058 function removeaux ()
3059     os.execute ( rmname .. " " ..
3060         sourcename ..".aux " .. sourcename .. "_html.aux " ..
3061         sourcename ..".toc " .. sourcename .. "_html.toc " ..
3062         sourcename ..".lof " .. sourcename .. "_html.lof " ..
3063         sourcename ..".lot " .. sourcename .. "_html.lot " ..
3064         sourcename ..".idx " .. sourcename .. "_html.idx " ..
3065         sourcename ..".ind " .. sourcename .. "_html.ind " ..
3066         sourcename ..".log " .. sourcename .. "_html.log " ..
3067         sourcename ..".gl* " .. sourcename .. "_html.gl* " ..
3068         "*_html_inc.*"
```

```
        )
end



-- Create lateximages based on lateximages.txt:
function createlateximages ()
print ("lwarpmk: Creating lateximages.")
local limagesfile = io.open("lateximages.txt")
-- Create the lateximages directory, ignore error if already exists
err = os.execute("mkdir lateximages")
-- Scan lateximages.txt
for line in limagesfile:lines() do
-- lwimgpage is the page number in the PDF which has the image
-- lwimgnum is the sequential lateximage number to assign for the image
i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
-- For each entry:
if ( (i~=nil) ) then
-- Separate out the image into its own single-page pdf:
err = os.execute(
"pdfseparate -f " .. lwimgpage .. " -l " ..
 lwimgpage .. " " .. sourcename .."_html.pdf lateximagetemp-%d.pdf")
-- Crop the image:
err = os.execute(
"pdfcrop --hires lateximagetemp-" .. lwimgpage ..".pdf lateximage-" .. lwimgnum ..".pdf")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Convert the image to svg:
err = os.execute(
"pdftocairo -svg lateximage-" .. lwimgnum ..".pdf lateximage-" .. lwimgnum ..".svg")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Move the result into lateximages/:
err = os.execute(
mvname .. " lateximage-" .. lwimgnum ..".svg lateximages" .. dirslash )
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
-- Remove the temporary files:
err = os.execute(
rmname .. " lateximage-" .. lwimgnum ..".pdf lateximagetemp-" .. lwimgpage ..".pdf")
if ( err ~= 0 ) then print ( "lwarpmk: File error.") ; os.exit(1) ; end
end
end -- do
io.close(limagesfile)
end -- function



-- Use latexmk to compile source and index:
-- fsuffix is "" for print, or "_html" for HTML
function compilelatexmk ( fsuffix )
    -- The recorder option is required to detect changes in <project>.tex
    -- while we are loading <project>_html.tex.
    err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
```

```
3119         .. "-e "
3120         .. opquote
3121         .. "$makeindex = q/" -- $
3122         .. xindycmd
3123         .. "  -M " .. xdyfile
3124         .. "  -L " .. language .. " /"
3125         .. opquote
3126         .. " -pdflatex=\"" .. latexname .." %O %S\" "
3127         .. sourcename..fsuffix ..".tex" ) ;
3128     if ( err ~= 0 ) then print ( "lwarpmk: Compile error.") ; os.exit(1) ; end
3129 end
3130
3131
3132
3133 -- lwarpmk --version :
3134
3135 if (arg[1] == "--version") then
3136 print ( "lwarpmk: " .. printversion )
3137
3138 else -- not --version
3139
3140 -- print intro:
3141
3142 print ("lwarpmk: " .. printversion .. "  Automated make for the LaTeX lwarp package.")
3143
3144 -- lwarpmk print:
3145
3146 if arg[1] == "print" then
3147 loadconf ()
3148 if ( latexmk == "true" ) then
3149     compilelatexmk ("")
3150     print ("lwarpmk: Done.")
3151 else -- not latexmk
3152     verifyfileexists (sourcename .. ".tex") ;
3153     -- See if up to date:
3154     if (
3155         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3156         (
3157             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3158             lfs.attributes ( sourcename .. ".pdf" , "modification" )
3159         )
3160     ) then
3161         -- Recompile if not yet up to date:
3162         manytimes("")
3163         print ("lwarpmk: Done.") ;
3164     else
3165         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3166     end
3167 end -- not latexmk
3168
```

```
elseif arg[1] == "print1" then
    loadconf ()
    verifyfileexists (sourcename .. ".tex") ;
    onetime("")
    print ("lwarpmk: Done.") ;

-- lwarp printindex:
-- Compile the index then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "printindex" then
loadconf ()
print ("lwarpmk: Processing the index.")
os.execute(
    xindycmd
    .. "  -M " .. xdyfile
    .. "  -L " .. language
    .. " " .. sourcename .. ".idx")
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarp printglossary:
-- Compile the glossary then touch the source
-- to trigger a recompile of the document:

elseif arg[1] == "printglossary" then
loadconf ()
print ("lwarpmk: Processing the glossary.")

os.execute(glossarycmd .. "  -L " .. language .. "  -I xindy -M " .. sourcename ..
    " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
    .. sourcename .. ".glo")
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk html:

elseif arg[1] == "html" then
loadconf ()
if ( latexmk == "true" ) then
    compilelatexmk ("_html")
    pdftohtml ()
    print ("lwarpmk: Done.")
else -- not latexmk
    verifyfileexists ( sourcename .. ".tex" ) ;
    -- See if exists and is up to date:
```

```
3219     if (
3220         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3221         (
3222             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3223             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3224         )
3225     ) then
3226         -- Recompile if not yet up to date:
3227         manytimes("_html")
3228         pdftohtml ()
3229         print ("lwarpmk: Done.")
3230     else
3231         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3232     end
3233 end -- not latexmk
3234
3235 elseif arg[1] == "html1" then
3236     loadconf ()
3237     verifyfileexists ( sourcename .. ".tex" ) ;
3238     onetime("_html")
3239     pdftohtml ()
3240     print ("lwarpmk: Done.")
3241
3242 elseif arg[1] == "pdftohtml" then
3243     loadconf ()
3244     pdftohtml ()
3245
3246 -- lwarpmk htmlindex:
3247 -- Compile the index then touch the source
3248 -- to trigger a recompile of the document:
3249
3250 elseif arg[1] == "htmlindex" then
3251 loadconf ()
3252 print ("lwarpmk: Processing the index.")
3253 os.execute(
3254     xindycmd
3255     .. "  -M " .. xdyfile
3256     .. "  -L " .. language
3257     .. " " .. sourcename .. "_html.idx"
3258 )
3259 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
3260 refreshdate ()
3261 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
3262 print ("lwarpmk: Done.")
3263
3264 -- lwarpmk htmlglossary:
3265 -- Compile the glossary then touch the source
3266 -- to trigger a recompile of the document:
3267
3268 elseif arg[1] == "htmlglossary" then
```

```
loadconf ()
print ("lwarpmk: Processing the glossary.")

os.execute(glossarycmd .. "  -L " .. language .. "  -I xindy -M " ..sourcename ..
    "_html -t " .. sourcename .. "_html.glg -o " ..sourcename ..
    "_html.gls " ..sourcename .. "_html.glo")

print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk limages:
-- Scan the lateximages.txt file to create lateximages,
-- then touch the source to trigger a recompile.

elseif arg[1] == "limages" then
loadconf ()
print ("lwarpmk: Processing images.")
createlateximages ()
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk again:
-- Touch the source to trigger a recompile.

elseif arg[1] == "again" then
loadconf ()
print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
refreshdate ()
print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
print ("lwarpmk: Done.")

-- lwarpmk clean:
-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*

elseif arg[1] == "clean" then
loadconf ()
removeaux ()
print ("lwarpmk: Done.")

-- lwarpmk cleanall
-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
--    and also project.pdf, *.html

elseif arg[1] == "cleanall" then
loadconf ()
removeaux ()
```

```
os.execute ( rmname .. " " ..
    sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
    "*.html"
    )
print ("lwarpmk: Done.")

-- lwarpmk with no argument :

elseif (arg[1] == nil) then
printhelp ()

-- lwarpmk -h or lwarpmk --help :

elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
printusage ()

else
print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
printhelp ()
end

end -- not --version
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting

\end{LWR@createlwarpmk}
```

34 Stacks

for HTML output:

```
\begin{warpHTML}
```

Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

34.1 Assigning depths

initial depths for empty stack entries:

```
\newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than `LWR@depthfinished`:

```
\newcommand*{\LWR@depthfinished}{-4}
\newcommand*{\LWR@depthpart}{-1}
\newcommand*{\LWR@depthchapter}{0}
\newcommand*{\LWR@depthsection}{1}
\newcommand*{\LWR@depthsubsection}{2}
\newcommand*{\LWR@depthsubsubsection}{3}
\newcommand*{\LWR@depthparagraph}{4}
\newcommand*{\LWR@depthsubparagraph}{5}
```

used by `\itemize`, `\enumerate`, `\description`:

```
\newcommand*{\LWR@depthlist}{6}
```

used by `\item`:

```
\newcommand*{\LWR@depthlistitem}{7}
```

34.2 Closing actions

A stack to record the action to take to close each nesting level: Add more levels of stack if necessary for a very deeply nested document, adding to `\pushclose` and `\popclose` as well.

```
\newcommand*{\LWR@closeone}{}% top of the stack
\newcommand*{\LWR@closetwo}{}
\newcommand*{\LWR@closethree}{}
\newcommand*{\LWR@closefour}{}
\newcommand*{\LWR@closefive}{}
\newcommand*{\LWR@closesix}{}
\newcommand*{\LWR@closeseven}{}
\newcommand*{\LWR@closeeight}{}
\newcommand*{\LWR@closenine}{}
\newcommand*{\LWR@closeten}{}
\newcommand*{\LWR@closeeleven}{}
\newcommand*{\LWR@closetwelve}{}
```

34.3 Closing depths

A stack to record the depth of each level:

⚠ Note that nested LaTeX structures may push depths which are non-sequential.

Ex:

```
\begin{itemize}
   \item{A}
   \begin{description}
      \item{B}
   \end{description}
\end{itemize}
```

```
\newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
\newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
\newcommand*{\LWR@closedepththree}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
\newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
\newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
\newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
\newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
\newcommand*{\LWR@closedepththen}{\LWR@depthnone}
\newcommand*{\LWR@closedepththeleven}{\LWR@depthnone}
\newcommand*{\LWR@closedepththtwelve}{\LWR@depthnone}
```

34.4 Pushing and popping the stack

`\pushclose` {⟨*action*⟩} {⟨*depth*⟩}

Pushes one return action and its LaTeX depth onto the stacks.

```
\NewDocumentCommand{\pushclose}{m m}
{
\global\let\LWR@closetwelve\LWR@closeeleven
\global\let\LWR@closeeleven\LWR@closeten
\global\let\LWR@closeten\LWR@closenine
\global\let\LWR@closenine\LWR@closeeight
\global\let\LWR@closeeight\LWR@closeseven
\global\let\LWR@closeseven\LWR@closesix
\global\let\LWR@closesix\LWR@closefive
```

```
\global\let\LWR@closefive\LWR@closefour
\global\let\LWR@closefour\LWR@closethree
\global\let\LWR@closethree\LWR@closetwo
\global\let\LWR@closetwo\LWR@closeone
\global\let\LWR@closeone#1
\global\let\LWR@closedepthtwelve\LWR@closedepttheleven
\global\let\LWR@closedepttheleven\LWR@closedepththten
\global\let\LWR@closedepththten\LWR@closedepthnine
\global\let\LWR@closedepthnine\LWR@closedeptheight
\global\let\LWR@closedeptheight\LWR@closedepthseven
\global\let\LWR@closedepthseven\LWR@closedepthsix
\global\let\LWR@closedepthsix\LWR@closedepthfive
\global\let\LWR@closedepthfive\LWR@closedepthfour
\global\let\LWR@closedepthfour\LWR@closedepththree
\global\let\LWR@closedepththree\LWR@closedepthtwo
\global\let\LWR@closedepthtwo\LWR@closedepthone
\global\let\LWR@closedepthone#2
}
```

`\popclose` Pops one action and its depth off the stacks.

```
\newcommand*{\popclose}
{
\global\let\LWR@closeone\LWR@closetwo
\global\let\LWR@closetwo\LWR@closethree
\global\let\LWR@closethree\LWR@closefour
\global\let\LWR@closefour\LWR@closefive
\global\let\LWR@closefive\LWR@closesix
\global\let\LWR@closesix\LWR@closeseven
\global\let\LWR@closeseven\LWR@closeeight
\global\let\LWR@closeeight\LWR@closenine
\global\let\LWR@closenine\LWR@closeten
\global\let\LWR@closeten\LWR@closeeleven
\global\let\LWR@closeeleven\LWR@closetwelve
\global\let\LWR@closedepthone\LWR@closedepthtwo
\global\let\LWR@closedepthtwo\LWR@closedepththree
\global\let\LWR@closedepththree\LWR@closedepthfour
\global\let\LWR@closedepthfour\LWR@closedepthfive
\global\let\LWR@closedepthfive\LWR@closedepthsix
\global\let\LWR@closedepthsix\LWR@closedepthseven
\global\let\LWR@closedepthseven\LWR@closedeptheight
\global\let\LWR@closedeptheight\LWR@closedepthnine
\global\let\LWR@closedepthnine\LWR@closedepththten
\global\let\LWR@closedepththten\LWR@closedepttheleven
\global\let\LWR@closedepttheleven\LWR@closedepthtwelve
}

\end{warpHTML}
```

35 Data arrays

These macros are similar to the arrayjobx package, except that `\LWR@setexparray`'s argument is expanded only once when assigned.

`name` has no backslash, `index` can be a number or a text name, and an empty `value` must be `\relax` instead of empty.

To assign an empty value:

```
\LWR@setexparray{name}{index}{}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@setexparray` {⟨*name*⟩} {⟨*index*⟩} {⟨*contents*⟩}

```
\NewDocumentCommand{\LWR@setexparray}{m m m}{%
\ifstrempty{#3}%
{\csdef{#1#2}{}}%
{\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
}
```

`\LWR@getexparray` {⟨*name*⟩} {⟨*index*⟩}

```
\newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
\end{warpHTML}
```

36 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

for HTML output:

```
\begin{warpHTML}
```

`\LWR@sanitized` The sanitized version of what was given to `\LWR@sanitize`. Characters are set to their detokenized versions. Required for underscores in labels and filenames.

```
\newcommand*{\LWR@sanitized}{}
```

`\LWR@sanitize` {⟨*text*⟩}

Sanitizes the text and returns the result in `\LWR@sanitized`.

```
\newcommand*{\LWR@sanitize}[1]{%
\LWR@traceinfo{LWR@sanitize: !#1!}%
\edef\LWR@sanitized{#1}%
\LWR@traceinfo{LWR@sanitize expanded: !\LWR@sanitized!}%
\edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
\LWR@traceinfo{LWR@sanitize result: !\LWR@sanitized!}%
}

\end{warpHTML}
```

37 HTML entities

for HTML output:

```
\begin{warpHTML}
```

HTML entites and HTML Unicode entities:

```
\let\LWR@origampersand\&
```

`\HTMLentity` {⟨*entitytag*⟩}

```
\newcommand*{\HTMLentity}[1]{%
% \LWR@traceinfo{HTMLentity \detokenize{#1}}%
\begingroup%
\LWR@FBcancel%
\LWR@origampersand#1;%
\endgroup
% \LWR@traceinfo{HTMLentity done}%
}
```

`\HTMLunicode` {⟨*hex_unicode*⟩}

```
\newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

`\&`

```
\renewrobustcmd*{\&}{\HTMLentity{amp}}
```

`\textless`
`\textgreater`

```
\let\LWR@origtextless\textless
\renewcommand*{\textless}{\HTMLentity{lt}}

\let\LWR@origtextgreater\textgreater
\renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
\end{warpHTML}
```

38 HTML filename generation

The filename of the homepage is set to `\HomeHTMLFilename.html`. The filenames of additional sections start with `\HTMLFilename`, to which is appended a section number or a simplified section name, depending on `FileSectionNames`.

for HTML & PRINT:

```
\begin{warpall}
```

`\BaseJobname` The `\jobname` of the printed version, even if currently compiling the HTML version. I.e. this is the `\jobname` without `_html` appended. This is used to set `\HomeHTMLFilename` if the user did not provide one.

```
\providecommand*{\BaseJobname}{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 7.3.1.

```
\providecommand*{\HTMLFilename}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 7.3.1.

```
\providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` {⟨*number*⟩}

Sets the file number for the next file to be generated. 0 is the home page. Use just before the next sectioning command, and set it to one less than the desired number of the next section. May be used to generate numbered groups of nodes such as 100+ for one chapter, 200+ for another chapter, etc.

```
\newcommand*{\SetHTMLFileNumber}[1]{%
\setcounter{LWR@htmlfilenumber}{#1}%
}
```

Bool `FileSectionNames` Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
\newbool{FileSectionNames}
\booltrue{FileSectionNames}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Ctr `LWR@htmlfilenumber`

Records the number of each HTML file as it is being created. Number 0 is the home page.

```
\newcounter{LWR@htmlfilenumber}
\setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` {⟨*htmlfilenumber or name*⟩}

Prints the filename for a given section: `\HTMLFilename{}filenumber/name.html`

```
\newcommand*{\LWR@htmlsectionfilename}[1]{%
\LWR@traceinfo{LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

Section 0 or empty is given the home filename. The filename must be detokenized for underscores.

```
% \LWR@traceinfo{about to assign temp}%
\edef\LWR@tempone{#1}%
\LWR@traceinfo{about to compare with ??}%
\ifthenelse{\equal{\LWR@tempone}{??}}%
{\LWR@traceinfo{found ??}}%
{\LWR@traceinfo{not found ??}}%
\LWR@traceinfo{about to compare with zero or empty}%
\ifthenelse{%
    \equal{\LWR@tempone}{0}%
    \OR \equal{\LWR@tempone}{}%
    \OR \equal{\LWR@tempone}{??}%
}%
{%
    \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
    \HomeHTMLFilename.html%
}%
```

For a LaTeX section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```
{%
    \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
    \ifthenelse{%
        \equal{\HTMLFilename}{} \AND
        \equal{\LWR@tempone}{Index} \OR
        \equal{\LWR@tempone}{index}%
    }%
    {%
```

```
        \LWR@traceinfo{Prefixing the index name with an underscore.}%
        \_#1.html%
    }%
```

Otherwise, create a filename with the chosen prefix:

```
    {\HTMLFilename#1.html}%
}%
\LWR@traceinfo{LWR@htmlsectionfilename Z}%
}
```

`\LWR@htmlrefsectionfilename` {⟨*label*⟩}

Prints the filename for the given label

```
\newcommand*{\LWR@htmlrefsectionfilename}[1]{%
\LWR@traceinfo{LWR@htmlrefsectionfilename: !\detokenize{#1}!}%
\LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
\LWR@traceinfo{LWR@htmlrefsectionfilename: done}%
}
```

```
\end{warpHTML}
```

39 Homepage link

for HTML output:

```
\begin{warpHTML}
```

`\LinkHome` `\LinkHome` may be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

```
\newcommand*{\LinkHome}{%
\LWR@subhyperrefclass{%
\HomeHTMLFilename.html}%
{Home}{linkhome}%
}
```

`\LWR@topnavigation` Creates a link to the homepage at the top of the page for use when the window is too narrow for the sideTOC.

```
\newcommand*{\LWR@topnavigation}{
\LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}
}
```

\LWR@botnavigation Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sideTOC.

```
\newcommand*{\LWR@botnavigation}{
\LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}
}
```

```
\end{warpHTML}
```

40 \LWRPrintStack diagnostic tool

⚠ Diagnostics tool: Prints the LaTeX nesting depth values for the stack levels. \LWR@startpars is used before printing the stack, so that \LWRPrintStack may be called from anywhere in the normal text flow.

for HTML output:

```
\begin{warpHTML}
```

\LWRPrintStack Prints the closedepth stack.

```
\newcommand*{\LWR@subprintstack}{
\LWR@closedepthone\ \LWR@closedepththwo\ \LWR@closedepththree\
\LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\
\LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\
\LWR@closedepthten\ \LWR@closedeptheleven\ \LWR@closedepthtwelve\
}

\newcommand*{\LWRPrintStack}{
\LWR@startpars
\LWR@subprintstack
}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

```
\newcommand*{\LWRPrintStack}{}
```

```
\end{warpprint}
```

41 Closing stack levels

for HTML output:

```
\begin{warpHTML}
```

Close one nested level:

```
\newcommand*{\LWR@closeoneprevious}{%

\LWR@closeone

\popclose
}
```

`\LWR@closeprevious` `{`⟨*depth*⟩`}` Close everything up to the given depth:

```
\newcommand*{\LWR@closeprevious}[1]{
\LWR@traceinfo{LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%
```

Close any pending paragraph:

```
\LWR@stoppars%
```

Close anything nested deeper than the desired depth. First close anything deeper, then at most one of the same level.

```
\whileboolexpr{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}%
{%
    \LWR@traceinfo{LWR@closeprevious: closing out depth \LWR@closedepthone}%
    \LWR@closeoneprevious%
}%
\ifboolexpr{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}%
{%
    \LWR@traceinfo{LWR@closeprevious: closing out depth \LWR@closedepthone}%
    \LWR@closeoneprevious%
}{}%
\LWR@traceinfo{LWR@closeprevious: done, depths are \LWR@subprintstack}%
}

\end{warpHTML}
```

42 PDF pages and styles

for HTML output:

```
\begin{warpHTML}
```

`\LWR@forcenewpage` New PDF page a before major environment.

This is used just before major environments, such as `verse`. Reduces the chance of an environment overflowing the HTML PDF output page.

```
\newcommand{\LWR@forcenewpage}{%
\ifinner\else%
\LWR@stoppars\LWR@orignewpage\LWR@startpars%
\fi%
}
```

\pagestyle, etc. are nullified for HTML output.

`\pagestyle` {⟨*style*⟩}

```
\renewcommand*{\pagestyle}[1]{}
```

`\thispagestyle` {⟨*style*⟩}

```
\renewcommand*{\thispagestyle}[1]{}
```

`\markboth` {⟨*left*⟩} {⟨*right*⟩}

```
\renewcommand*{\markboth}[2]{}
```

`\markright` {⟨*right*⟩}

```
\renewcommand*{\markright}[1]{}
```

`\raggedbottom`

```
\renewcommand*{\raggedbottom}{}
```

`\flushbottom`

```
\renewcommand*{\flushbottom}{}
```

`\sloppy`

```
\renewcommand*{\sloppy}{}
```

`\fussy`

```
\renewcommand*{\fussy}{}
```

`\pagenumbering` * {⟨*commands*⟩}

```
\RenewDocumentCommand{\pagenumbering}{s m}{}
```

```
\end{warpHTML}
```

43 HTML tags, spans, divs, elements

for HTML output:

```
\begin{warpHTML}
```

43.1 Mapping LaTeX Sections to HTML Sections

```
\newcommand*{\LWR@tagtitle}{h1}
\newcommand*{\LWR@tagtitleend}{/h1}
\newcommand*{\LWR@tagpart}{h2}
\newcommand*{\LWR@tagpartend}{/h2}
\newcommand*{\LWR@tagchapter}{h3}
\newcommand*{\LWR@tagchapterend}{/h3}
\newcommand*{\LWR@tagsection}{h4}
\newcommand*{\LWR@tagsectionend}{/h4}
\newcommand*{\LWR@tagsubsection}{h5}
\newcommand*{\LWR@tagsubsectionend}{/h5}
\newcommand*{\LWR@tagsubsubsection}{h6}
\newcommand*{\LWR@tagsubsubsectionend}{/h6}
\newcommand*{\LWR@tagparagraph}{span class="paragraph"}
\newcommand*{\LWR@tagparagraphend}{/span}
\newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\newcommand*{\LWR@tagsubparagraphend}{/span}

\newcommand*{\LWR@tagregularparagraph}{p}
```

43.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for `pdflatex` and `xelatex`.

(*Emulates or patches code by* Daniel Flipo.)

```
\providecommand*{\LWR@FBcancel}{}

\AtBeginDocument{%
\@ifundefined{frenchbsetup}%
{}%
{%
    \frenchbsetup{FrenchFootnotes=false}%
```

```
%
    \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
    \renewrobustcmd*{\FBcolonspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;%
        \endgroup%
    }%
    \renewrobustcmd*{\FBthinspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\#x202f;% \,
        \endgroup%
    }%
    \renewrobustcmd*{\FBguillspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~,  for \og xyz \fg{}
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBmedkern}{}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\#x202f;% \,
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBthickkern}{}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~
        \endgroup%
    }%
    \renewrobustcmd*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
    \ifFBunicode%
    \else%
        \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
        \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
    \fi%
}%
}
```

43.3 HTML tags

`\LWR@htmltagc` {⟨*tag*⟩} Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOF, LOT.

```
\newcommand*{\LWR@htmltagc}[1]{%
{%
\LWR@traceinfo{LWR@htmltagc !\detokenize{#1}!}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\protect\LWR@origtextless%
#1%
\protect\LWR@origtextgreater%
\endgroup%
}%
}
```

Env `LWR@nestspan` Disable `minipage`, `\parbox`, and HTML `<div>`s inside a `<span>`.

⚠ `\begin{LWR@nestspan}` must follow the opening `<span>` tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

⚠ `\end{LWR@nestspan}` must follow the `</span>` or a `<p>` may appear inside the span.

```
\newcommand*{\LWR@nestspanitem}{%
\if@newlist\else{\LWR@htmltagc{br /}}\fi%
\LWR@origitem%
}

\newenvironment*{LWR@nestspan}
{%
\LWR@traceinfo{LWR@nestspan starting}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{%
    \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
}%
{% not in a lateximage
    \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
    \addtocounter{LWR@spandepth}{1}%
    \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
    \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
    \renewcommand{\BlockClassSingle}[2]{##2}%
    \renewcommand{\LWR@forcenewpage}{}%
    \renewcommand{\LWR@liststart}{%
        \let\item\LWR@nestspanitem%
    }%
    \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
}% not in a lateximage
\LWR@traceinfo{LWR@nestspan starting: done}%
}% starting env
{% ending env
\LWR@traceinfo{LWR@nestspan ending}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
```

```
{}%
{\addtocounter{LWR@spandepth}{-1}}%
\LWR@traceinfo{LWR@nestspan ending: done}%
}

\AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}
```

`\LWR@htmlspan` {⟨*tag*⟩} {⟨*text*⟩}

`\LWR@spandepth` is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to correctly handle the paragraph tags before and after the span.

```
\NewDocumentCommand{\LWR@htmlspan}{m +m}{%
\LWR@ensuredoingapar%
\LWR@htmltagc{#1}%
\begin{LWR@nestspan}%
#2%
\LWR@htmltagc{/#1}%
\end{LWR@nestspan}%
}
```

`\LWR@htmlspanclass` [⟨*style*⟩] {⟨*class*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
\LWR@traceinfo{LWR@htmlspanclass #2}%
\LWR@ensuredoingapar%
\LWR@subhtmlelementclass{span}[#1]{#2}%
\begin{LWR@nestspan}%
#3%
\LWR@htmltagc{/span}%
\LWR@traceinfo{LWR@htmlspanclass done}%
\end{LWR@nestspan}%
}
```

`\LWR@htmltag` {⟨*tag*⟩}

Print an HTML tag: <tag>

```
\newcommand*{\LWR@htmltag}[1]{%
% \LWR@traceinfo{LWR@htmltagb !\detokenize{#1}!}%
\LWR@htmltagc{#1}%
% \LWR@traceinfo{LWR@htmltagb: done}%
}
```

43.4 Block tags and comments

In the following, \origttfamily breaks ligatures, which may not be used for HTML codes:

\LWR@htmlopencomment
\LWR@htmlclosecomment

```
\newcommand*{\LWR@htmlopencomment}{%
{%
% \LWR@traceinfo{LWR@htmlopencomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\LWR@origtextless{}!{-}{-}%
\endgroup%
}%
}

\newcommand*{\LWR@htmlclosecomment}{%
{%
% \LWR@traceinfo{LWR@htmlclosecomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
{-}{-}\LWR@origtextgreater{}%
\endgroup%
}%
}
```

\LWR@htmlcomment {⟨*comment*⟩}

```
\newcommand{\LWR@htmlcomment}[1]{%
\LWR@htmlopencomment{}%
{%
\LWR@origttfamily% break ligatures
#1%
}%
\LWR@htmlclosecomment{}}
```

\LWR@htmlblockcomment {⟨*comment*⟩}

```
\newcommand{\LWR@htmlblockcomment}[1]
{\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars}
```

\LWR@htmlblocktag {⟨*tag*⟩} print a stand-alone HTML tag

```
\newcommand*{\LWR@htmlblocktag}[1]{%
\LWR@stoppars%
\LWR@htmltag{#1}%
\LWR@startpars%
}
```

43.5 Div class and element class

\LWR@subhtmlelementclass {⟨*element*⟩} [⟨*style*⟩] {⟨*class*⟩}

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
\NewDocumentCommand{\LWR@subhtmlelementclass}{m O{} m}{%
\LWR@traceinfo{LWR@subhtmlelementclass #1 #3}%
\ifblank{#2}%
{\LWR@htmltag{#1 class="#3"}}% empty option
{\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
\LWR@traceinfo{LWR@subhtmlelementclass done}%
}
```

\LWR@htmlelementclass {⟨*element*⟩} {⟨*class*⟩} [⟨*style*⟩]

```
\NewDocumentCommand{\LWR@htmlelementclass}{m o m}{%
\LWR@stoppars%
\LWR@subhtmlelementclass{#1}[#2]{#3}%
\LWR@startpars%
}
```

\LWR@htmlelementclassend {⟨*element*⟩} {⟨*class*⟩}

```
\newcommand*{\LWR@htmlelementclassend}[2]{%
\LWR@stoppars%
\LWR@htmltag{/#1}%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{End of #1 ``#2''}%
}{}%
\LWR@startpars%
}
```

\LWR@htmldivclass [⟨*style*⟩] {⟨*class*⟩}

```
\NewDocumentCommand{\LWR@htmldivclass}{o m}{%
```

```
\LWR@htmlelementclass{div}[#1]{#2}%
}
```

`\LWR@htmldivclassend` {⟨*class*⟩}

```
\newcommand*{\LWR@htmldivclassend}[1]{%
\LWR@htmlelementclassend{div}{#1}%
}
```

43.6 Single-line elements

A single-line element, without a paragraph tag for the line of text:

`\LWR@htmlelementclassline` {⟨*element*⟩} [⟨*style*⟩] {⟨*class*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
\LWR@stoppars
\LWR@subhtmlelementclass{#1}[#2]{#3}%
#4%
\LWR@htmltag{/#1}
\LWR@startpars
}
```

43.7 HTML5 semantic elements

`\LWR@htmlelement` {⟨*element*⟩}

```
\newcommand*{\LWR@htmlelement}[1]{%
\LWR@htmlblocktag{#1}
}
```

`\LWR@htmlelementend` {⟨*element*⟩}

```
\newcommand*{\LWR@htmlelementend}[1]{%
\LWR@stoppars
\LWR@htmltag{/#1}
\LWR@startpars
}

\end{warpHTML}
```

43.8 High-level block and inline classes

These are high-level commands which allow the creation of arbitrary block or inline sections which may be formatted with css.

Nullified versions are provided for print mode.

For other direct-formatting commands, see section 79.

Env `BlockClass` `[`⟨*style*⟩`]` `{`⟨*class*⟩`}` High-level interface for `<div>` classes.

Ex: `\begin{BlockClass}{class} text \end{BlockClass}`

for HTML output:
```
\begin{warpHTML}
\NewDocumentEnvironment{BlockClass}{o m}%
{\LWR@htmldivclass[#1]{#2}}
{\LWR@htmldivclassend{#2}}
\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}
\NewDocumentEnvironment{BlockClass}{o m}{}{}%
\end{warpprint}
```

`\BlockClassSingle` `{`⟨*class*⟩`}` `{`⟨*text*⟩`}` A single-line `<div>`, without a paragraph tag for the line of text.

for HTML output:
```
\begin{warpHTML}
\newcommand{\BlockClassSingle}[2]{%
\LWR@htmlelementclassline{div}{#1}{#2}%
}
\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}
\newcommand{\BlockClassSingle}[2]{#2}
\end{warpprint}
```

`\InlineClass` `[`⟨*style*⟩`]` `{`⟨*class*⟩`}` `{`⟨*text*⟩`}` High-level interface for inline span classes.

for HTML output:
```
\begin{warpHTML}
\NewDocumentCommand{\InlineClass}{o m +m}{%
\LWR@htmlspanclass[#1]{#2}{#3}%
}
\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}
\NewDocumentCommand{\InlineClass}{o m +m}{#3}%
\end{warpprint}
```

Env `LWR@BlockClassWP` {⟨*WPstyle*⟩} {⟨*HTMLstyle*⟩} {⟨*class*⟩} Low-level interface for `<div>` classes with an automatic float ID. These are often used when `\ifbool{FormatWP}`.

for HTML output:

```
\begin{warpHTML}
\NewDocumentEnvironment{LWR@BlockClassWP}{m m m}%
{%
\LWR@stoppars%
\ifbool{FormatWP}%
{%
    \addtocounter{LWR@thisautoidWP}{1}%
    \LWR@htmltag{%
        div class="#3" %
        id="autoidWP-\arabic{LWR@thisautoidWP}"%
        \ifblank{#1}{}{ style="#1"}%
    }%
}% FormatWP
{% not FormatWP
    \LWR@htmltag{%
        div class="#3"%
        \ifblank{#2}{}{ style="#2"}%
    }%
}% not FormatWP
\LWR@startpars%
}
{\LWR@htmldivclassend{#3}}
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{}{}%
\end{warpprint}
```

43.9 Closing HTML tags

for HTML output:

```
\begin{warpHTML}
```

Sections H1, H2, etc. do not need a closing HTML tag, but we add a comment for readability:

```
\newcommand*{\LWR@printclosepart}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
\newcommand*{\LWR@printclosechapter}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
\newcommand*{\LWR@printclosesection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
\newcommand*{\LWR@printclosesubsection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
\newcommand*{\LWR@printclosesubsubsection}
```

```
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
\newcommand*{\LWR@printcloseparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
\newcommand*{\LWR@printclosesubparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
\newcommand*{\LWR@printcloselistitem}
    {\LWR@htmltag{/li}}
\newcommand*{\LWR@printclosedescitem}
    {\LWR@htmltag{/dd}}
\newcommand*{\LWR@printcloseitemize}
    {\LWR@htmltag{/ul}}
\newcommand*{\LWR@printcloseenumerate}
    {\LWR@htmltag{/ol}}
\newcommand*{\LWR@printclosedescription}
    {\LWR@htmltag{/dl}}

\end{warpHTML}
```

44 Paragraph handling

These commands generate the HTML paragraph tags when allowed and required.

Paragraph tags are or are not allowed depending on many conditions. Section 45 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output:

```
\begin{warpHTML}
```

Ctr `LWR@spandepth` Do not create paragraph tags inside of an HTML span.

```
\newcounter{LWR@spandepth}
\setcounter{LWR@spandepth}{0}
```

Bool `LWR@doingstartpars` Tells whether paragraphs may be generated.

```
\newbool{LWR@doingstartpars}
\boolfalse{LWR@doingstartpars}
```

Bool `LWR@doingapar` Tells whether have actually generated and are currently processing paragraph text.

```
\newbool{LWR@doingapar}
\global\boolfalse{LWR@doingapar}
```

`\LWR@ensuredoingapar` If are about to print something visible, and if allowed to start a new paragraph, ensure that are `LWR@doingapar`, so that paragraph tags are placed:

```
\newcommand*{\LWR@ensuredoingapar}{%
\ifbool{LWR@doingstartpars}%
{\global\booltrue{LWR@doingapar}}%
{}%
}
```

`\LWR@openparagraph`

```
\newcommand*{\LWR@openparagraph}
{%
```

See if paragraph handling is enabled:

```
\ifbool{LWR@doingstartpars}%
{% handling pars
```

See if have already started a `lateximage` or a `<span>`. If so, do not generate nested paragraph tags.

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }% nested par tags?
```

If so: Do nothing if already started a `lateximage` page. Cannot nest a `lateximage`. Also do nothing if already inside a `<span>`. Do not nest paragraph tags inside a `<span>`.

```
    {}% no nested par tags
```

Else: No `lateximage` or `<span>` has been started yet, so it's OK to generate paragraph tags.

```
    {% yes nest par tags
        \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Now have started a paragraph.

```
        \global\booltrue{LWR@doingapar}%
```

At the endof each paragraph, generate closing tag and do regular /par stuff. (Attempting to use the everyhook cr hook for \LWR@closeparagraph does not work well.)

```
        \let\par\LWR@closeparagraph%
    }% end of yes nest par tags
}% end of handling pars
{}% not handling pars
}
```

\LWR@closeparagraph

```
\newcommand*{\LWR@closeparagraph}
{%
```

See if paragraph handling is enabled:

```
\ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
{% handling pars
```

See if already started a lateximage or a :

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }%
```

Do nothing if already started a lateximage or a , but add a parbreak if in a span but not a lateximage.

```
    {% no nested par tags
        \ifboolexpr{
            test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
            test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
        }%
        {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
        {}%
    }% no nested par tags
```

If have not already started a lateximage or a :

```
    {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer doing a paragraph:

```
        \global\boolfalse{LWR@doingapar}%
% Disable the special \env{minipage} \& \cs{hspace} interaction
% until a new minipage is found:
%    \begin{macrocode}
        \global\boolfalse{LWR@minipagethispar}%
    }% end of yes nest par tags
}% end of handling pars
```

Add a parbreak if in a span, but not in a table outside a row:

```
{% not handling pars
    \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
    {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
    {}%
}% not handling pars
```

In most cases, finish with a LaTeX \par, but in the case of paragraphs between lines in a tabular fetch the next token instead:

```
\ifboolexpr{%
    not bool {LWR@doingapar} and
    test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
    test {
        \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}
    } and
    bool {LWR@intabularmetadata} and
    not bool {LWR@tableparcell} and
    test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
}%
{%
    \LWR@getmynexttoken%
}{%
    \LWR@origpar%
}%
}
```

```
\end{warpHTML}
```

45 Paragraph start/stop handling

These commands allow/disallow the generation of HTML paragraph tags.

Section 44 has the commands which actually generate the tags.

The everyhook package is used to generate the opening paragraph tags. The closing tags are generated by \par.

for HTML output:

```
\begin{warpHTML}
```

\LWR@startpars Begin handling HTML paragraphs. This allows an HTML paragraph to start, but one has not yet begun.

```
\newcommand*{\LWR@startpars}%
{%
\LWR@traceinfo{LWR@startpars}%
```

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

```
    \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
    {}%
```

If not currently in paragraph mode:

```
    {%
```

At the start of each paragraph, generate an opening tag:

```
        \PushPreHook{par}{\LWR@openparagraph}%
```

At the end of each paragraph, generate closing tag and do regular /par actions:

```
        \let\par\LWR@closeparagraph

    }% an intentionally blank line
```

Are now handling paragraphs, but have not yet actually started one:

```
    \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
    \global\boolfalse{LWR@doingapar}%
}% nestspan
\LWR@traceinfo{LWR@startpars: done}%
}
```

`\LWR@stoppars` Stop handling HTML paragraphs. Any currently open HTML paragraph is closed, and no more will be opened.

```
\newcommand*{\LWR@stoppars}%
{%
```

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

```
    \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
    {%
```

Print a closing tag:

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
        \global\boolfalse{LWR@doingapar}%
```

Disable the special `minipage` & `\hspace` interaction until a new minipage is found:

```
        \global\boolfalse{LWR@minipagethispar}

    }% an intentionally blank line
```

If was not in an HTML paragraph:

```
    {}%
```

See if currently allowing HTML paragraphs:

```
    \ifbool{LWR@doingstartpars}%
```

If so: clear the par hook to no longer catch paragraphs:

```
    {\ClearPreHook{par}}%
```

Else: do nothing

```
    {}%
```

no longer in paragraph mode

```
    \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
    \global\boolfalse{LWR@doingapar}%
}% nestspan
}

\end{warpHTML}
```

46 Page headers and footers

for HTML & PRINT:

```
\begin{warpall}
```

In the following, catcode is manually changed back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
\newcommand{\LWR@firstpagetop}{} % for the home page alone
\newcommand{\LWR@pagetop}{} % for all other pages
\newcommand{\LWR@pagebottom}{}
```

`\HTMLFirstPageTop` {⟨*text and logos*⟩}

```
\newcommand{\HTMLFirstPageTop}[1]{%
    \renewcommand{\LWR@firstpagetop}{#1}%
}
```

\HTMLPageTop {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageTop}[1]{%
    \renewcommand{\LWR@pagetop}{#1}%
}
```

\HTMLPageBottom {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageBottom}[1]{%
    \renewcommand{\LWR@pagebottom}{#1}%
}

\end{warpall}
```

47 CSS

for HTML output:

```
\begin{warpHTML}
```

\LWR@currentcss The css filename to use. This may be changed mid-document using \CSSFilename, allowing different css files to be used for different sections of the document.

```
\newcommand*{\LWR@currentcss}{lwarp.css}
```

\CSSFilename {⟨*new-css-filename.css*⟩} Assigns the css file to be used by the following HTML pages.

```
\newcommand*{\CSSFilename}[1]{%
\renewcommand*{\LWR@currentcss}{#1}%
\@onelevel@sanitize\LWR@currentcss%
}

\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand*{\CSSFilename}[1]{}
\end{warpprint}
```

48 Title, HTML meta author, HTML meta description

for HTML output:

```
\begin{warpHTML}
```

`\title` {⟨*title*⟩} Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
\let\LWR@origtitle\title

\renewcommand*{\title}[1]{%
    \LWR@origtitle{#1}%
    \begingroup%
        \renewcommand{\thanks}[1]{}%
        \protected@xdef\thetitle{#1}%
    \endgroup%
}
```

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

`\HTMLTitle` {⟨*Titlename*⟩} The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
\providecommand{\thetitle}{}

\newcommand{\theHTMLTitle}{\thetitle}

\newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor` {⟨*authorname*⟩} The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the titling package is used.

```
\providecommand{\theauthor}{}

\newcommand{\theHTMLAuthor}{\theauthor}

\newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using `\HTMLDescription`, allowing different HTML descriptions to be used for different sections of the document.

⚠ Do not use double quotes, and do not exceed 150 characters.

`\HTMLDescription` {⟨*New HTML meta description.*⟩} Assigns the HTML file's description meta tag.

```
\newcommand{\LWR@currentHTMLDescription}{}

```

```
\newcommand{\HTMLDescription}[1]{%
\renewcommand{\LWR@currentHTMLDescription}{#1}
}

\end{warpall}
```

49 Footnotes

lwarp uses native LaTeX footnote code, although with its own \box to avoid the LaTeX output routine. The usual functions work as-is.

⚠ pfnote numbers

While emulating pfnote, lwarp is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. lwarp therefore uses continuous footnote numbering even for pfnote.

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

49.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the LaTeX box \LWR@footnotes. Using this instead of the original \footins box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 49.4 for the implementation.

49.2 Minipage footnotes

See section 49.5 for how minipage footnotes are gathered. See section 78.3 for how minipage footnotes are placed into the document.

49.3 Titlepage thanks

See section 56.6 for titlepage footnotes.

49.4 Regular page footnote implementation

for HTML & PRINT:

```
4053 \begin{warpall}
```

Ctr `FootnoteDepth`
Default: 5

Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`. The default of 5 places footnotes before each `\subparagraph` or higher. See table 7 for a table of LaTeX section headings.

```
4054 \newcounter{FootnoteDepth}
4055 \setcounter{FootnoteDepth}{5}
```

```
4056 \end{warpall}
```

for HTML output:

```
4057 \begin{warpHTML}
```

Patch LaTeX footnotes to use a new `\box` instead of an insert for lwarp footnotes. This avoids having the original `\footins` appear at the bottom of a `lateximage`, which is on its own new page.

```
4058 \newbox\LWR@footnotes
```

Much of the following has unneeded print-mode formatting removed.

`\@makefntext` {⟨*text*⟩}

```
4059 \long\def\@makefntext#1{\textsuperscript{\@thefnmark}~#1}
```

`\@makefnmark`

```
4060 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

Footnotes may be in regular text, in which case paragraphs are tagged, or in a table data cell or `lateximage`, in which case paragraph tags must be added manually.

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

`\LWR@footnotetext` {⟨*text*⟩}

```
4061 \long\def\LWR@footnotetext#1{%
4062 \LWR@traceinfo{LWR@footnotetext}%
4063 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
    \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
    \protected@edef\@currentlabel{%
        \csname p@footnote\endcsname\@thefnmark%
    }% @currentlabel
```

Open a group:

```
    \color@begingroup%
```

Use HTML superscripts even inside a `lateximage`:

```
    \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
    \@makefntext{#1}%
```

Closing paragraph tag:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {\par}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
```

Close the group:

```
    \color@endgroup%
}% vbox
}%
```

\@footnotetext {⟨*text*⟩}

```
\let\@footnotetext\LWR@footnotetext
```

49.5 Minipage footnote implementation

Patch LaTeX minipage footnotes to use a new \box instead of an insert for lwarp minipage footnotes. This avoids having the original \@mpfootins appear at the bottom of a lateximage, which is on its own new page.

```
\newbox\LWR@mpfootnotes
```

\@mpfootnotetext {⟨*text*⟩}

```
\long\def\@mpfootnotetext#1{%
\global\setbox\LWR@mpfootnotes\vbox{%
    \unvbox\LWR@mpfootnotes
    \reset@font\footnotesize
    \hsize\columnwidth
    \@parboxrestore
    \protected@edef\@currentlabel
    {\csname p@mpfootnote\endcsname\@thefnmark}%
    \color@begingroup
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%

    \@makefntext{%
        \ignorespaces#1%
    }%
```

Don't add the closing paragraph tag if are inside a lateximage:

```
    \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
    {}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
```

```
    \color@endgroup%
}% vbox
}
```

49.6 Printing pending footnotes

\LWR@printpendingfootnotes Enclose the footnotes in a class, print, then clear.

```
\newcommand*{\LWR@printpendingfootnotes}{%
\ifvoid\LWR@footnotes\else
    \LWR@forcenewpage
    \begin{BlockClass}{footnotes}
    \LWR@origmedskip
    \unvbox\LWR@footnotes
    \setbox\LWR@footnotes=\vbox{}
    \end{BlockClass}
\fi
}
```

\LWR@maybeprintpendingfootnotes {⟨*depth*⟩} Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
\newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
\ifboolexpr{
    not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
    bool{FormatEPUB} or
    bool{FormatWP}
}%
{\LWR@printpendingfootnotes}%
{}%
}
```

\LWR@printpendingmpfootnotes Enclose the minipage footnotes in a class, print, then clear.

```
\newcommand*{\LWR@printpendingmpfootnotes}{%
\ifvoid\LWR@mpfootnotes\else
    \LWR@forcenewpage
    \begin{BlockClass}{footnotes}
    \LWR@origvspace*{\baselineskip}
    \unvbox\LWR@mpfootnotes
    \setbox\LWR@mpfootnotes=\vbox{}
    \end{BlockClass}
\fi
}
```

```
\end{warpHTML}
```

50 Marginpars

`\marginpar` `[`⟨*left*⟩`] {`⟨*right*⟩`}` `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `<br />` tags.

`\marginparBlock` `[`⟨*left*⟩`] {`⟨*right*⟩`}` To include block-related macros, use `\marginparBlock`, which takes the same arguments but creates a `<div>` instead of a `<span>`. A line break will occur in the text where the `\marginBlock` occurs.

for HTML output:

```
\begin{warpHTML}
```

`\marginpar` `[`⟨*left*⟩`] {`⟨*right*⟩`}`

```
\renewcommand{\marginpar}[2][]{%
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
    \LWR@htmlspanclass{marginpar}{#2}%
}%
}
```

`\marginparBlock` `[`⟨*left*⟩`] {`⟨*right*⟩`}`

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```
\newcommand{\marginparBlock}[2][]{%
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
\begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
```

```
#2
\end{BlockClass}
}%
}
```

\reversemarginpar

```
\renewcommand*{\reversemarginpar}{}
```

\normalmarginpar

```
\renewcommand*{\normalmarginpar}{}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

\marginparBlock `[`⟨*left*⟩`]` {⟨*right*⟩}

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

Print version.

```
\LetLtxMacro\marginparBlock\marginpar
```

```
\end{warpprint}
```

51 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenamenoblanks`.
- `\LWR@newhtmlfile` finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT:

```
\begin{warpall}
```

Ctr FileDepth {⟨*section depth*⟩} determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
\newcounter{FileDepth}
\setcounter{FileDepth}{-5}
```

Bool `CombineHigherDepths` Combile higher-level sections together into one file?

```
\newbool{CombineHigherDepths}
\booltrue{CombineHigherDepths}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@thisfilename` The currently-active filename or number.

```
\newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
\newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamenoblanks` {⟨*filename*⟩}

Convert blanks into dashes, removes short words, store result in `\LWR@thisfilename`.

⚠ Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` LaTeX commands which appear in section names and TOC captions.

```
\newcommand*{\LWR@filenamenoblanks}[1]{%
\begingroup
```

Locally temporarily disable direct-formatting commands, not used in filenames:

```
\LWR@nullfonts%
\renewcommand*{\LWR@htmltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
\LWR@traceinfo{LWR@filenamenoblanks \#1: !#1!}%
\edef\LWR@thisnewfilename{#1}%
\LWR@traceinfo{LWR@filenamenoblanks edef: !\LWR@thisnewfilename!}%
\fullexpandarg%
```

Convert spaces into hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\LWR@origampersand}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textless}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textgreater}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]

\StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]

\StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textbackslash}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{~{}}{-}[\LWR@thisnewfilename]
%     "~{}" for babel
\StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{`}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
\StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
```

```
\StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
%     emdash
\StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
%     endash
\global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
\endgroup
}
```

`\LWR@newhtmlfile` {⟨*section name*⟩}

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and CSS are, then adds navigation, side TOC, header, and starts the text body.

```
\newcommand*{\LWR@newhtmlfile}[1]{
\LWR@traceinfo{LWR@newhtmlfile: !#1!}
```

At the bottom of the ending file:

```
\LWR@htmlelementclassend{section}{textbody}

\LWR@printpendingfootnotes

```

No footer between files if EPUB:

```
\ifbool{FormatEPUB}
{}
{
    \LWR@htmlelement{footer}
```

```
    \LWR@pagebottom

    \LWR@htmlelementend{footer}
}
```

No bottom navigation if are finishing the home page or formatting for EPUB or a word-processor.

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
\LWR@stoppars
\LWR@htmltag{/body}\LWR@orignewline
\LWR@htmltag{/html}\LWR@orignewline
\LWR@orignewpage

\addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into `\LWR@thisfilename`. If not using a filename, the file number will be used instead.

```
\ifbool{FileSectionNames}%
{\LWR@filenamenoblanks{#1}}
{\renewcommand*{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}}
```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
\LWR@traceinfo{LWR@newhtmlfile: about to print start file}%
\LWR@htmlblockcomment{%
|Start file|%
\LWR@htmlsectionfilename{\LWR@thisfilename}|%
}
```

At the top of the starting file:

```
\LWR@stoppars


\LWR@filestart{ -- #1}% there is an EMdash in front of the #1

```

No navigation between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@topnavigation}

```

No header if between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \LWR@htmlelement{header}

    \LWR@pagetop

    \LWR@htmlelementend{header}
}

```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\ifcsvoid{thetitle}{}{\LWR@printthetitle}}

```

No sideTOC if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@sidetoc}

```

Start of the <textbody>:

```
\LWR@htmlelementclass{section}{textbody}

```

Keep paragraph tags disabled for now:

```
\LWR@stoppars

```

Track the page numbers:

```
\setcounter{LWR@latestautopage}{\value{page}}
\LWR@traceinfo{LWR@newhtmlfile: done}
}
```

```
\end{warpHTML}
```

52 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing LaTeX code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

Unicode

⚠

Section names and the resulting filenames with accented characters are partially supported, depending on the ability of `pdflatex` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that `pdflatex` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.

XeLaTeX and LuaLaTeX directly support accented section and file names.

for HTML output:

```
\begin{warpHTML}
```

52.1 User-level starred section commands

`\ForceHTMLPage`

For HTML output, forces the next section to be on its own HTML page, if `FileDepth` allows, even if starred. For use with `\printindex` and others which generate a starred section which should be on its own HTML page. Also see `\ForceHTMLTOC`.

For print output, no effect.

```
\newbool{LWR@forcinghtmlpage}
\boolfalse{LWR@forcinghtmlpage}

\newcommand*{\ForceHTMLPage}{%
\global\booltrue{LWR@forcinghtmlpage}%
}
```

`\ForceHTMLTOC`

For HTML output, forces the next section to have a TOC entry, even if starred. For use with `\printindex` and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with tocbibind. Also see `\ForceHTMLPage`.

For print output, no effect.

```
\newbool{LWR@forcinghtmltoc}
\boolfalse{LWR@forcinghtmltoc}
```

```
4334
4335 \newcommand*{\ForceHTMLTOC}{%
4336 \global\booltrue{LWR@forcinghtmltoc}%
4337 }

4338 \end{warpHTML}
```

for PRINT output:

```
4339 \begin{warpprint}
4340 \newcommand*{\ForceHTMLPage}{}
4341 \newcommand*{\ForceHTMLTOC}{}
4342 \end{warpprint}
```

for HTML output:

```
4343 \begin{warpHTML}
```

52.2 Book class commands

`\mainmatter` ⚠ Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
4344 \newbool{LWR@mainmatter}
4345 \DeclareDocumentCommand{\mainmatter}{}{%
4346 \booltrue{LWR@mainmatter}%
4347 }
```

`\frontmatter` Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
4348 \DeclareDocumentCommand{\frontmatter}{}{%
4349 \boolfalse{LWR@mainmatter}%
4350 }
```

`\backmatter` Declare the back matter section of the document. Does not reset the page number.

```
4351 \DeclareDocumentCommand{\backmatter}{}{%
4352 \boolfalse{LWR@mainmatter}
4353 }
```

52.3 Sectioning support macros

`\LWR@sectionnumber` {⟨*section type*⟩}

Typeset a section number and its trailing space with CSS formatting:

```
\newcommand*{\LWR@sectionnumber}[1]{%
\InlineClass{sectionnumber}{#1}%
}
```

`autosec` A tag used by the TOC and index.

`\LWR@createautosec` {⟨*section type*⟩}

Create an autosection tag.

```
\newcommand*{\LWR@createautosec}[1]{%
\LWR@htmltag{#1 id="autosec-\arabic{page}"}%
}
```

`\LWR@pushoneclose` {⟨*depth*⟩} {⟨*printclose*⟩} Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

⚠ `\LWR@stoppars` must be executed first.

```
\NewDocumentCommand{\LWR@pushoneclose}{m m}{%
\LWR@traceinfo{LWR@pushoneclose #1}%
    \pushclose{#2}{#1}%
}
```

`\LWR@startnewdepth` {⟨*depth*⟩} {⟨*printclose*⟩}

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

⚠ `\LWR@stoppars` must be executed first.

```
\NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
\LWR@closeprevious{#1}%
```

Push a new section depth:

```
\LWR@pushoneclose{#1}{#2}%
}
```

Ctr `LWR@prevFileDepth` Remembers the previous `LWR@FileDepth`.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
\newcounter{LWR@prevFileDepth}
\setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\@seccntformat` {⟨*sectiontype*⟩}

```
\def\@seccntformat#1{\csname the#1\endcsname\protect\quad}
```

`\simplechapterdelim` Used by tocbibind and anonchap.

```
\newcommand*{\simplechapterdelim}{}
```

`\@chapcntformat` {⟨*sectiontype*⟩}

`\let` to `\@seccntformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from tocbibind or anonchap.

```
\let\@chapcntformat\@seccntformat
```

Ctr `LWR@currentautosec` Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following autosec label is created.

```
\newcounter{LWR@currentautosec}
```

`\LWR@section` * [⟨*TOC name*⟩] {⟨*name*⟩} {⟨*sectiontype*⟩}

The common actions for the high-level sectioning commands.

```
\DeclareDocumentCommand{\LWR@section}{m m m m}{%
\LWR@traceinfo{LWR@section |#2| |#3|}%
\LWR@traceinfo{LWR@section: not an empty section}%
\LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
\global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for `\part*`, for appendix.

Generate a new LaTeX page so that TOC and index page number points to the section:

```
\LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
\IfBooleanT{#1}{\LWR@traceinfo{LWR@section: starred}}%
\ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
\ifthenelse{%
    \(%
        \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
        \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
        \(\boolean{LWR@forcinghtmlpage}\)%
    \)%
    \AND%
    \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
    \AND%
    \(%
        \NOT\boolean{CombineHigherDepths}\OR%
        \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
    \)%
    \AND%


    \(% phantomsection
        \NOT\isempty{#3}%
        \OR%
        \(\NOT\equal{#1}{\BooleanTrue}\)%
    \)%
}%
```

If so: start a new HTML file:

```
{% new file
    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
    \IfNoValueTF{#2}%
```

If no optional entry

```
    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
    {\LWR@newhtmlfile{#2}}%
}% new file
```

Else: No new HTML file:

```
{% not new file
```

Generate a new LaTeX page so that TOC and index page number points to the section:

```
    \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
    \LWR@orignewpage%

}% not new file
```

Remember this section's name for \nameref:

```
\IfValueT{#3}{%
\LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
\IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
}%
```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```

\ifbool{HTMLDebugComments}{%
    \begingroup%
    \LWR@nullfonts%
    \LWR@htmlcomment{Opening #4 ``#3''{}}%
    \endgroup%
}{}%

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
\ifthenelse{%
    \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

Create the opening tag with an autosec:

```
\LWR@traceinfo{LWR@section: about to LWR@createautosec}%
\LWR@createautosec{\csuse{LWR@tag#4}}%


\setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
\IfBooleanTF{#1}%
{%
\LWR@traceinfo{LWR@section: starred}%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
\ifbool{LWR@forcinghtmltoc}%
{\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
{}%
}% starred
```

Not starred, so step counter and add to TOC:

```
{% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
    \ifthenelse{%
        \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
    }%
    {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For `article` class, lwarp assumes that all is mainmatter.

```
        \LWR@traceinfo{LWR@section: about to test main matter}%
        \ifbool{LWR@mainmatter}%
        {%
            \LWR@traceinfo{LWR@section: yes mainmatter}%
            \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
            \LWR@traceinfo{LWR@section: about to addcontentsline}%
            \addcontentsline{toc}{#4}%
            {%
                 \protect\numberline{\csuse{the#4}}%
                {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
            }%
            \LWR@traceinfo{LWR@section: finished addcontentsline}%
        }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
        {% not main matter
            \LWR@traceinfo{LWR@section: no main matter}%
            \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
        }% end of not main matter
    }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
    {%
        \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
    }%
```

For part, print the section type:

```
    \ifbool{LWR@mainmatter}%
    {%
        \ifthenelse{%
            \(\cnttest{\csuse{LWR@depth#4}}{<=}%
                {\value{secnumdepth}}\) \AND%
            \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)%
        }%
        {\csuse{#4name}~{}}%
        {}%
```

Print the section number:

```
        \LWR@traceinfo{LWR@section: about to print section number}%
        \ifthenelse{%
            \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
        }%
        {%
            \ifstrequal{#4}{chapter}%
            {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
            {\protect\LWR@sectionnumber{\@seccntformat{#4}}}%
        }%
        {}%
        \LWR@traceinfo{LWR@section: finished print section number}%
    }{}%
}% end of not starred
```

Print the section name:

```
\LWR@traceinfo{LWR@section: about to print the section name}%
#3%
```

Close the heading tag, such as /H2:

```
\LWR@traceinfo{LWR@section: about to close the heading tag}%
\LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a LaTeX label:

```
\LWR@traceinfo{LWR@section: about to create the LaTeX label}%
\label{autopage-\theLWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
\ifthenelse{%
    \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

If not starred, remember the previous depth to possibly trigger a new HTML page.

HOWEVER, allow a `\part*` to start a new HTML page. This is used by appendix.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when a `\listoftables` was tested in the middle of the document. The `\chapter*` for the list was not allowing a new HTML page for the section following it while `CombineHigherDepths` was true.

```
\ifthenelse{%
    \NOT\equal{#1}{\BooleanTrue}\OR%
    \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
}%
{% not starred
    \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
}% not starred
{}%
```

Reset to defaults if not a phantomsection:

```
\ifstrempty{#3}%
{}%
{%
\global\boolfalse{LWR@forcinghtmlpage}%
\global\boolfalse{LWR@forcinghtmltoc}%
}%
%
\LWR@traceinfo{LWR@section: done}%
}
```

52.4 \section and friends

`\part` `*` `[`⟨*TOC name*⟩`]` `{`⟨*name*⟩`}`

```
\newcommand{\part@preamble}{}% for koma-script

\DeclareDocumentCommand{\part}{s o m}{%
```

```
\LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%

\LWR@section{#1}{#2}{#3}{part}%

\part@preamble% for koma-script
\renewcommand{\part@preamble}{}%
}
```

`\chapter` * [⟨*TOC name*⟩] [⟨*heading name*⟩] {⟨*name*⟩}

```
\let\@printcites\relax% for quotchap package

\newcommand{\chapter@preamble}{}% for koma-script

\@ifundefined{chapter}
{}
{%
\DeclareDocumentCommand{\chapter}{s o o m}{%
\IfValueTF{#2}{
\LWR@traceinfo{chapter #2}%
}{
\LWR@traceinfo{chapter #4}%
}
\LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%

\LWR@section{#1}{#2}{#4}{chapter}%

\@printcites% for quotchap package

\chapter@preamble% for koma-script
\renewcommand{\chapter@preamble}{}%
}
}
```

`\section` * [⟨*TOC name*⟩] [⟨*heading name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\section}{s o o m}{%
\IfValueTF{#2}{
\LWR@traceinfo{section #2}%
}{
\LWR@traceinfo{section #4}%
}
```

```
\LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%

\LWR@section{#1}{#2}{#4}{section}%
}
```

`\subsection` `* [⟨TOC name⟩] {⟨name⟩}`

```
\DeclareDocumentCommand{\subsection}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%

\LWR@section{#1}{#2}{#3}{subsection}%
}
```

`\subsubsection` `* [⟨TOC name⟩] {⟨name⟩}`

```
\DeclareDocumentCommand{\subsubsection}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsubsubsection}%
{\LWR@printclosesubsubsection}%

\LWR@section{#1}{#2}{#3}{subsubsection}%
}
```

`\paragraph` `* [⟨TOC name⟩] {⟨name⟩}`

```
\DeclareDocumentCommand{\paragraph}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%

\LWR@section{#1}{#2}{#3}{paragraph}%
}
```

`\subparagraph` `* [⟨TOC name⟩] {⟨name⟩}`

```
\DeclareDocumentCommand{\subparagraph}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
\LWR@stoppars%
```

```
4596
4597 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4598
4599 \LWR@section{#1}{#2}{#3}{subparagraph}%
4600 }

4601 \end{warpHTML}
```

53 Starting a new file

for HTML & PRINT:

```
4602 \begin{warpall}
```

`\HTMLLanguage` Default language for the HTML `lang` tag.

```
4603 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4604
4605 \newcommand*{\HTMLLanguage}[1]{%
4606 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4607 }

4608 \end{warpall}
```

for HTML output:

```
4609 \begin{warpHTML}
```

`\LWR@filestart` {⟨*title_suffix*⟩}

Creates the opening HTML tags.

```
4610 \newcommand*{\LWR@filestart}[1]{
4611 \LWR@traceinfo{LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4612 \begingroup
4613 \LWR@traceinfo{LWR@filestart: A}
4614 \LWR@nullfonts
4615 \LWR@traceinfo{LWR@filestart: B}
```

Create the page's HTML header:

```
4616 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4617 \LWR@traceinfo{LWR@filestart: C}
```

The language is user-adjustable:

```
\LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
\LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
\LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
\ifthenelse{\equal{\theHTMLAuthor}{}}%
{}%
{\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline}%
```

lwarp is the generator:

```
\LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
    \LWR@orignewline
```

If there is a description, add it now:

```
\ifdefempty{\LWR@currentHTMLDescription}{}{%
\LWR@htmltag{%
meta name="description" content="\LWR@currentHTMLDescription" /}%
    \LWR@orignewline
}%
```

Mobile-friendly viewport:

```
\LWR@htmltag{meta name="viewport" %
content="width=device-width, initial-scale=1.0" /}%
    \LWR@orignewline
```

IE patch:

```
\LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
\LWR@htmltag{%
script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"}%
\LWR@htmltag{/script}\LWR@orignewline
\LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
\ifthenelse{\equal{\theHTMLTitle}{}}%
{}%
{\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%
```

The page's stylesheet:

```
\LWR@htmltag{%
link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
\LWR@orignewline
```

Optional MathJax support. The HTML tags must be turned off during the verbatim input, and the paragraph handling which was turned on at the end of verbatim input must be immediately turned off again.

```
\ifbool{mathjax}%
{%
    \begingroup%
    \LWR@restoreoriglists%
    \boolfalse{LWR@verbtags}
    \verbatiminput{lwarp_mathjax.txt}%
    \booltrue{LWR@verbtags}
    \endgroup%
    \LWR@stoppars
}% end of mathjax
{}%
```

End of the header:

```
\LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
\LWR@htmltag{body}\LWR@orignewline
\endgroup
\LWR@traceinfo{LWR@filestart: done}
}
```

```
\end{warpHTML}
```

54 Starting HTML output

for HTML output:

```
\begin{warpHTML}
```

`\LWR@LwarpStart` Executed at the beginning of the entire document.

```
\catcode`\$=\active
\newcommand*{\LWR@LwarpStart}
{%
\LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
\ifbool{FormatWP}{%
    \setcounter{FileDepth}{-5}%
    \boolfalse{HTMLDebugComments}%
}{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
\edef\LWR@strresult{\HomeHTMLFilename}
\edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
\edef\LWR@strresult{\HTMLFilename}
\edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
\LWR@origonecolumn%
\LWR@origpagestyle{empty}%
```

Reduce chance of line overflow in verbatim environments:

```
\LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
\LWR@origraggedright%
\LetLtxMacro{\\}{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
\linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{2ex}
```

For the lateximages record file:

```
\immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
\setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
\renewcommand{\ps@plain}{}
```

\centering
\raggedleft
\raggedright

Not used in the HTML environment:

```
\renewcommand*{\centering}{}
\renewcommand*{\raggedleft}{}
\renewcommand*{\raggedright}{}
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
\LetLtxMacro{\LWR@origtabular}{\tabular}
\LetLtxMacro{\LWR@origendtabular}{\endtabular}
\LetLtxMacro{\tabular}{\LWR@tabular}
\LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
\let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in amsmath environments and is also patched by cleveref.

Label in HTML

```
\let\LWR@origltx@label\ltx@label
\let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
\let\TeX\LWR@TeX
\let\LaTeX\LWR@LaTeX
\let\LuaTeX\LWR@LuaTeX
\let\LuaLaTeX\LWR@LuaLaTeX
\let\XeTeX\LWR@XeTeX
\let\XeLaTeX\LWR@XeLaTeX
\let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
\global\boolfalse{LWR@doingapar}
\global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
\mainmatter
\LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
\LWR@traceinfo{LWR@lwarpStart: Starting new file.}
\LWR@filestart{}
\LWR@traceinfo{LWR@lwarpStart: Generating first header.}
\LWR@htmltag{header}\LWR@orignewline
\LWR@startpars
\LWR@firstpagetop
\LWR@stoppars
\LWR@htmltag{/header}\LWR@orignewline
\LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
\LWR@htmltag{section class="textbody"}
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native LaTeX environments, as well as those provided by enumitem, enumerate, and paralist.

```
\LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
\catcode`\$=\active
```

Allow HTML paragraphs to begin:

```
\LWR@startpars
\LWR@traceinfo{LWR@lwarpStart: done}
}
\catcode`\$=3% math shift until lwarp starts

\end{warpHTML}
```

55 Ending HTML output

for HTML output:

```
\begin{warpHTML}
```

`\LWR@requesttoc` {⟨*boolean*⟩} {⟨*suffix*⟩} Requests that a toc, lof, or lot be generated.

```
\newcommand*{\LWR@requesttoc}[2]{%
```

```
\ifbool{#1}
{
    \expandafter\newwrite\csuse{tf@#2}
    \immediate\openout \csuse{tf@#2} \jobname.#2\relax
}{}
}
```

\LWR@LwarpEnd Final stop of all HTML output:

```
\newcommand*{\LWR@LwarpEnd}
{
\LWR@stoppars
\LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
\LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
\LWR@printpendingfootnotes
```

Create the footer:

```
\LWR@htmlelement{footer}

\LWR@pagebottom

\LWR@htmlelementend{footer}
```

No bottom navigation if are finishing the home page, or if formatting for an EPUB or word processor.

Presumably has a table-of-contents.

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
}
```

```
\LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
\LWR@htmltag{/body}\LWR@orignewline
\LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
\LWR@orignewpage
```

For `lateximage` commands:

```
\immediate\closeout\LWR@lateximagesfile
}
```

```
\end{warpHTML}
```

56 Title page

package support
⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

`\published` and `\subtitle`

If using the titling package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 56.7.

affiliation

lwarp provides for the `\author` macro an additional `\affiliation` macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using titlingpage's `\theauthor` in the main text.

reusing titlepage information

The titling package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for `\author` the `\and` is replaced to generate a simple inline list of authors separated by commas. Note: `\theauthor` does not work well with authblk unless the traditional LaTeX syntax is used.

⚠ `\theauthor`, authblk

custom titlepages

`\printtitle`, `\printauthor`, etc., are provided for use inside a custom `titlepage` or `titlingpage` environment, and these retain the `\thanks` and `\affiliation`.

`\printthanks`

`\printthanks` has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` is not used.

⚠

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` instead of `\footnote` for acknowledgements, etc.

56.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

`\author` `\and` {⟨*author*⟩} While using `\maketitle` and print mode, the author is treated as a single-column `tabular` and the `\and` feature finishes the current `tabular` then starts a new one for the next author. Each author thus is placed into its own `tabular`, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a `<div>` of class `author`, and each individual author is inside a `<div>` of class `oneauthor`.

`\@title` `\@author` `\@date` `\@title`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column `tabular`, which becomes multiple single-column `tabular`s if multiples authors are included. For HTML these `tabular`s become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

`\printtitle` `\printauthor` `\printdate` `\printtitle`, etc. are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\theauthor` `\thedate` `\thetitle`, `\theauthor`, and `\thedate` are available if titling has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\HTMLPageBottom` {⟨*text*⟩}

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

⚠ `\theauthor` `\theauthor` does not work well if authblk is used. If `\theauthor` is important, it is recommended to use the standard LaTeX syntax for `\author`, optionally with lwarp's `\affiliation` macro as well.

⚠ affiliations After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors

names separated by spaces. This fails when affiliations are included on their own table rows.

\affiliation A solution, provide here, is to define a macro \affiliation which, during \maketitle, starts a new row and adds the affiliation, but after \maketitle is finished \affiliation is re-defined to discard its argument, thus printing only the author names when \author is later used inline.

56.2 Changes for \affiliation

\affiliation {⟨*text*⟩}

Adds the affiliation to the author for use in \maketitle.

Inside titlepage, this macro prints its argument. Outside, it is null.

for HTML & PRINT:

```
\begin{warpall}
\newrobustcmd{\affiliation}[1]{}
\end{warpall}
```

for PRINT output:

```
\begin{warpprint}

\AtBeginEnvironment{titlepage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}

\AtBeginDocument{
\@ifpackageloaded{titling}{
\AtBeginEnvironment{titlingpage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}
}{}% titling loaded
}% AtBeginDocument

\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
```

Env titlepage Sets up a of class titlepage. Provided even for memoir class, since it is used by \maketitle.

```
\DeclareDocumentEnvironment{titlepage}{}
{
\renewrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{##1}}
\LWR@printpendingfootnotes
```

```
\LWR@forcenewpage
\BlockClass{titlepage}
}
{
\endBlockClass
\LWR@printpendingfootnotes
}

\end{warpHTML}
```

56.3 Printing the thanks

for HTML & PRINT:

```
\begin{warpall}
```

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
\newcommand*{\printthanks}{\@thanks}

\end{warpall}
```

56.4 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

for HTML output:

```
\begin{warpHTML}
```

`\printtitle`

```
\newcommand*{\printtitle}
{
\LWR@stoppars
\LWR@htmltag{\LWR@tagtitle}%
\@title%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
\newcommand*{\LWR@printthetitle}
{
\LWR@stoppars
\LWR@htmltag{\LWR@tagtitle}%
\thetitle%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

\printauthor HTML version.

```
\newcommand*{\printauthor}{
```

The entire author block is contained in a named author:

```
\begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
\renewcommand{\and}{%
\end{BlockClass}
\begin{BlockClass}{oneauthor}
}
```

Individual authors are contained in a named oneauthor:

```
\begin{BlockClass}{oneauthor}
\@author
\end{BlockClass}
\end{BlockClass}
}
```

\printdate

```
\newcommand*{\printdate}{%
\begin{BlockClass}{titledate}
\@date
\end{BlockClass}
}

\end{warpHTML}
```

56.5 Printing the title, etc. in print form

The following are for printing the title, etc. in a titlepage or a titlingpage in print form:

for PRINT output:

```
\begin{warpprint}
```

\printtitle

```
\newcommand*{\printtitle}{{\Huge\@title}}
```

\printauthor Print mode.

```
\newcommand*{\printauthor}
    {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

\printdate

```
\newcommand*{\printdate}{{\small\textit{\@date}}}
```

```
\end{warpprint}
```

56.6 \maketitle for HTML output

An HTML of class titlepage is used.

\thanks are a form of footnotes used in the title page. See section 49 for other kinds of footnotes.

See \thanksmarkseries{series}, below, to set the style of the footnote marks.

for HTML output:

```
\begin{warpHTML}
```

```
\@ifclassloaded{memoir}
{
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@arabic\c@footnote}%
}
}{% not memoir
\if@titlepage
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@arabic\c@footnote}%
}
\else
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
}
\fi
}% not memoir
```

\LWR@maketitlesetup Patches \thanks macros.

```
\newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\LWR@setfootnoteseries%
\def\@makefnmark{\textsuperscript{\thefootnote}}
```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\textsuperscript{\@thefnmark}~%
```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

Print the text:

```
##1%
}%
}
```

\@fnsymbol {⟨*counter*⟩}

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by pdftotext.

```
\def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
   \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
   **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
   \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}
```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage is used for all classes.

```
\begin{titlepage}
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Close the HTML titlepage div and cleanup:

```
\end{titlepage}
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@maketitle\relax
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@date\@empty
\global\let\@title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}
```

\@maketitle HTML mode. Typesets the title, etc.:

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
    \@title
    \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    \begin{BlockClass}{author}
        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
            \@author
        \end{BlockClass}
    \end{BlockClass}
```

```
4888     \begin{BlockClass}{titledate}
4889     \@date
4890     \end{BlockClass}
4891 }
```

\LWR@titlingmaketitle `\maketitle` for use inside an HTML `titlingpage` environment.

```
4892 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
4893 \@thanks
```

Set up special patches:

```
4894 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4895 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4896 \@thanks
4897 }
```

```
4898 \end{warpHTML}
```

56.7 `\published` and `\subtitle`

\subtitle and \published To add `\subtitle` and `\published` to the titlepage, load the titling package and use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

⚠ titling hooks After titling is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks. Do not use `\AddSubtitlePublished` if the user has patched the titling hooks for some other reason. Portions are marked `\warpprintonly` to reduce extra tags in HTML. Similarly, `BlockClass` has no effect in print mode. Thus, the following may be marked `warpall`.

for HTML & PRINT:

```
4899 \begin{warpall}
```

\AddSubtitlePublished Adds \published and \subtitle, and related.

```
\AfterPackage{titling}{
\newcommand*{\AddSubtitlePublished}{%

\newcommand{\@published}{}

\newcommand{\published}[1]{\gdef\@published{##1}}

\renewcommand*{\maketitlehooka}{\printpublished}

\newcommand*{\printpublished}{%
\warpprintonly{\begin{center}\unskip}%
\begin{BlockClass}{published}%
\warpprintonly{\large\itshape}%
\@published%
\end{BlockClass}%
\warpprintonly{\end{center}}%
}

\newcommand{\@subtitle}{}

\newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}

\renewcommand*{\maketitlehookb}{\printsubtitle}

\newcommand*{\printsubtitle}{%
\warpprintonly{\begin{center}\unskip}%
\begin{BlockClass}{subtitle}%
\warpprintonly{\Large\itshape}%
\@subtitle%
\end{BlockClass}%
\warpprintonly{\end{center}}%
}

}% \AddSubtitlePublished
}% AfterPackage


\end{warpall}
```

57 Abstract

The following code replaces the LaTeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output:

```
\begin{warpHTML}
```

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```
\providecommand*{\abstractname}{Abstract}
```

Env abstract

```
\DeclareDocumentEnvironment{abstract}{}
{
\LWR@forcenewpage
\BlockClass{abstract}
\BlockClassSingle{abstracttitle}{\abstractname}
}
{
\endBlockClass
}

\end{warpHTML}
```

58 Quote and verse

58.1 Citations and attributions

\attribution for use inside quote, quotation, verse:

ex: \attribution{author name} --- \citetitle{book name}

for HTML output:

```
\begin{warpHTML}
\newcommand{\attribution}[1]{%
\InlineClass{attribution}{--\,#1}}% emdash
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand{\attribution}[1]{\textsc{---\,#1}}
\end{warpprint}
```

\citetitle for use inside quote, quotation, verse:

for HTML output:

```
\begin{warpHTML}
\newcommand{\citetitle}[1]{%
\InlineClass{citetitle}{--\,#1}}% emdash
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand{\citetitle}[1]{\textsl{---\,#1}}
\end{warpprint}
```

58.2 Quotes, quotations

for HTML output:

```
\begin{warpHTML}
```

Env `quote`

```
\renewenvironment*{quote}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquote}
}
{\LWR@htmlblocktag{/blockquote}}

\renewenvironment*{quotation}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquotation}
}
{\LWR@htmlblocktag{/blockquotation}}

\end{warpHTML}
```

58.3 Verse

`\attrib` The documentation for the verse and memoir packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. lwarp provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len `\leftskip`
Len `\leftmargini`
Len `\TMLvleftskip`
Len `\TMLleftmargini`

These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and

\HTMLleftmargini are provided to control the margins in HTML output. These new lengths may be set by the user before any verse environment, and persist until they are manually changed again. One reason to change \HTMLleftmargini is if there is a wide \flagverse in use, such as the word "Chorus", in which case the value of \HTMLleftmargini should be set to a wide enough length to contain "Chorus". The default is wide enough for a stanza number.

⚠ spacing Horizontal spacing relies on pdftotext's ability to discern the layout (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargini or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

for HTML & PRINT:

```
\begin{warpall}
```

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set \vleftskip and \leftmargini for print output, and optionally select different values for HTML.

Len \TMLvleftskip Sets \vleftskip inside a verse environment in HTML.

```
\newlength{\HTMLvleftskip}
\setlength{\HTMLvleftskip}{1em}
```

Len \TMLleftmargini Sets \leftmargini inside a verse environment in HTML.

```
\newlength{\HTMLleftmargini}
\setlength{\HTMLleftmargini}{4.5em}
```

```
\end{warpall}
```

59 Verbatim and tabbing

for HTML & PRINT:

```
\begin{warpall}
```

Len \VerbatimHTMLWidth Width to use in HTML Verbatim environment.

This width is used when placing line numbers to the right. Ignored during print output.

```
\newlength{\VerbatimHTMLWidth}
\setlength{\VerbatimHTMLWidth}{4in}
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
\newbool{LWR@verbtags}
\booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` `[`⟨*style*⟩`] {`⟨*class*⟩`}`

Encloses a verbatim environment with the given CSS class.

```
\newcommand*{\LWR@atbeginverbatim}[2][]
{%
```

Avoid excessive space between lines:

```
\setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
\LWR@stoppars%
```

Create a new `pre` of the given class. The tags may temporarily be turned off for internal use, such as loading the MathJax script.

```
\ifbool{LWR@verbtags}{%
    \LWR@htmltag{pre class="#2"
    \ifthenelse{\equal{#1}{}}{}{style="#1"}%
}\LWR@orignewline% pre
}{}%
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the CSS for the `verse` class.

```
\begingroup%
\LWR@origttfamily%
```

Since inside a `<pre>`, restore the original list processing:

```
\LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
\LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain LaTeX `\hspace` functionality:

```
\LetLtxMacro{\hspace}{\LWR@orighspace}%
}
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
\newcommand*{\LWR@afterendverbatim}{%

\endgroup%
```

At the end of the environment, close the `pre`:

```
\ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}

}{}%
```

Resume regular paragraph handling:

```
\LWR@startpars%
}
```

`\verbatiminput` {⟨*filename*⟩}

Patch `\verbatiminput` to add HTML tags:

```
\let\LWRV@origverbatim@input\verbatim@input

\renewcommand{\verbatim@input}[2]{%
\ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
\LWRV@origverbatim@input{#1}{#2}%
\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
```

Env `verbatim`

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verbatim.}
\AtBeginEnvironment{verbatim}{%
\LWR@forcenewpage%
\LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-2\baselineskip}%
}
\AfterEndEnvironment{verbatim}{%
    \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
}
```

Env `tabbing` The `tabbing` environment works, except that SVG math and `lateximages` do not yet work inside the environment.

```
\AfterEndPreamble{
\LWR@traceinfo{Patching tabbing.}
\AtBeginEnvironment{tabbing}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{tabbing}{%
    \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
}
}

\end{warpHTML}
```

60 Theorems

`\newtheorem` {⟨*text*⟩} [⟨*counter*⟩] -or- [⟨*oldname*⟩] {⟨*text*⟩}

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class theoremcontents.

for HTML output:

```
\begin{warpHTML}
```

`\@begintheorem` {⟨*name*⟩} {⟨*number*⟩}

```
\renewcommand{\@begintheorem}[2]{%
\LWR@forcenewpage
\BlockClass{theoremcontents}
\trivlist
\item[\InlineClass{theoremlabel}{#1\ #2\ }]\itshape
}
```

`\@opargbegintheorem` {⟨*name*⟩} {⟨*number*⟩} {⟨*oparg*⟩}

```
\renewcommand{\@opargbegintheorem}[3]{%
\LWR@forcenewpage
```

```
\BlockClass{theoremcontents}
\trivlist
\item[\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }]\itshape
}
```

\@endtheorem

```
\renewcommand*{\@endtheorem}{%
\endtrivlist
\endBlockClass% theoremcontents
}
```

```
\end{warpHTML}
```

61 Lists

The environments `itemize`, `enumerate`, and `description` are patched when lwarp is started. These patches support the standard LaTeX environments, as well as those of enumerate, enumitem, and paralist, and at least the French version of babel. Additional patches are done on a package-specific basis.

The LaTeX source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.

empty item

To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

```
begin{itemize}
item \mbox{}
    \begin{itemize}
...
    \end{itemize}
item \
    \begin{itemize}
  ...
    \end{itemize}
```

61.1 List environment

for HTML output:

```
\begin{warpHTML}
```

The following may be locally redefined by `enumerate` or `description`.

```
\newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\newcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
```

`\@mklab` Removes PDF spacing.

```
\def\@mklab#1{%
%     \hfil %
    #1}
```

`\@donoparitem` Modified for HTML output by replacing TEX boxes with plain text. Also removes PDF spacing.

```
\def\@donoparitem{%
  \@noparitemfalse
%   \global\setbox\@labels\hbox{\hskip -\leftmargin
%                               \unhbox\@labels
%                                \hskip \leftmargin}%
%   \if@minipage\else
%     \@tempskipa\lastskip
%     \vskip -\lastskip
%     \advance\@tempskipa\@outerparskip
%     \advance\@tempskipa -\parskip
%     \vskip\@tempskipa
%   \fi
}
```

`\@item` Modified for HTML output by replacing TEX boxes with plain text. Also removes PDF spacing.

```
\def\@item[#1]{%
  \if@noparitem
    \@donoparitem
  \else
%     \if@inlabel
%       \indent
        \par
%     \fi
    \ifhmode
%       \unskip\unskip
        \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
```

```
%         \addpenalty\@beginparpenalty
%         \addvspace\@topsep
%         \addvspace{-\parskip}%
      \fi
    \else
%       \addpenalty\@itempenalty
%       \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
%   \everypar{%
    \@minipagefalse
    \global\@newlistfalse

%     \if@inlabel
%       \global\@inlabelfalse

%       {\setbox\z@\lastbox
%        \ifvoid\z@
%          \kern-\itemindent
%        \fi}%

%         \LWR@@labels
%         \renewcommand{\LWR@@labels}{}
%       \box\@labels
%       \penalty\z@
%     \fi

%     \if@nobreak
%       \@nobreakfalse
%       \clubpenalty \@M
%     \else
%       \clubpenalty \@clubpenalty
%       \everypar{}%
%     \fi}%

  \if@noitemarg
    \@noitemargfalse
    \if@nmbrlist

      \refstepcounter\@listctr
    \fi
  \fi

%     \appto{\LWR@@labels}{\makelabel{#1}}%
%     \makelabel{#1}
        #1
%   \sbox\@tempboxa{\makelabel{#1}%
```

```
%   \global\setbox\@labels\hbox{%
%     \unhbox\@labels
%     \hskip \itemindent
%     \hskip -\labelwidth
%     \hskip -\labelsep
%     \ifdim \wd\@tempboxa >\labelwidth
%       \box\@tempboxa

%     \else
%       \hbox to\labelwidth {\unhbox\@tempboxa}%
%     \fi
%     \hskip \labelsep}%
  \ignorespaces%
}
```

`\@nbitem`

```
\def\@nbitem{%
%   \@tempskipa\@outerparskip
%   \advance\@tempskipa -\parskip
%   \addvspace\@tempskipa
}
```

`\LWR@listitem` [⟨*label*⟩]

Handles `\item` inside a `list`, `itemize`, or `enumerate`.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@listitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

Env `list` {⟨*label*⟩} {⟨*commands*⟩}

```
\newcommand*{\LWR@liststart}{%
\LWR@traceinfo{LWR@liststart}%
\LWR@stoppars%
\LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
\LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
\LWR@startpars%
\setlength{\topsep}{0pt}%
```

```
\setlength{\partopsep}{0pt}%
\setlength{\itemsep}{0pt}%
\setlength{\parsep}{0pt}%
\setlength{\leftmargin}{0pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\listparindent}{0pt}%
\setlength{\itemindent}{0pt}%
\setlength{\labelsep}{1em}%
}

\newcommand*{\LWR@listend}{%
\LWR@traceinfo{LWR@listend}%
\LWR@stoppars%
\LWR@closeprevious{\LWR@depthlist}%
\LWR@startpars%
}
```

61.2 Itemize

`\LWR@itemizeitem` [⟨*label*⟩]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@itemizeitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

Env `itemize` [⟨*options*⟩]

```
\newcommand*{\LWR@itemizestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
\let\item\LWR@itemizeitem%
}
```

61.3 Enumerate

An HTML unordered list is used with customized LaTeX-generated labels.

Env `enumerate` [⟨*options*⟩]

```
\newcommand*{\LWR@enumeratestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}
\let\item\LWR@itemizeitem%
}
```

61.4 Description

`\LWR@descitem` [⟨*label*⟩] Handles an `\item` inside a description.

```
\newcommand*{\LWR@descitem}[1][]%
{%
\LWR@stoppars%
\LWR@setlatestname{#1}%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem}%
```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for `descriptions` only. This causes lwarp to mistakenly place an empty span between HTML list tags.

```
\LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original `\item` code:

```
\LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
\LetLtxMacro{\hspace}{\LWR@hspace}%
\LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
\LWR@orignewline%
\LWR@htmltag{dd}%
\LWR@startpars%
}
```

Env `description` [⟨*options*⟩]

```
\newcommand*{\LWR@descriptionstart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
```

```
\renewcommand*{\LWR@printopenlist}{dl}
\let\item\LWR@descitem%
}
```

61.5 Patching the lists

```
\let\LWR@orig@trivlist\@trivlist
\let\LWR@origtrivlist\trivlist
\let\LWR@origendtrivlist\endtrivlist
```

`\LWR@patchlists` Patches list environments.

`\LWr@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native LaTeX environments, as well as those provided by enumitem, enumerate, and paralist.

```
\newcommand*{\LWR@patchlists}{%
    \let\item\LWR@listitem%
    \renewcommand*{\@trivlist}{%
        \LWR@traceinfo{@trivlist start}%
        \LWR@liststart%
        \LWR@orig@trivlist%
        \LWR@traceinfo{@trivlist done}%
    }%
    \renewcommand*{\endtrivlist}{%
        \LWR@traceinfo{endtrivlist start}%
        \LWR@origendtrivlist\LWR@listend%
        \LWR@traceinfo{endtrivlist done}%
    }%
    \renewcommand*{\trivlist}{%
        \LWR@traceinfo{trivlist}%
        \LWR@origtrivlist%
        \def\makelabel####1{\LWR@orighspace{3em}####1}%
    }%
    \renewcommand*{\itemize}{\LWR@itemizestart\LWR@origitemize}%
    \renewcommand*{\enumerate}{\LWR@enumeratestart\LWR@origenumerate}%
    \renewcommand*{\description}{\LWR@descriptionstart\LWR@origdescription}%
%     \AtBeginEnvironment{itemize}{\LWR@itemizestart}%
%     \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}%
%     \AtBeginEnvironment{description}{\LWR@descriptionstart}%
}
```

`\LWR@restoreoriglists` Restores the original `trivlist` environment.

```
\newcommand*{\LWR@restoreoriglists}{%
\let\@trivlist\LWR@orig@trivlist%
\let\trivlist\LWR@origtrivlist%
\let\endtrivlist\LWR@origendtrivlist%
\LetLtxMacro\item\LWR@origitem%
\LetLtxMacro\itemize\LWR@origitemize%
\LetLtxMacro\enditemize\LWR@endorigitemize%
\LetLtxMacro\enumerate\LWR@origenumerate%
\LetLtxMacro\endenumerate\LWR@endorigenumerate%
\LetLtxMacro\description\LWR@origdescription%
\LetLtxMacro\enddescription\LWR@endorigdescription%
}

\end{warpHTML}
```

62 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

62.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, siunitx S columns, or the packages multirow, longtable, supertabular, or xtab.

Defining environments:

⚠ misplaced alignment alignment tab character &

- When defining environments or macros which include `tabular` and instances of the & character, it may be necessary to make & active before the environment or macro is defined, then restore & to its default catcode after, using the following commands. These are are ignored in print mode.

  ```
  \StartDefiningTabulars
  <define macros or environments using tabular and &
  here>
  \EndDefiningTabulars
  ```

  ⚠ floatrow

  This includes before and after defining any macro which used `\ttabbox` from floatrow.

⚠ `tabular` inside another environment

- When creating a new environment which contains a `tabular` environment, lwarp's emulation of the tabular does not automatically resume

when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

Cell contents:

⚠ paragraphs

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.

⚠ \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
... &  \multirow{2}{.5in}{text}  & ...
... &  \mrowcell                 & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 208.2 for \multicolumnrow.

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text}  & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use \TabularMacro just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:

⚠ * column specification

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

\multirow

- In \multirow cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the \multirow which do not appear in the HTML version.

⚠ \newcolumntype

- \newcolumntype is ignored; unknown column types are set to l.

Rules:

vertical rules

- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.

full-width rules

- \toprule, \midrule, \bottomrule, and \hline ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use \cmidrule followed by \bottomrule, it may be necessary to use:

```
\cmidrule{2-3} \\[-2ex]
\bottomrule
```

The optional -2ex is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a "misplaced \noalign" error occurs, change

```
This & That \endhead
```

to

```
\warpprintonly{This & That \endhead}
```

and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevent to HTML output.

Other:

- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

longtable headings

- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}.

⚠ S columns

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarp's tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

62.2 Token lookahead

Used by \LWR@futurenonspacelet to look at the next token.

for HTML output:

```
\begin{warpHTML}
```

\LWR@mynexttoken

```
\newcommand\LWR@mynexttoken\relax
```

\futurelet copies the next token then executes a function to analyze

\LWR@futurenonspacelet does the same, but ignores intervening white space

Based on the booktabs style:

\LWR@futurenonspacelet

```
\def\LWR@futurenonspacelet#1{\def\LWR@cs{#1}%
\afterassignment\LWR@fnslone\let\nexttoken= }
\def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
\def\LWR@fnsltwo{%
\expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
\else\let\next=\nexttoken\fi\next}
\def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
\newcommand*{\LWR@getmynexttoken}{%
\LWR@traceinfo{LWR@getmynexttoken}%
% nothing must follow this next line
\LWR@futurenonspacelet\LWR@mynexttoken\LWR@tabledatacolumntag
}
```

62.3 Tabular variables

Bool LWR@startedrow True if should print a row tag before this column.

```
\newbool{LWR@startedrow}
\boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
\newbool{LWR@tabularcelladded}
\boolfalse{LWR@tabularcelladded}
```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```
\newbool{LWR@doinghline}
\boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
\newbool{LWR@doingtbrule}
\boolfalse{LWR@doingtbrule}
```

Bool LWR@doingcmidrule True if the next row will have a cmidrule above it.

This is used by \LWR@tabularfinishrow to force a final empty row to create the border for the \cmidrule.

```
\newbool{LWR@doingcmidrule}
\boolfalse{LWR@doingcmidrule}
```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
\newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```
\newbool{LWR@skippingmrowcell}
```

Bool `LWR@skippingmcolrowcell` True if are doing an empty `\multicolumnrow` cell, and thus there is no data tag to close, and do not print `@` and `!` columns.

```
\newbool{LWR@skippingmcolrowcell}
```

Bool `LWR@skipatbang` True if just finished a `\multicolumn` so should not create the trailing `@` or `!` columns table data cells.

```
\newbool{LWR@skipatbang}
```

Bool `LWR@emptyatbang` True if finishing a row and should print empty `@` or `!` column table data cells.

```
\newbool{LWR@emptyatbang}
```

Bool `LWR@intabularmetadata` True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
\newbool{LWR@intabularmetadata}
\boolfalse{LWR@intabularmetadata}
```

Ctr `LWR@tabularDepth` Tracks whether `&` is being used inside a `tabular`.

```
\newcounter{LWR@tabulardepth}
\setcounter{LWR@tabulardepth}{0}
```

Ctr `LWR@tabularpardepth` Tracks whether should look ahead at the next token when encountering a `\par` while processing tabular contents.

When `LWR@tabularpardepth` is deeper than `LWR@tabulardepth` then lwarp has started looking at the contents of the `tabular`, and thus any `\par`s encountered must be followed by another token lookahead.

```
\newcounter{LWR@tabularpardepth}
\setcounter{LWR@tabularpardepth}{0}
```

```
\newcommand*{\LWR@colsresult}{}%temp storage for column format results
\newcommand*{\LWR@pposition}{}
\newcommand*{\LWR@pleft}{}
\newcommand*{\LWR@pright}{}
```

`\LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalcols`, not counting `@` and `!` columns.

Will contain a string such as `llrrccpc`, exactly one letter per LaTeX table column, without `@`, `!`, `>`, `<`, or the vertical bar.

```
\newcommand*{\LWR@tablecolspec}{}
```

\LWR@strresult Holds the result of `Str` functions.

```
\providecommand*{\LWR@strresult}{}
\providecommand*{\LWR@strresulttwo}{}
```

\LWR@origcolspec Holds the original column specs given to `tabular`.

```
\newcommand*{\LWR@origcolspec}{}
```

Ctr LWR@tablecolspecwidth Holds the number of tokens in the table columns specification.

This is includes one for each `@`, `!`, `<`, `>` column, and also one for each of the parameters of `p`, `@`, `!`, `<`, `>` columns, and three for each `D` column.

(This is not the total # of LaTeX columns in the table.)

```
\newcounter{LWR@tablecolspecwidth}
```

Ctr LWR@tablecolspecindex While parsing the LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
\newcounter{LWR@tablecolspecindex}
```

Ctr LWR@tablecolindex While parsing the LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
\newcounter{LWR@tablecolindex}
```

Ctr LWR@tabletotalcols While parsing a table column specification, begins at 0 and increments by 1 per LaTeX table column. Eventually holds the final number of LaTeX table columns in each row, not counting `@` and `!` columns. (In HTML, `@` and `!` cells become their own columns, but are not included in `LWR@tabletotalcols`.)

```
\newcounter{LWR@tabletotalcols}
```

Ctr LWR@tabletotalcolsnext Holds the next LaTeX table column index while parsing, equal to one more than `LWR@tabletotalcols`.

```
\newcounter{LWR@tabletotalcolsnext}
```

LWR@colatspec A data array of specifications for `@` columns. The leftmost's index is `leftedge`, the others are counter values. See section 35.

LWR@colbangspec A data array of specifications for `!` columns. The leftmost's index is `leftedge`, the

others are counter values. See section 35.

`LWR@colbeforespec` A data array of specifications for > columns.

`LWR@colafterspec` A data array of specifications for < columns.

`LWR@colbarspec` A data array of specifications for vertical rules.

62.4 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:
http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860

`\LWR@instertatbangcols`

```
\newcommand*{\LWR@insertatbangcols}{%
\ifbool{LWR@skipatbang}%
{}%
{%
    \LWR@printatbang{at}{\arabic{LWR@tablecolindex}}%
    \LWR@printatbang{bang}{\arabic{LWR@tablecolindex}}%
}%
}
```

`\LWR@closetabledatacell` If `LWR@skippingmrowcell` or `LWR@skippingmcolrowcell` then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
\newcommand*{\LWR@closetabledatacell}{%
\global\booltrue{LWR@intabularmetadata}%
\ifbool{LWR@exitingtabular}{}%
{% not exiting tabular
    \ifboolexpr{bool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
    {%
```

If not skipping a `\multicolumnrow` cell, insert the `@` and `!` columns after this non-existant column.

```
        \ifbool{LWR@skippingmcolrowcell}%
        {}%
        {\LWR@insertatbangcols}%
    }%
    {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```
        \unskip%
        \ifboolexpr{%
            bool{LWR@tabularmutemods} or
            bool{LWR@skipatbang} or
            bool{LWR@emptyatbang}
        }%
        {}%
        {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tablecolindex}}}%
```

Close paragraphs:

```
        \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
        \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
        \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
        \global\booltrue{LWR@tabularcelladded}%
        \LWR@insertatbangcols%
    }% not skipping mrowcell
}% not exiting tabular
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
\global\boolfalse{LWR@skipatbang}%
}
```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```
\let\LWR@origampmacro&
```

```
\end{warpHTML}
```

62.4.1 Localizing & catcodes

for HTML & PRINT:

```
\begin{warpall}
```

⚠ misplaced alignment tab character &

Place \StartDefiningTabulars and \EndDefiningTabulars before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of & must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
misplaced alignment tab character &
```

\StartDefiningTabulars Place before defining something with & in it.

```
\newcommand{\StartDefiningTabulars}{%
\LWR@traceinfo{StartDefiningTabulars}%
\warpHTMLonly{\catcode`\&=\active}%
}
```

\EndDefiningTabulars Place after defining something with & in it.

```
\newcommand{\EndDefiningTabulars}{%
\LWR@traceinfo{EndDefiningTabulars}%
\warpHTMLonly{\catcode`\&=4}%
}
```

```
\end{warpall}
```

62.4.2 Handling &

for HTML output:

```
\begin{warpHTML}
```

& Will behave depending on whether it is being used inside `tabular`.

& is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

```
\newcommand*{\LWR@tabularampersand}{%
\LWR@traceinfo{LWR@tabularampersand}%
\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
{%
```

If not skipping a multirow cell, close the current data cell.

```
    \unskip%
    \LWR@closetabledatacell%
```

Move to the next column.

```
    \addtocounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
    \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
5366     \LWR@getmynexttoken%
5367 }%
```

If not inside a tabular, performs the original action:

```
5368 {\LWR@origampmacro}%
5369 }
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded. Also, cleveref uses the ampersand in one of its options.

& is made active inside a `tabular`.

& is left alone when in math alignments.

62.4.3 Filling an unfinished row

`\LWR@tabularfinishrow` Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
5370 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5371 \ifboolexpr{%
5372     not bool {LWR@exitingtabular} or%
5373     bool{LWR@doingtbrule} or%
5374     bool{LWR@doingcmidrule} or%
5375     bool{LWR@doinghline} or%
5376     bool{LWR@startedrow}%
5377 }{%
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
5378 \begingroup%
```

If generating a final row for the `\bottomrule` borders, turn off the `@`, `!`, `<`, and `>` column output:

```
5379 \ifbool{LWR@exitingtabular}{%
```

```
    \booltrue{LWR@tabularmutemods}%
}{}%
```

Reenable the table data tags until finished with the final row:

```
\global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
\whileboolexpr{%
    test {
        \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {
            \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}
        }%
    )%
}%
{%
    \LWR@tabledatasinglecolumntag%
```

The following is essentially `\LWR@tabularampersand` with `LWR@emptyatbang` added to empty the following cells:

```
    \LWR@closetabledatacell%
    \addtocounter{LWR@tablecolindex}{1}%
    \boolfalse{LWR@tabularcelladded}%
    \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
    \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
    {\LWR@getmynexttoken}%
    {}%
}%
```

Reenable the original `LWR@exitingtabular` to close the entire table:

```
\endgroup%
\global\boolfalse{LWR@emptyatbang}%
}{}% ifboolexpr
}
```

62.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```
\NewDocumentCommand{\LWR@tabularendofline}{s o}{%
\ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
{\LWR@tabularfinishrow}%
{\LWR@closetabledatacell}%
%    \begin{macrocode}
\LWR@htmltag{/tr}\LWR@orignewline
\global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}
```

Additional setup:

```
\global\boolfalse{LWR@doinghline}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
\LWR@clearmidrules%
```

Start at first column:

```
\setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
\boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide between single column data tag or a special case:

```
\LWR@getmynexttoken%
}
```

62.6 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
\newcommand*{\LWR@colparameter}{}
```

\LWR@parseatcolumn Handles @{text} columns.

```
\newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
\LWR@traceinfo{at column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
    \LWR@traceinfo{at the left edge: %
    \LWR@getexparray{LWR@colatspec}{leftedge}}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
    \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
    \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

\LWR@parsebangcolumn Handles !{text} columns.

```
\newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
\LWR@traceinfo{bang column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
    \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
\newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```
\newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsebarcolumn` Handles vertical rules.

```
\newcommand*{\LWR@parsebarcolumn}{%
\LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
\ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
{% left edge of the table:
    \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
}%
{% not at the left edge:
    \LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcols}}{tvertbarr}%
}%
\booltrue{LWR@validtablecol}%
}
```

62.7 Parsing 'l', 'c', or 'r' columns

`\LWR@parsenormalcolumn` {⟨*thiscolumn*⟩}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
\newcommand*{\LWR@parsenormalcolumn}[1]{%
\appto\LWR@tablecolspec{#1}%
\addtocounter{LWR@tabletotalcols}{1}%
\addtocounter{LWR@tabletotalcolsnext}{1}%
```

```
\LWR@traceinfo{normal column \arabic{LWR@tabletotalcols}: #1}%
\LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcolsnext}}{}%
\booltrue{LWR@validtablecol}%
}
```

62.8 Parsing 'p', 'm', or 'b' columns

`\LWR@parsepcolumn` {⟨*thiscolumn*⟩} The width will be ignored.

```
\newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
\addtocounter{LWR@tablecolspecindex}{1}%
}
```

62.9 Parsing 'D' columns

From the dcolumn package.

`\LWR@parseDcolumn` {⟨*thiscolumn*⟩} The three parameters will be ignored.

```
\newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
\addtocounter{LWR@tablecolspecindex}{3}%
}
```

Table 8: Tabular baseline

l	p	m	b	r
l	par par par	mid mid mid	bot bot bot	r

62.10 Parsing the column specifications

⚠ HTML css cannot exactly match the LaTeX concept of a baseline for a table row. Table 8 shows the LaTeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

```
\newbool{LWR@validtablecol}
```

`\LWR@parsetablecols` {⟨*colspecs*⟩}

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The final number of cells in each row is stored in `LWR@tabletotalcols`.

```
\newcommand*{\LWR@parsetablecols}[1]{%
\LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
\renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
\expandarg%
\StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
\renewcommand*{\LWR@tablecolspec}{}%
```

Table 9: Tabular HTML column conversions

l, r, c: Converted to table cells without paragraph tags.
Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

p: Converted to table cells with paragraph tags. Ref: Table 8, LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate LaTeX baselines.

m: With paragraph tags, CSS `vertical-align:middle`.

b: With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

P, M, B: Horizontally-centered versions.

S: Converted to 'r'. From the siunitx package.

D: Converted to 'c'. From the dcolumn package.

@, !, >, <: One each, in that order.

|: Vertical rule.

Unknown: Converted to 'l'.

`\newcolumn`: Currently treated as unknown.

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
\setcounter{LWR@tabletotalcols}{0}%
\setcounter{LWR@tabletotalcolsnext}{1}%
\LWR@setexparray{LWR@colatspec}{leftedge}{}%
\LWR@setexparray{LWR@colatspec}{1}{}%
\LWR@setexparray{LWR@colatspec}{2}{}%
\LWR@setexparray{LWR@colatspec}{3}{}%
\LWR@setexparray{LWR@colbangspec}{leftedge}{}%
\LWR@setexparray{LWR@colbangspec}{1}{}%
\LWR@setexparray{LWR@colbangspec}{2}{}%
\LWR@setexparray{LWR@colbangspec}{3}{}%
\LWR@setexparray{LWR@colbeforespec}{1}{}%
\LWR@setexparray{LWR@colbeforespec}{2}{}%
\LWR@setexparray{LWR@colbeforespec}{3}{}%
\LWR@setexparray{LWR@colafterspec}{1}{}%
\LWR@setexparray{LWR@colafterspec}{2}{}%
\LWR@setexparray{LWR@colafterspec}{3}{}%
\LWR@setexparray{LWR@colbarspec}{leftedge}{}%
\LWR@setexparray{LWR@colbarspec}{1}{}%
\LWR@setexparray{LWR@colbarspec}{2}{}%
\LWR@setexparray{LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
\setcounter{LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
\LWR@traceinfo{about to StrLen}%
\expandarg%
\StrLen{\LWR@origcolspec}[\LWR@strresult]%
\fullexpandarg%
\LWR@traceinfo{finished StrLen}%
\setcounter{LWR@tablecolspecwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@tablecolspecindex}}{>}{\value{LWR@tablecolspecwidth}}%
    }%
}%
{%
```

Place the next single-character column type into `\LWR@strresult`:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@strresult]%
\LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: \LWR@strresult}%
\fullexpandarg%
```

Not yet found a valid column type

```
\boolfalse{LWR@validtablecol}%
```

Note that the parameter for a `p{spec}` column is a token list which will NOT match l,c,r,p.

```
\IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
\IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
\IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
\IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
\IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the dcolumn package:

```
\IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
\IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions "P", "M", and "B":
`\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}`

```
\IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an `l` column:

```
\ifbool{LWR@validtablecol}{}{%
    \LWR@traceinfo{invalid column type: \LWR@strresult}%
    \LWR@parsenormalcolumn{l}%
}%
\addtocounter{LWR@tablecolspecindex}{1}%
}% whiledo
}%
```

62.11 Starting a new row

\LWR@maybenewtablerow

If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for `hline` or `tbrule` if necessary.

```
\newcommand*{\LWR@maybenewtablerow}
{%
\ifbool{LWR@startedrow}%
{}% started the row
{% not started the row
```

Remember that now have started the row:

```
    \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
    \global\booltrue{LWR@intabularmetadata}%
    \ifbool{LWR@doinghline}%
    {%
        \LWR@htmltag{tr class="hline" }%
        \LWR@orignewline%
    }%
    {% not doing hline
        \ifbool{LWR@doingtbrule}%
        {%
            \LWR@htmltag{tr class="tbrule"}%
            \LWR@orignewline%
        }%
        {\LWR@htmltag{tr}\LWR@orignewline}%
    }% end of not doing hline
}% end of not started the row
}
```

62.12 Printing vertical bar tags

\LWR@printbartag {⟨*index*⟩}

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```
\newcommand*{\LWR@printbartag}[1]{%
\LWR@traceinfo{LWR@printbartag !#1!}%
\ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
{}% muting or empty
{% not muting
    \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{#1}}%
    \ifdefempty{\LWR@tempone}{}{ \LWR@tempone}%
}% not muting
\LWR@traceinfo{LWR@printbartag done}%
}
```

62.13 Printing at or bang tags

\LWR@printatbang {⟨*at -or- bang*⟩} {⟨*index*⟩}

```
\newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```
\edef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}%
\LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```
\ifdefempty{\LWR@atbangspec}%
{}%
{% not empty
    \LWR@htmltag{%
        td class="td#1%
        \LWR@subaddcmidruletrim{}{}%
        \LWR@printbartag{#2}%
        "%
        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \LWR@tdendstyles%
    }%
```

Create an empty cell if muting for the \bottomrule:

```
    \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
    {}%
    {\LWR@atbangspec}%
%
    \LWR@htmltag{/td}\LWR@orignewline%
    \global\booltrue{LWR@tabularcelladded}%
}% not empty
}%
```

\LWR@addleftmostbartag

```
\newcommand*{\LWR@addleftmostbartag}{%
\ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
    \LWR@printbartag{leftedge}%
}{}%
}
```

\LWR@tabularleftedge

```
\newcommand*{\LWR@tabularleftedge}{%
\ifnumcomp{\value{LWR@tablecolindex}}{=}{1}%
{%
    \LWR@printatbang{at}{leftedge}%
    \LWR@printatbang{bang}{leftedge}%
}% left edge
{}% not left edge
}
```

62.14 Data opening tag

\LWR@thiscolspec Temporary storage.

```
\newcommand*{\LWR@thiscolspec}{}
```

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment

```
\newcommand*{\LWR@tabledatasinglecolumntag}%
{%
\LWR@traceinfo{LWR@tabledatasinglecolumntag}%
\LWR@maybenewtablerow%
```

Don't start a new paragraph tag if have already started one:

```
\ifbool{LWR@intabularmetadata}%
{%
```

If have found the end of tabular command, do not create the next data cell:

```
    \ifbool{LWR@exitingtabular}{}%
    {% not exiting tabular
```

Print the `@` and `!` contents before first column:

```
        \LWR@tabularleftedge%
```

Fetch the current column's alignment character into `\LWR@strresult`:

```
        \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
\LWR@traceinfo{LWR@tabledatasinglecolumntag: about to print td tag}%
        \LWR@htmltag{td class="td%
```

append this column's spec:

```
        \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
        \LWR@addcmidruletrim%
        \LWR@addleftmostbartag%
        \LWR@printbartag{\arabic{LWR@tablecolindex}}%
        "%


        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@thiscolspec]%
        \LWR@addformatwpalignment{\LWR@thiscolspec}%
        \LWR@tdendstyles%
        }%
\LWR@traceinfo{LWR@tabledatasinglecolumntag: done printing td tag}%
```

If this is a p, m, b, or X column, allow paragraphs:

```
        \ifboolexpr{%
            test{ \ifdefstring{\LWR@strresult}{p} } or
            test{ \ifdefstring{\LWR@strresult}{m} } or
            test{ \ifdefstring{\LWR@strresult}{b} } or
            test{ \ifdefstring{\LWR@strresult}{P} } or
            test{ \ifdefstring{\LWR@strresult}{M} } or
            test{ \ifdefstring{\LWR@strresult}{B} } or
```

```
            test{ \ifdefstring{\LWR@strresult}{X} }
        }%
        {% allow pars
            \LWR@traceinfo{LWR@tabledatasinglecolumntag: about to LWR@startpars}%
            \global\booltrue{LWR@tableparcell}%
            \LWR@startpars%
            \LWR@traceinfo{LWR@tabledatasinglecolumntag: done with LWR@startpars}%
        }% allow pars
        {}% no pars
```

Print the > contents unless muted for the \bottomrule:

```
        \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
        {}%
        {%
            \LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tablecolindex}}%
        }%
        \global\boolfalse{LWR@intabularmetadata}%
    }% not exiting tabular
}{}% in tabular metadata
\LWR@traceinfo{LWR@tabledatasinglecolumntag: done}%
}%
```

62.15 Midrules

LWR@midrules LWR@midrules is a data array (section 35) of columns each containing a non-zero width if a midrule should be created for this column.

LWR@trimlrules LWR@trimlrules is a data array (section 35) of columns containing l if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 35) of columns containing r if a midrule should be right trimmed for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
\newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
\newlength{\LWR@heavyrulewidth}
\setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
\newlength{\LWR@lightrulewidth}
```

```
\setlength{\LWR@lightrulewidth}{.05em}
```

Len `\LWR@cmidrulewidth` The default width of the rule.

```
\newlength{\LWR@cmidrulewidth}
\setlength{\LWR@cmidrulewidth}{.03em}
```

Len `\LWR@thiscmidrulewidth` The width of the next rule, defaulting to `\LWR@cmidrulewidth`.

If not `\LWR@cmidrulewidth`, a style will be used to generate the custom width.

Assigned from the `LWR@midrules` array.

```
\newlength{\LWR@thiscmidrulewidth}
\setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```

`\LWR@clearmidrules` Start new midrules. Called at beginning of tabular and also at `\\`.

Clears all `LWR@midrules` and `LWR@trimruels` markers for this line.

```
\newcommand*{\LWR@clearmidrules}
{%
\setcounter{LWR@midrulecounter}{1}%
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{\value{LWR@tablecolspecwidth}}%
    }%
}%
{%
\LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
\setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
\LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
\LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
\addtocounter{LWR@midrulecounter}{1}%
}%
}
```

`\LWR@subcmidrule` {⟨*width*⟩} {⟨*trim*⟩} {⟨*leftcolumn*⟩} {⟨*rightcolumn*⟩}

Marks `LWR@midrules` data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

`LWR@doingcmidrule` is set to force an empty row at the end of the tabular to create the rule.

```
\newcommand*{\LWR@subcmidrule}[4]{%
\setcounter{LWR@midrulecounter}{#3}%
\whileboolexpr{%
```

```
    not test {%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
    }%
}%
{%
    \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
    \addtocounter{LWR@midrulecounter}{1}%
}% whiledo
\IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
\IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
\booltrue{LWR@doingcmidrule}%
}
```

\LWR@docmidrule [⟨*width*⟩] {⟨*trim*⟩} {⟨*leftcolumn-rightcolumn*⟩}

Marks LWR@midrules array elements to be "Y" from left to right columns. Also marks trimming for the L and/or R columns.

```
\NewDocumentCommand{\LWR@docmidrule}{O{\LWR@cmidrulewidth} D(){} >{\SplitArgument{1}{-}}m}%
{\LWR@subcmidrule{#1}{#2}#3}
```

Used to compute margins, tabular trims:

```
\newlength{\LWR@templengthone}%
\newlength{\LWR@templengthtwo}%
```

Used to add a style to a table data cell:

```
\newboolean{LWR@tdhavecellstyle}
```

\LWR@tdstartstyles Begins possibly adding a table data cell style.

```
\newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

\LWR@tdaddstyle Starts adding a table data cell style.

```
\newcommand*{\LWR@tdaddstyle}{%
\ifbool{LWR@tdhavecellstyle}%
{; }%
{ style="}%
\booltrue{LWR@tdhavecellstyle}%
}
```

\LWR@tdendstyles Finishes possibly adding a table data cell style. Prints the closing quote.

```
\newcommand*{\LWR@tdendstyles}{%
\ifbool{LWR@tdhavecellstyle}{%
    "%
    \global\boolfalse{LWR@tdhavecellstyle}%
}{}%
}
```

`\LWR@subaddcmidruletrim` {⟨*lefttrim*⟩} {⟨*righttrim*⟩} Adds a `\cmidrule` with optional trim.

```
\newcommand*{\LWR@subaddcmidruletrim}[2]{%
\setlength{\LWR@templengthone}{\LWR@getexparray{LWR@midrules}{\arabic{LWR@tablecolindex}}}%
\ifdimcomp{\LWR@templengthone}{>}{0pt}%
{%
```

Print the class without left and right trim letters appended:

```
    \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
    \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
}%
{%
    \setlength{\LWR@thiscmidrulewidth}{0pt}%
}%
}
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
\newcommand*{\LWR@addcmidruletrim}{%
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@tablecolindex}}}%
}
```

`\LWR@addrulewidth` {⟨*thiswidth*⟩} {⟨*defaultwidth*⟩}

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`:

```
\ifboolexpr{%
    test{\ifdimcomp{#1}{=}{0pt}}%
    or ( test{\ifdimcomp{#1}{=}{#2}} and not bool{FormatWP} )%
}%
{}% default width
{% custom width
```

Ensure that the width is wide enough to display in the browser:

```
    \LWR@forceminwidth{#1}%
```

Begin adding a style:

```
    \LWR@tdaddstyle%
```

The style itself:

```
    \uselengthunit{PT}%
    border-top:\rndprintlength{\LWR@atleastonept} solid %
```

The darkness of the color depends on the thickness of the rule:

```
    \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
    {\#A0A0A0}%
    {% lightrule or heaver
        \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
        {\#808080}%
        {black}%
    }% lightrule or heavier
}% custom width
}
```

`\LWR@addcmidrulewidth` Adds a `style` for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand{\LWR@addcmidrulewidth}{%
\LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
}
```

`\LWR@WPcell` {⟨*text-align*⟩} {⟨*vertical-align*⟩}

```
\newcommand*{\LWR@WPcell}[2]{%
\LWR@tdaddstyle%
text-align:#1; vertical-align:#2%
}
```

\LWR@addformatwpalignment If FormatWP, adds a style for the alignment.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
\newcommand*{\LWR@addformatwpalignment}[1]{%
\ifbool{FormatWP}{%
\IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
\IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
\IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
\IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
\IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
}{}%
}
```

62.16 Multicolumns

62.16.1 Parsing multicolumns

```
\newcounter{LWR@tablemulticolswidth}
```

Indexes into the multicolumn specification:

```
\newcounter{LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
\newbool{LWR@mcolvertbarl}
\newbool{LWR@mcolvertbarr}
```

\LWR@printmccoltype {⟨*colspec*⟩} Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

```
\newcommand*{\LWR@printmccoltype}[1]{%
\LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
\IfStrEq{\LWR@strresult}{l}{l}{}%
\IfStrEq{\LWR@strresult}{c}{c}{}%
\IfStrEq{\LWR@strresult}{r}{r}{}%
\IfStrEq{\LWR@strresult}{p}{p}{}%
\IfStrEq{\LWR@strresult}{m}{m}{}%
\IfStrEq{\LWR@strresult}{b}{b}{}%
\IfStrEq{\LWR@strresult}{P}{P}{}%
\IfStrEq{\LWR@strresult}{M}{M}{}%
\IfStrEq{\LWR@strresult}{B}{B}{}%
\IfStrEq{\LWR@strresult}{S}{r}{}%
\IfStrEq{\LWR@strresult}{X}{p}{}%

\IfStrEq{\LWR@strresult}{|}{%
    \ifnumcomp{\value{LWR@tablemulticolspos}}{=}{1}%
        {\booltrue{LWR@mcolvertbarl}}%
        {\booltrue{LWR@mcolvertbarr}}%
}{}%
\LWR@traceinfo{lwr@printmccoltype done}%
}
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
\newcommand*{\LWR@multicolpartext}{%
\LWR@startpars%
\LWR@multicoltext%
\LWR@stoppars%
}
```

`\LWR@multicolother` {⟨*colspec*⟩} For `@`, `!`, `>`, `<`, print the next token without paragraph tags:

```
\newcommand*{\LWR@multicolother}[1]{%
\addtocounter{LWR@tablemulticolspos}{1}%
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
\LWR@strresult%
```

A valid column data type was found:

```
\booltrue{LWR@validtablecol}%
}
```

`\LWR@multicolskip` Nothing to print for this column type.

```
\newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
\booltrue{LWR@validtablecol}%
}
```

`\LWR@printmccoldata` {⟨*colspec*⟩} Print the data for any valid column type found.

```
\newcommand*{\LWR@printmccoldata}[1]{%
\LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
\boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles `@`, `>`, `<` as it comes to them.

```
\IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{D}{%
\addtocounter{LWR@tablemulticolspos}{3}% skip parameters
\LWR@multicoltext%
}{}%
\IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
\ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```
\LWR@traceinfo{lwr@printmccoldata done}%
}
```

`\parsemulticolumnalignment` {⟨*1: colspec*⟩} {⟨*2: printresults*⟩}

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@`, `>`, `<`, is a token list which will NOT match `l`, `c`, `r`, or `p`.

```
\newcommand*{\LWR@parsemulticolumnalignment}[2]{%
\setcounter{LWR@tablemulticolspos}{1}%
\StrLen{#1}[\LWR@strresult]%
\setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@tablemulticolspos}}{>}{\value{LWR@tablemulticolswidth}}%
    }
}%
{%
```

Execute the assigned print function for each token in the column spec:

```
#2{#1}%
```

Move to the next token in the column spec:

```
\addtocounter{LWR@tablemulticolspos}{1}%
}%
}
```

62.16.2 Multicolumn factored code

```
\newcommand{\LWR@multicoltext}{}
```

To find multicolumn right trim:

```
\newcounter{LWR@lastmulticolumn}
```

`\LWR@domulticolumn` [⟨*1: vpos*⟩] [⟨*2: #rows*⟩] {⟨*3: numLaTeXcols*⟩} {⟨*4: numHTMLcols*⟩} {⟨*5: colspec*⟩} {⟨*6: text*⟩}

```
\NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%
```

```
\LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
\renewcommand{\LWR@multicoltext}{%
#6%
\booltrue{LWR@validtablecol}%
}%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
\setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
\addtocounter{LWR@lastmulticolumn}{#3}%
\addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
\LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
\LWR@htmltag{td colspan="#4" %

\IfValueT{#2}{ % rows?
rowspan="#2" %

\IfValueT{#1}{% vpos?
\ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
\ifstrequal{#1}{t}{style="vertical-align:top" }{}%
}% vpos?
}% rows?

 class="td%
```

Print the column type and vertical bars:

```
\boolfalse{LWR@mcolvertbarl}%
\boolfalse{LWR@mcolvertbarr}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule” for a horizontal rule:

```
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%
```

Also add vertical bar class.

```
\ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
\ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%
```

Close the class tag's opening quote:

```
"%
```

```
\LWR@tdstartstyles%
\LWR@addcmidrulewidth%
\LWR@addformatwpalignment{#5}%
\LWR@tdendstyles%
}% end of the opening table data tag
\global\boolfalse{LWR@intabularmetadata}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
}
```

62.16.3 Multicolumn

`\LWR@htmlmulticolumn` {⟨*numcols*⟩} {⟨*alignment*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
{%
```

Figure out how many extra HTML columns to add for `@` and `!` columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tablecolindex}{#1}%
\addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing `@` or `!` columns for this cell:

```
\booltrue{LWR@skipatbang}%
}
```

62.16.4 Longtable captions

longtable captions use `\multicolumn`.

Bool `LWR@starredlongtable` Per the caption pacakge, step the counter if `longtable*`.

```
\newbool{LWR@starredlongtable}
\boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
\providecommand*{\LTcaptype}{table}
```

`\LWR@longtabledatacaptiontag` * [⟨*toc entry*⟩] {⟨*caption*⟩}

```
\NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
{%
```

Remember the latest name for `\nameref`:

```
\IfValueTF{#2}{% optional given?
    \ifblank{#2}% optional empty?
    {\LWR@setlatestname{#3}}% empty
    {\LWR@setlatestname{#2}}% given and non-empty
}% optional given
{\LWR@setlatestname{#3}}% no optional
```

Create a multicolumn across all the columns:

Figure out how many extra HTML columns to add for `@` and `!` columns found between the first and the last column:

```
\LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
{% \LWR@domulticolumn
\IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
{% yes star
    \LWR@figcaption%
    #3%
    \endLWR@figcaption%
```

```
}%
{% No star:
```

Not the star version:

Don't step the counter if `\caption[]{A caption.}`

```
    \ifbool{LWR@starredlongtable}%
    {%
        \ifblank{#2}% TOC entry
        {}%
        {%
            \refstepcounter{\LTcaptype}%
            \protected@edef\@currentlabel{%
            \csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
        }%
    }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
    \LWR@figcaption%
    \csuse{fnum@\LTcaptype}\CaptionSeparator#3%
    \endLWR@figcaption%
```

See if an optional caption was given:

```
    \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
    {}%
```

If the optional caption was given, but might only be `[]`:

```
    {% TOC entry not empty
        \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is `[]`:

```
        {% No TOC entry
            \addcontentsline%
            {\csuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
            {\ignorespaces #3\protect\relax}%
            }%
        }% end of No TOC entry
```

The optional caption has text enclosed:

```
        {% yes TOC entry
            \addcontentsline%
            {\csuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
            {\ignorespaces #2\protect\relax}%
            }%
        }% end of yes TOC entry
    }% end of TOC entry not empty
}% end of no star
```

Skip any trailing `@` or `!` columns for this cell:

```
\booltrue{LWR@skipatbang}%
}% end of \LWR@domulticolumn

\addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
\addtocounter{LWR@tablecolindex}{-1}

}
```

62.16.5 Counting HTML tabular columns

The LaTeX specification for a table includes a number of columns separated by the `&` character. These columns differ in content from line to line. Additional virtual columns may be specified by the special `@` and `!` columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, `@` and `!` columns are placed into their own tabular columns. Thus, a LaTeX `\multicolumn` command may span several additional `@` and `!` columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```
\newcounter{LWR@tabhtmlcolindex}
\newcounter{LWR@tabhtmlcolend}
\newcounter{LWR@tabhtmlcoltotal}
```

`\LWR@subtabularhtmlcolumns` `{`⟨*index*⟩`}`

Factored from `\LWr@tabularhtmlcolumns`, which follows.

```
\newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the @ specification for this column:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the @ specification is not empty, add to the count:

```
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
}
```

`\LWR@tabularhtmlcolumns` {⟨*starting LaTeX column*⟩} {⟨*number LaTeX columns*⟩}

Compute the total number of HTML columns being spanned, considering the starting LaTeX table column and the number of LaTeX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
\newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the LaTeX span, to which additional @ and ! columns may be added:

```
\setcounter{LWR@tabhtmlcolindex}{#1}%
\setcounter{LWR@tabhtmlcoltotal}{#2}%
\setcounter{LWR@tabhtmlcolend}{#1}%
\addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
\ifnumcomp{\value{LWR@tabhtmlcolindex}}{=}{1}{%
    \LWR@subtabularhtmlcolumns{leftedge}%
}{}%
```

Walk across the LaTeX columns looking for @ and ! columns:

```
\whileboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}%
    }%
```

```
}%
{%
    \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
    \addtocounter{LWR@tabhtmlcolindex}{1}%
}% whiledo
}

\end{warpHTML}
```

62.17 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 208 for the HTML versions.

for HTML & PRINT:

```
\begin{warpall}
```

`\multicolumnrow` {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex

After the user may have

```
\AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

```
\@ifundefined{@xmultirow}
{}% no version of multirow was loaded
{% \@xmultirow defined, so some version of multirow was loaded
```

`\@ifpackageloaded{multirow}` determines if v2.0 or later of multirow was used, which included the `\ProvidesPackage` macro.

```
\@ifpackageloaded{multirow}{% v2.0 or newer
\@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
{% v2.0+:
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
}
```

```
{% loaded but older, probably not executed:
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}
}% packageloaded{multirow}
```

If not `\@ifpackageloaded{multirow}` but `\@xmultirow` is defined, then this must be v1.6 or earlier, which did not `\ProvidesPackage{multirow}`, and did not have the `vposn` option.

```
{% v1.6 or older did not \ProvidePackage
\DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
{\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}
```

The user-level interface. This is provided if the HTML version was not already given.

```
\providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
}% \@xmultirow defined, so multirow was loaded
}% AtBeginDocument

\end{warpall}
```

62.18 Utility macros inside a table

for HTML output:

```
\begin{warpHTML}
```

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
\newcommand*{\LWR@donothing}{}
```

In case bigdelim is not loaded:

```
\newcommand*{\ldelim}{}
\newcommand*{\rdelim}{}

\end{warpHTML}
```

62.19 Special-case tabular markers

for HTML & PRINT:

```
\begin{warpall}
```

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatcally start a new HTML table data cell yet. See section 8.8.

```
\newcommand*{\TabularMacro}{}
\end{warpall}
```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

⚠ `tabular` inside another environment

When creating a new environment which contains a `tabular` environment, lwarp's emulation of the tabular does not automatically resume when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

for HTML output:

```
\begin{warpHTML}
\newcommand*{\ResumeTabular}{%
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
\LWR@getmynexttoken%
}
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand*{\ResumeTabular}{}
\end{warpprint}
```

62.20 Checking for a new table cell

for HTML output:

```
6096 \begin{warpHTML}
```

Bool `LWR@exitingtabular`

When `\end` is found, turns off the next opening data tag.

```
6097 \newbool{LWR@exitingtabular}
```

Bool `LWR@tabularmutemods`

Mutes HTML output for `@`, `!`, < and >.

This is used while printing the final row to generate `\bottomrules`.

```
6098 \newbool{LWR@tabularmutemods}
```

`\LWR@tabledatacolumntag`

Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
6099 \newcommand*{\LWR@tabledatacolumntag}%
6100 {%
6101 \LWR@traceinfo{LWR@tabledatacolumntag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
6102 \let\mynext\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```
6103 \ifdefequal{\LWR@mynexttoken}{\end}%
6104     {\global\booltrue{LWR@exitingtabular}}{}%
```

`longtable` can have a caption in a cell

```
6105 \ifdefequal{\LWR@mynexttoken}{\caption}%
6106     {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
6107 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
6108     {\let\mynext\LWR@donothing}{}%
6109 \ifdefequal{\LWR@mynexttoken}{\multirow}%
6110     {\let\mynext\LWR@donothing}{}%
6111 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
6112     {\let\mynext\LWR@donothing}{}%
```

If an \mrowcell, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mrowcell}%
    {\let\mynext\LWR@donothing}{}%
```

If an \mcolrowcell, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\hline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\firsthline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\lasthline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\toprule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\midrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\cmidrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\specialrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\cline}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\bottomrule}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
    {\let\mynext\LWR@donothing}{}%
%
\ifdefequal{\LWR@mynexttoken}{\ldelim}%
    {\let\mynext\LWR@donothing}{}%
%
```

```
\ifdefequal{\LWR@mynexttoken}{\rdelim}%
    {\let\mynext\LWR@donothing}{}%
```

Ignore an empty line between rows:

```
\ifdefequal{\LWR@mynexttoken}{\par}%
    {\let\mynext\LWR@donothing}{}%
```

no action for an \end token

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:

```
\LWR@traceinfo{LWR@tabledatacolumntag: about to do mynext}%
\mynext%
\LWR@traceinfo{LWR@tabledatacolumntag: done}%
}
```

```
\end{warpHTML}
```

62.21 \mrowcell

for HTML & PRINT:

```
\begin{warpall}
```

\mrowcell

The user must insert \mrowcell into any \multirow cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mrowcell}{}
```

```
\end{warpall}
```

62.22 \mcolrowcell

for HTML & PRINT:

```
\begin{warpall}
```

\mcolrowcell

The user must insert \mcolrowcell into any \multicolumnrow cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mcolrowcell}{}
```

```
\end{warpall}
```

62.23 New \tabular definition

for HTML output:
```
\begin{warpHTML}
```

These are default defininitions in case booktabs is not loaded, and are not expected to used, but must exist as placeholders.

```
\newcommand*{\LWR@origtoprule}[1]{\hline}
\newcommand*{\LWR@origmidrule}[1]{\hline}
\LetLtxMacro\LWR@origcmidrule\cline
\newcommand*{\LWR@origbottomrule}[1]{\hline}
\newcommand*{\LWR@origaddlinespace}[1][]{}
\newcommand*{\LWR@origmorecmidrules}{}
\newcommand*{\LWR@origspecialrule}[3]{\hline}
```

`\LWR@hline` The definition of `\hline` depends on whether tabls has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
\AtBeginDocument{
\@ifpackageloaded{lwarp-tabls}
{
\newcommand*{\LWR@hline}[1][]{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
    {\booltrue{LWR@doinghline}}%
    \LWR@getmynexttoken}%
}
{
\newcommand*{\LWR@hline}{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
    {\booltrue{LWR@doinghline}}%
    \LWR@getmynexttoken}%
}
}% AtBeginDocument
```

`\LWR@nullifyNoAutoSpacing` For babel-french, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses xstring.

```
\AtBeginDocument{
\@ifundefined{frenchbsetup}%
{% no babel-french
    \newcommand*{\LWR@nullifyNoAutoSpacing}{}
}% no babel-french
{% yes babel-french
    \newcommand*{\LWR@nullifyNoAutoSpacing}{%
```

```
        \NoAutoSpacing%
        \renewcommand*{\NoAutoSpacing}{}%
        \renewcommand*{\LWR@FBcancel}{}%
    }
}% yes babel-french
}% AtBeginDocument
```

Env `LWR@tabular` [⟨*vertposition*⟩] {⟨*colspecs*⟩}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since siunitx might redefine tabular in the user's document.

```
\StartDefiningTabulars

\newenvironment*{LWR@tabular}[2][]
{%
\LWR@traceinfo{LWR@tabular started}%
\addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
\global\boolfalse{LWR@doinghline}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
```

For babel-french, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```
\LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the `@` and `!` columns.

```
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
```

Create the `table` tag:

```
\global\booltrue{LWR@intabularmetadata}%
\LWR@forcenewpage
\LWR@htmlblocktag{table}%
```

Parse the table columns:

```
\LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
\LWR@stoppars%
```

Track column #:

```
\setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
\boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
\LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
\LetLtxMacro{\\}{\LWR@tabularendofline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynextoken` to see if the next token might create a new data cell:

The optional parameter for `\hline` supports the tabls package.

```
\LWR@traceinfo{LWR@tabular: redefining macros}%
\renewcommand*{\hline}{\LWR@hline}%
%
\RenewDocumentCommand{\cline}{m}%
{\LWR@docmidrule{##1}\LWR@getmynexttoken}%


\DeclareDocumentCommand{\toprule}{o d()}
    {%
        \IfValueTF{##1}%
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doingtbrule}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\midrule}{o d()}%
```

```
    {%
        \IfValueTF{##1}%
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doinghline}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\cmidrule}{O{\LWR@cmidrulewidth} d() m}%
{\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\bottomrule}{o d()}
    {%
        \IfValueTF{##1}%
            {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
                {\booltrue{LWR@doingtbrule}}%
            }%
    \LWR@getmynexttoken}%
%
\DeclareDocumentCommand{\addlinespace}{o}{}%
\DeclareDocumentCommand{\morecmidrules}{}{}%
\DeclareDocumentCommand{\specialrule}{m m m d()}%
    {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}\LWR@getmynexttoken}%
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use `\LWR@getmynexttoken`.

```
\renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
\renewcommand*{\mrowcell}{%
    \LWR@maybenewtablerow%
    \LWR@tabularleftedge%
    \global\booltrue{LWR@skippingmrowcell}%
}%
\renewcommand*{\mcolrowcell}{%
    \LWR@maybenewtablerow%
    \global\booltrue{LWR@skippingmcolrowcell}%
}%
\LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%
```

Reset for new processing:

```
\global\boolfalse{LWR@tableparcell}%
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
```

```
\global\boolfalse{LWR@skipatbang}%
\global\boolfalse{LWR@emptyatbang}%
```

Set & for its special meaning inside the tabular:

```
\StartDefiningTabulars%
\protected\gdef&{\LWR@tabularampersand}%
```

Nest one level deeper of tabular paragraph handling:

```
\addtocounter{LWR@tabularpardepth}{1}%
```

Look ahead for a possible table data cell:

```
\LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
\LWR@getmynexttoken%
}%
```

Ending the environment:

```
{%
\LWR@traceinfo{LWR@tabular ending}%
```

Unnest one level of tabular paragraph handling:

```
\addtocounter{LWR@tabularpardepth}{-1}%
\ifboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {%
            \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
        }%
    )%
}%
{%
    \LWR@tabularfinishrow%
}%
{%
    \LWR@closetabledatacell%
}%
\LWR@htmlblocktag{/tr}%
\LWR@htmlblocktag{/table}%
\global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
\addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
\protected\gdef&{\LWR@origampmacro}%
\EndDefiningTabulars%
\LWR@traceinfo{LWR@tabular finished ending}%
}

\EndDefiningTabulars
```

```
\end{warpHTML}
```

63 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

for HTML output:

```
\begin{warpHTML}
```

63.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
\providecommand*{\@currentlabelname}{}
```

`\LWR@stripperiod` {⟨*text*⟩} [⟨.⟩]

Removes a trailing period.

```
\def\LWR@stripperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` {⟨*object name*⟩}

Table 10: Cross-referencing data structures

Original LaTeX: (print and HTML)

`\refstepcounter`**:** Steps the couunter and sets `\@currentlabel`.

`\@currentlabel`**:** `\p@<ctr>\the<ctr>` Updated by `\refstepcounter`.

`\label`**:** Writes to the `.aux` file:
`\newlabel{<label>}{{\@currentlabel}{\thepage}}`

`\newlabel`**:** When the `.aux` file is read, sets `\r@<label>`.

`\r@<label>`**:** Set to: `{{\@currentlabel}{\thepage}}`

`\ref`**:** Returns the first part of `\r@<label>`.

`\pageref`**:** Returns the second part of `\r@<label>`.

Added by lwarp: (HTML only)

`\label`**:** Adds HTML tags (section 63.3), plus `\splabel` data (section 63.2):

zLWR@name: The section name for this label.
zLWR@htmlfilenumer: The filenumber or name for this label.
zLWR@lateximagedepth: The `lateximagedepth` for this label.
zLWR@lateximagenumber: The `lateximagenumber` for this label.

`\nameref`**:** Emualted from hyperref for lwarp. See section 63.4.

`\ref` **and** `\nameref`**:** Adds HTML tags. See section 63.4.

Added by amsmath: (print and HTML)

`\label`**:** Execution is delayed until the math environment is completed.

`\ltx@label`**:** LaTeX `\label`, (HTML: patched by lwarp,) later patched by cleveref.

Added by cleveref: (print and HTML)

`\refstepcounter:` Added: sets `\cref@currentlabel`.

`\cref@currentlabel`**:** (`<type>=<ctr>` unless an alias is used):
`[<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>}` Also see section 49.4 for use with footnotes.

`\label`**:** Writes to the `.aux` file:
`\newlabel{<label>@cref}{{\cref@currentlabel}{\thepage}}`

`\newlabel`**:** (Unchanged.) When the `.aux` file is read, sets `\r@<label>@cref`.

`\r@<label>@cref`**:** Set to: `{{\cref@currentlabel}{\thepage}}`

Utility functions: See `\cref@getlabel`, `\cref@gettype`, `\cref@getcounter`, `\cref@getprefix`.

Cross-referencing names: `\crefname` and `\Crefname` assign human-readable names for references to this counter type.

Additionally patched by lwarp: (HTML only)

`\cref`**, etc.:** Modified for lwarp. See section 76.

`\label` **inside math:** See section 69.5.1.

Footnotes: See `\noteentry` in section 49.4.

Removes \label, strips any final period, and remembers the result.

```
\newcommand*{\LWR@setlatestname}[1]{%
```

Remove \label and other commands from the name, the strip any final period. See zref-titleref and gettitlestring.

```
\GetTitleStringExpand{#1}%
\edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
\edef\@currentlabelname{%
\expandafter\LWR@stripperiod\@currentlabelname%
\ltx@empty.\ltx@empty\@nil%
}%
}
```

63.2 Zref setup

See:
http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference

Create a new property list called special:

```
\zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
\zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
\zref@newprop{zLWR@htmlfilenumber}{%
\ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
}%
```

Additional properties for lateximages:

```
\zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
\zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

zLWR@htmlfilenumber property holds the file number or name

Add a LWR@htmlfilenumber property, and lateximage properties to special:

```
\zref@addprop{special}{zLWR@name}
```

```
\zref@addprop{special}{zLWR@htmlfilenumber}
\zref@addprop{special}{zLWR@lateximagedepth}
\zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
\newcommand*{\LWR@spref}[2]{%
\zref@extractdefault{#1}{#2}{??}%
}
```

`\LWR@nameref` {⟨*label*⟩} Returns the section name for this label:

```
\newcommand*{\LWR@nameref}[1]{%
\LWR@spref{#1}{zLWR@name}%
}
```

`\LWR@htmlfileref` {⟨*label*⟩} Returns the file number for this label:

```
\newcommand*{\LWR@htmlfileref}[1]{%
% DO NOT USE \LWR@traceinfo HERE!  Will be expanded.
\LWR@spref{#1}{zLWR@htmlfilenumber}%
}
```

`\LWR@lateximagedepthref` {⟨*label*⟩} Returns the `lateximagedepth` for this label:

```
\newcommand*{\LWR@lateximagedepthref}[1]{%
\LWR@spref{#1}{zLWR@lateximagedepth}%
}
```

`\LWR@lateximagenumberref` {⟨*label*⟩} Returns the `lateximagenumber` for this label:

```
\newcommand*{\LWR@lateximagenumberref}[1]{%
\LWR@spref{#1}{zLWR@lateximagenumber}%
}
```

`\LWR@splabel` {⟨*label*⟩} Sanitize the name and then creates the label:

```
\newcommand*{\LWR@splabel}[1]{%
\LWR@traceinfo{LWR@splabel !#1!}%
\LWR@setlatestname{\@currentlabelname}%
\zref@labelbylist{#1}{special}%
}
```

63.3 Labels

`\LWR@sublabel` {⟨*label*⟩} Creates an HTML id tag.

`\detokenize` is used to allow underscores in the labels.

```
\newcommand*{\LWR@sublabel}[1]{%
\LWR@traceinfo{LWR@sublabel !#1!}%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored…)

```
    \LWR@sanitize{#1}%
    \ifbool{LWR@doingstartpars}%
    {% pars allowed
        \ifbool{LWR@doingapar}%
        {% par started
            \LWR@htmltag{a id="\LWR@sanitized"}\LWR@htmltag{/a}%
        }% par started
        {% par not started
            \LWR@stoppars%
            \LWR@htmltag{a id="\LWR@sanitized"}\LWR@htmltag{/a}%
            \LWR@startpars%
        }% par not started
    }% pars allowed
    {% pars not allowed
        \LWR@htmltag{a id="\LWR@sanitized"}\LWR@htmltag{/a}%
    }% pars not allowed
}% not lateximage
}
```

`\LWR@newlabel` (⟨*bookmark*⟩) {⟨*label*⟩} [⟨*type*⟩]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular LaTeX `\label`.

clevereref later encases this to add its own cross-referencing.

The optional ⟨*bookmark*⟩ is per the memoir class, and is ignored.

The optional ⟨*type*⟩ is per the ntheorem package, and is ignored.

```
\NewDocumentCommand{\LWR@newlabel}{d() m o}{%
\LWR@traceinfo{LWR@newlabel: starting}%
\LWR@traceinfo{LWR@newlabel: !#2!}%
% \@bsphack%
```

Create a traditional LaTeX label, as modified by cleveref:

```
\LWR@origlabel{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
\LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
\LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
\LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
\LWR@splabel{#2}%
\LWR@sublabel{#2}%
% \@esphack%
\LWR@traceinfo{LWR@newlabel: done}%
}
```

63.4 References

`\LWR@startref` `{`⟨*label*⟩`}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
\newcommand*{\LWR@startref}[1]
{%
\edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
\LWR@sanitize{#1}%
\LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
\LWR@htmltag{a href="%
\LWR@traceinfo{LWR@startref B}%
\LWR@htmlrefsectionfilename{#1}%
\LWR@traceinfo{LWR@startref C}%
\#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
\LWR@traceinfo{LWR@startref D: !#1!}%
\ifthenelse{\equal{\LWR@lidref}{??}}%
```

"??" if LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```
{%
    \LWR@traceinfo{LWR@startref D0: ??}%
    ??%
}%
```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
{%
    \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
    \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
    {%
        \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
        lateximage\LWR@lateximagenumberref{#1}%
    }%
    {%
        \LWR@traceinfo{LWR@startref D3}%
```

\detokenize is used to allow underscores in the labels:

```
        \LWR@sanitized%
    }%
}%
\LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
"}%
\LWR@traceinfo{LWR@startref F}%
}
```

\LWR@subnewref {⟨*label*⟩} {⟨*label or sub@label*⟩}

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as "1(b)".

```
\NewDocumentCommand{\LWR@subnewref}{m m}{%
\LWR@traceinfo{LWR@subnewref #1 #2}%
\LWR@startref{#1}%
\LWR@origref{#2}%
\LWR@htmltag{/a}%
}
```

`\ref` * {⟨*label*⟩} `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * {⟨*label*⟩} Create an internal document reference link, or without a link if starred per hyperref.

```
\NewDocumentCommand{\LWR@newref}{s m}{%
\LWR@traceinfo{LWR@newref !#2!}%
\IfBooleanTF{#1}%
{\LWR@origref{#2}}%
{\LWR@subnewref{#2}{#2}}%
}
```

`\pagerefPageFor` Text for page references.

```
\newcommand*{\pagerefPageFor}{see }
```

`\pageref` * {⟨*label*⟩} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
\NewDocumentCommand{\LWR@newpageref}{s m}{%
\IfBooleanTF{#1}%
{(\pagerefPageFor\LWR@origref{#2})}%
{(\cpageref{#2})}%
}
```

`\nameref` {⟨*label*⟩}

```
\DeclareRobustCommand*{\nameref}[1]{%
\LWR@traceinfo{nameref A}%
\LWR@startref{#1}%
\LWR@traceinfo{nameref B}%
\LWR@nameref{#1}%
\LWR@traceinfo{nameref C}%
\LWR@htmltag{/a}%
\LWR@traceinfo{nameref D}%
}
```

`\Nameref` {⟨*label*⟩} In print, adds the page number. In HTML, does not.

```
\LetLtxMacro\Nameref\nameref
```

63.5 Hyper-references

⚠ Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`

⚠ Do not tell other packages that hyperref is emulated. Some packages patch various commands if hyperref is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.

⚠ Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
% DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
% \EmulatesPackage{hyperref}[2015/08/01]% Disabled.  Do not do this.
```

Emulates hyperref:

`\@currentHref` Added to support backref.

```
\AtBeginDocument{
\def\@currentHref{%
autopage-\theLWR@currentautosec%
}
}
```

Create a link with a text name:

`\LWR@subhyperref` {⟨*URL*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@subhyperref}{m +m}{%
\LWR@traceinfo{LWR@subhyperref !#1!}%
\LWR@sanitize{#1}%
\LWR@htmltag{%
    a href="\LWR@sanitized" %
%     a href="%
%         \begingroup\@sanitize#1\endgroup%
%     " %
    target="\_{}blank"\LWR@orignewline%
}%
#2%
\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\LWR@subhyperrefclass {⟨*URL*⟩} {⟨*text*⟩} {⟨*htmlclass*⟩}

```
\NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
% \LWR@sanitize{#1}%
\LWR@htmltag{%
%     a href="\LWR@sanitized"
    a href="%
        \begingroup\@sanitize#1\endgroup%
    " %
    class="#3"\LWR@orignewline%
}%
#2%
\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\href [⟨*options*⟩] {⟨*URL*⟩} {⟨*text*⟩}

Create a link with accompanying text:

```
\NewDocumentCommand{\href}{O{} m +m}{%
\LWR@ensuredoingapar%
\LWR@subhyperref{#2}{#3}%
}
```

\nolinkurl {⟨*URL*⟩}

Print the name of the link without creating the link:

```
\newcommand*{\nolinkurl}[1]{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\LWR@templink%
}
```

\url {⟨*URL*⟩}

Create a link whose text name is the address of the link. The url package may redefine `\url`, so it is `\let` to `\LWR@urlahere` and also redefined by lwarp-url.

```
\NewDocumentCommand{\url}{m}{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\href{\LWR@templink}{\LWR@templink}%
}
```

\LWR@subinlineimage [⟨*alttag*⟩] {⟨*class*⟩} {⟨*filename*⟩} {⟨*extension*⟩} {⟨*style*⟩}

```
\newcommand*{\LWR@subinlineimage}[5][]{%
\ifblank{#1}%
{\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}%
{\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"}}%
}
```

```
\end{warpHTML}
```

Table 11: Float data structures

For each <type> of float (figure, table, etc.) there exists the following:

counter <type>: A counter called <type>, such as figure, table.

\<type>name: Name. \figurename prints "Figure", etc.

\ext@<type>: File extension. \ext@figure prints "lof", etc.

\fps@<type>: Placement.

\the<type>: Number. \thetable prints the number of the table, etc.

\p@<type>: Parent's number. Prints the number of the [within] figure, etc.

\fnum@<type>: Prints the figure number for the caption.
\<type>name \the<type>, "Figure 123".

\<type>: Starts the float environment. \figure or \begin{figure}

\end<type>: Ends the float environment. \endfigure or \end{figure}

\tf@<ext>: The LaTeX file identifier for the output file.

LWR@have<type>: A boolean rememebering whether a \listof was requested for a float of this type.

File with extension lo<f,t,a-z>: An output file containing the commands to build the \listof<type><name> "table-of-contents" structure.

Cross-referencing names: For cleveref's \cref and related, \crefname and \Crefname assign human-readable names for references to this float type.

64 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each <type> of float.

\@makecaption is redefined to print the float number and caption text, separated by \CaptionSeparator, which works with the babel package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: "Figure 123 – Caption text".

64.1 Float captions

for HTML output:
```
\begin{warpHTML}
```

`\LWR@floatbegin` {⟨*type*⟩} [⟨*placement*⟩]

Begins a `\newfloat` environment.

```
\NewDocumentCommand{\LWR@floatbegin}{m o}{%
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
\addtocounter{LWR@thisautoid}{1}%
\booltrue{LWR@freezethisautoid}%
```

```
\begingroup%
```

Settings while inside the environment:

```
\LWR@origraggedright%
```

Open an HTML `figure` tag:

```
\LWR@htmltag{figure id="autoid-\arabic{LWR@thisautoid}" class="#1"}%
\ifbool{FormatWP}{%
    \LWR@orignewline%
    \LWR@BlockClassWP{}{}{wp#1}%
}{}%
```

```
\renewcommand*{\@captype}{#1}%
\caption@settype{#1}%
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== begin #1 ===

}{}%
}
```

`\@float` `\@dlbfloat` Support packages which create floats directly.

```
\let\@float\LWR@floatbegin
\let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
\newcommand*{\LWR@floatend}{%
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== end ===

}{}%
\LWR@stoppars%
```

Close an HTML figure tag:

```
\ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
\LWR@htmlelementend{figure}%
\endgroup%
\boolfalse{LWR@freezethisautoid}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
}
```

\end@float
\end@dlbfloat
Support packages which create floats directly.

```
\let\end@float\LWR@floatend
\let\end@dblfloat\LWR@floatend
```

Ctr LWR@thisautoid A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```
\newcounter{LWR@thisautoid}
```

Ctr LWR@thisautoidWP A sequential counter for all word processor conversion s. This is used to convince LibreOffice to form a frame around this element.

```
\newcounter{LWR@thisautoidWP}
```

Bool LWR@freezethisautoid Prevents multiple increments of \LWR@thisautoid inside a float.

```
\newbool{LWR@freezethisautoid}
\boolfalse{LWR@freezethisautoid}
```

\LWR@newautoidanchor Adds a new <autoid> anchor.

```
\newcommand*{\LWR@newautoidanchor}{%
\ifbool{LWR@freezethisautoid}{}{%
    \addtocounter{LWR@thisautoid}{1}%
    \LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"}\LWR@htmltag{/a}%
```

```
}%
}
```

`\@captype` Remembers which float type is in use.

```
\newcommand*{\@captype}{}
```

64.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
\AtBeginDocument{\providecommand*{\CaptionSeparator}{:~}}
```

`\@makecaption` {⟨*name and num*⟩} {⟨*text*⟩}

Prints the float type and number, the caption separator, and the caption text.

```
\AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

64.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the LaTeX PDF file, the LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special LaTeX labels). This page number is used to generate an `autopage` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autoid` `<id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr `LWR@nextautoid` Tracks `autoid` for floats. Tracks `autopage` for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
\newcounter{LWR@nextautoid}
\newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` {⟨*autopage*⟩} {⟨*autoid*⟩}

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autoid` are remembered for `\l@figure` to use when creating the HTML links.

```
\newcommand*{\LWRsetnextfloat}[2]{%
```

```
\setcounter{LWR@nextautopage}{#1}%
\setcounter{LWR@nextautoid}{#2}%
}
```

Ctr `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autoid` by the modified `\addcontentsline` just before each float's entry.

```
\newcounter{LWR@latestautopage}
\setcounter{LWR@latestautopage}{1}
```

Env `LWR@figcaption` Encapsulates a caption inside `<figcaption>`, and if `FormatWP` then also a `<div>` with an italic style.

```
\newenvironment*{LWR@figcaption}
{%
\LWR@htmlblocktag{figcaption}
\ifbool{FormatWP}{%
\begin{BlockClass}[font-style:italic]{italic}
\LWR@origvspace*{\baselineskip}
}{}%
}
{%
\ifbool{FormatWP}{\end{BlockClass}}{}%
\LWR@htmlblocktag{/figcaption}%
}
```

```
\let\LWR@origcaption@begin\caption@begin
\let\LWR@origcaption@end\caption@end
\let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
\newcommand{\LWR@caption@begin}
{
\LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
\begingroup%
```

The caption code was not allowing the closing `par` tag:

```
\renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
\RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
\RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
\LWR@figcaption%
\LWR@origcaption@begin%
}
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
\newcommand{\LWR@caption@end}
{%
\LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
\LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
\endLWR@figcaption%
\endgroup%
% \leavevmode% avoid bad space factor (0) error
\LWR@traceinfo{LWR@caption@end: done}%
}
```

`\caption@begin` `\caption@end` Low-level patches to create HTML tags for captions.

```
\AtBeginDocument{
\let\caption@begin\LWR@caption@begin
\let\caption@end\LWR@caption@end
}
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
\let\LWR@origcaptionlistentry\captionlistentry

\renewcommand*{\captionlistentry}{%
\LWR@ensuredoingapar%
\LWR@origcaptionlistentry%
}

\def\LWR@LTcaptionlistentry{%
\LWR@ensuredoingapar%
```

```
\LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"}\LWR@htmltag{/a}%
  \bgroup
  \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
          {\egroup\LWR@LT@captionlistentry}}%
\def\LWR@LT@captionlistentry#1{%
  \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%
```

\addcontentsline Patched to write the autopage and autoid before each float's entry. No changes if writing .toc For a theorem, automatically defines \ext@<type> as needed, to mimic and reuse the float mechanism.

```
\let\LWR@origaddcontentsline\addcontentsline

\renewcommand*{\addcontentsline}[3]{%
\ifstrequal{#1}{toc}{}{% not TOC
    \LWR@newautoidanchor%
    \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}%
    \addtocontents{\@nameuse{ext@#2}}{%
        \protect\LWRsetnextfloat%
        {\arabic{LWR@latestautopage}}%
        {\arabic{LWR@thisautoid}}%
    }%
}% not TOC
\LWR@origaddcontentsline{#1}{#2}{#3}%
}
```

Pkg capt-of Pkg caption Either package provides \captionof, which is later patched at the beginning of the document.

\captionof Patched to handle paragraph tags.

```
\AtBeginDocument{
\let\LWR@origcaptionof\captionof

\renewcommand*{\captionof}{%
\LWR@stoppars
\LWR@origcaptionof%
}
}

\end{warpHTML}
```

65 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A "sidetoc" is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular LaTeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output:
```
\begin{warpHTML}
```

65.1 Reading and printing the TOC

`\LWR@myshorttoc` `{`⟨*toc/lof/lot/sidetoc*⟩`}`

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular LaTeX, the file is not reset after being read, since the sideTOC may be referred to again in each HTML page.

```
\newcommand*{\LWR@myshorttoc}[1]{%
\LWR@traceinfo{LWR@myshorttoc: #1}%
\LWR@ensuredoingapar%
```

Only if the file exists:

```
\IfFileExists{\jobname.#1}{%
\LWR@traceinfo{LWR@myshorttoc: loading}%
```

⚠ Make `@` a regular letter. Many of the commands in the file will have `@` characters in them, so `@` must be made a regular letter.

⚠ disabled For `pdflatex`, also change to `latin1` encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it `utf8`.

```
\begingroup%
% \ifxetexorluatex%
% \else
% \inputencoding{latin1}% currently disabled
% \fi
\makeatletter%
```

Read in the TOC file:

```
\@input{\jobname.#1}%
% \makeatother
\endgroup%
}%
{}%
\LWR@traceinfo{LWR@myshorttoc: done}%
}
```

`\LWR@subtableofcontents` {⟨*toc/lof/lot*⟩} {⟨*sectionstarname*⟩}

Places a TOC/LOF/LOT at the current position.

```
\NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
\@ifundefined{chapter}
{\LWR@closeprevious{\LWR@depthsection}}
{\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
\LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
\@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML `nav` containing the TOC/LOF/LOT:

```
\LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
\LWR@myshorttoc{#1}
```

Close the `nav`:

```
\LWR@htmlelementclassend{nav}{#1}
}
```

`\@starttoc` {⟨*ext*⟩}

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
\let\LWR@orig@starttoc\@starttoc

\renewcommand{\@starttoc}[1]{
\LWR@htmlelementclass{nav}{#1}
\LWR@orig@starttoc{#1}
\LWR@htmlelementclassend{nav}{#1}
}
```

Bool `LWR@copiedsidetoc` Used to only copy the toc file to the sidetoc a single time.

(listings and perhaps other packages would re-use `\tableofcontents` for their own purposes, causing the sidetoc to be copied more than once, and thus end up empty.)

```
\newbool{LWR@copiedsidetoc}
\boolfalse{LWR@copiedsidetoc}
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. newfloat uses `\listoffigures` for all future float types.

```
\AtBeginDocument{
\let\LWR@origtableofcontents\tableofcontents

\renewcommand*{\tableofcontents}{%
```

Do not print the table of contents if formatting for a word processor, which will presumably auto-generate its own updated table of contents:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{

=== table of contents ===

}
{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
    \ifbool{LWR@copiedsidetoc}{}{%
        \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
        \booltrue{LWR@copiedsidetoc}%
    }%
    \LWR@printpendingfootnotes
    \LWR@origtableofcontents
}
}% \tableofcontents
}% AtBeginDocument
```

\listoffigures

```
\let\LWR@origlistoffigures\listoffigures

\renewcommand*{\listoffigures}{
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{

=== list of figures ===

}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoffigures
}
}
```

\listoftables

```
\let\LWR@origlistoftables\listoftables

\renewcommand*{\listoftables}{
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{

=== list of tables ===

}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoftables
}
}
```

65.2 High-level TOC commands

\listof {⟨*type*⟩} {⟨*title*⟩}

Emulate the \listof command from the float package (section 151). Used to create lists of custom float types. Also used to redefine the standard LaTeX \listoffigures and \listoftables commands.

```
\NewDocumentCommand{\listof}{m +m}{%
\LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
\expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
\immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
    \jobname.\csuse{ext@#1}\relax
}
```

65.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

CSS related to sideTOC:

`nav.sidetoc`: The entire sideTOC.

`div.sidetoctitle`: The title.

`div.sidetoccontents`: The table of contents.

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Ctr `SideTOCDepth` Controls how deep the side-TOC gets. Use a standard LaTeX section level similar to `tocdepth`.

```
\newcounter{SideTOCDepth}
\setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
\newcommand{\sidetocname}{Contents}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@sidetoc` Creates the actual side-TOC.

```
\newcommand*{\LWR@sidetoc}{
\LWR@forcenewpage
\LWR@stoppars

```

The entire sideTOC is placed into a `nav` of class `sidetoc`.

```
\LWR@htmlelementclass{nav}{sidetoc}
```

```
6764
6765 \setcounter{tocdepth}{\value{SideTOCDepth}}
6766
```

The title is placed into a `<div>` of class `sidetoctitle`, and may contain paragraphs.

```
6767 \begin{BlockClass}{sidetoctitle}
6768 \sidetocname
6769 \end{BlockClass}
```

The table of contents is placed into a `<div>` of class `sidetoccontents`.

```
6770 \begin{BlockClass}{sidetoccontents}
6771 \LinkHome
6772
6773 \LWR@myshorttoc{sidetoc}
6774 \end{BlockClass}
6775 \LWR@htmlelementclassend{nav}{sidetoc}
6776 }
```

65.4 Low-level TOC line formatting

`\numberline` `{`⟨*number*⟩`}`

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
6777 \newcommand*{\LWR@numberline}[1]{%
6778 \LWR@sectionnumber{#1}\quad%
6779 }
6780
6781 \LetLtxMacro\numberline\LWR@numberline
```

`\hypertoc` `{`⟨*1: depth*⟩`}` `{`⟨*2: type*⟩`}` `{`⟨*3: name*⟩`}` `{`⟨*4: page*⟩`}`

Called by `\l@section`, etc. to create a hyperlink to a section.

The `autopage` label is always created just after the section opens.

#1 is depth

#2 is `section`, `subsection`, etc.

#3 the text of the caption

#4 page number

```
\NewDocumentCommand{\hypertoc}{m m +m m}{%
\LWR@traceinfo{hypertoc !#1!#2!#3!#4!}%
```

Respond to tocdepth:

```
\ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
        \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4%
    }{#3}{toc#2}%
    \LWR@stoppars%
}
{}
\LWR@traceinfo{hypertoc done}%
}
```

Ctr `lofdepth` TOC depth for figures.

```
\@ifclassloaded{memoir}{}{
\newcounter{lofdepth}
\setcounter{lofdepth}{1}
}
```

Ctr `lotdepth` TOC depth for tables.

```
\@ifclassloaded{memoir}{}{
\newcounter{lotdepth}
\setcounter{lotdepth}{1}
}
```

`\hypertocfloat` {⟨*1: depth*⟩} {⟨*2: type*⟩} {⟨*3: ext of parent*⟩} {⟨*4: caption*⟩} {⟨*5: page*⟩}

#1 is depth

#2 is `figure`, `table`, etc.

#3 is `lof`, `lot`, of the parent.

#4 the text of the caption

#5 page number

```
\newcommand{\hypertocfloat}[5]{%
\LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
\@ifundefined{c@#3depth}{%
\newcounter{#3depth}%
\setcounter{#3depth}{1}%
}{}%
```

Respond to `lofdepth`, etc.:

```
\LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
\ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
    \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
    \#autoid-\arabic{LWR@nextautoid}}%
    {#4}{toc#2}%
    \LWR@stoppars%
}{}%
}
```

Automatically called by `\contentsline`:

```
\renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
\DeclareDocumentCommand{\l@chapter}{m m}
    {\hypertoc{0}{chapter}{#1}{#2}}
\renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
\renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
\renewcommand{\l@subsubsection}[2]
    {\hypertoc{3}{subsubsection}{#1}{#2}}
\renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
\renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
\renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
\renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

\end{warpHTML}
```

66 Index and glossary

See:
http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output:

```
\begin{warpHTML}

\newcounter{LWR@autoindex}
\setcounter{LWR@autoindex}{0}

\newcounter{LWR@autoglossary}
\setcounter{LWR@autoglossary}{0}
```

`\printindex`

```
\let\LWR@origprintindex\printindex

\renewcommand*{\printindex}
{
\LWR@startpars
\LWR@origprintindex
}
```

Env `theindex`

```
\@ifundefined{chapter}
{\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
{\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}

\renewenvironment*{theindex}{%
\LWR@indexsection{\indexname}%
\let\item\LWR@indexitem%
\let\subitem\LWR@indexsubitem%
\let\subsubitem\LWR@indexsubsubitem%
}{}
```

`\LWR@indexitem`

```
\newcommand{\LWR@indexitem}{

```

```
\InlineClass{indexitem}{}
}
```

\LWR@indexitem

```
\newcommand{\LWR@indexsubitem}{

\InlineClass{indexsubitem}{}
}
```

\LWR@indexitem

```
\newcommand{\LWR@indexsubsubitem}{

\InlineClass{indexsubsubitem}{}
}
```

\@wrindex {⟨*term*⟩} Redefined to write the LWR@autoindex counter instead of page

```
\def\LWR@wrindex#1{%
\addtocounter{LWR@autoindex}{1}%
\LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}%
\protected@write\@indexfile{}%
{\string\indexentry{#1}{\arabic{LWR@autoindex}}}%
\endgroup
\@esphack}

\let\@wrindex\LWR@wrindex
```

\@wrglossary {⟨*term*⟩} Redefined to write the LWR@latestautopage counter instead of page

```
\def\@wrglossary#1{%
\addtocounter{LWR@autoglossary}{1}%
\LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
\protected@write\@glossaryfile{}%
{\string\glossaryentry{#1}{\theLWR@autoglossary}}%
\endgroup
\@esphack}
```

\hyperindexref {⟨*autosecnumber*⟩}

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

```
\newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}

\end{warpHTML}
```

for PRINT output: A null command for print mode, in case hyperref was not used:

```
\begin{warpprint}
\newcommand{\hyperindexref}[1]{#1}
\end{warpprint}
```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
\begin{warpall}
\providecommand{\glo@name}{}
\end{warpall}
```

67 Bibliography presentation

for HTML output:

```
\begin{warpHTML}
```

Env `thebibliography` To emphasize document titles in the bibliography, the following redefines `\em` inside `thebibliography` to gather everything until the next closing brace, then display these tokens with `\textit`.

Adapted from `embracedef.sty`, which is by Takayuki YATO*:*
https://gist.github.com/zr-tex8r/b72555e3e7ad2f0a37f1

```
\AtBeginDocument{
\AtBeginEnvironment{thebibliography}{
\providecommand*{\LWR@newem}[1]{\textit{#1}}

\renewrobustcmd{\em}{%
  \begingroup
    \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
    \afterassignment\LWR@em@after
    \toks@\bgroup
}

\def\LWR@em@finish#1{%
    \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
  \endgroup
  \LWR@em@after\egroup
}
}% \AtBeginEnvironment{thebibliography}
}% \AtBeginDocument


\end{warpHTML}
```

68 Restoring original formatting

for HTML output:

```
\begin{warpHTML}
```

`\LWR@restoreorigformatting`

Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating SVG math or a `lateximage`. A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

```
\newcommand*{\LWR@restoreorigformatting}{%
\LWR@traceinfo{LWR@restoreorigformatting}%
\linespread{1}%


\LetLtxMacro\ref\LWR@origref%{} syntax highlighting
\RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
\RenewDocumentEnvironment{BlockClass}{o m}{}{}%
\renewcommand{\BlockClassSingle}[2]{##2}%
\LetLtxMacro{\hspace}{\LWR@orighspace}%
\LetLtxMacro\rule\LWR@origrule%
\let\,\LWR@origcomma% disable HTML short unbreakable space
\let\textellipsis\LWR@origtextellipsis%
\let\textless\LWR@origtextless%
\let\textgreater\LWR@origtextgreater%
\LetLtxMacro{\textrm}{\LWR@origtextrm}%
\LetLtxMacro{\textsf}{\LWR@origtextsf}%
\LetLtxMacro{\texttt}{\LWR@origtexttt}%
\LetLtxMacro{\textbf}{\LWR@origtextbf}%
\LetLtxMacro{\textmd}{\LWR@origtextmd}%
\LetLtxMacro{\textit}{\LWR@origtextit}%
\LetLtxMacro{\textsl}{\LWR@origtextsl}%
\LetLtxMacro{\textsc}{\LWR@origtextsc}%
\LetLtxMacro{\textup}{\LWR@origtextup}%
\LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
\LetLtxMacro{\emph}{\LWR@origemph}%
\LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
\LetLtxMacro{\sffamily}{\LWR@origsffamily}%
\LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
\LetLtxMacro{\bfseries}{\LWR@origbfseries}%
\LetLtxMacro{\mdseries}{\LWR@origmdseries}%
\LetLtxMacro{\upshape}{\LWR@origupshape}%
\LetLtxMacro{\slshape}{\LWR@origslshape}%
\LetLtxMacro{\scshape}{\LWR@origscshape}%
\LetLtxMacro{\itshape}{\LWR@origitshape}%
\LetLtxMacro{\em}{\LWR@origem}%
\LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
\let\sp\LWR@origsp%
```

```
\let\sb\LWR@origsb%
\LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
\LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
\LetLtxMacro\textsubscript\LWR@origtextsubscript%
\LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
\LetLtxMacro\underline\LWR@origunderline%
\let~\LWR@origtilde%
\let\enskip\LWR@origenskip%
\let\quad\LWR@origquad%
\let\qquad\LWR@origqquad%
\LetLtxMacro{\tabular}{\LWR@origtabular}%
\LetLtxMacro{\endtabular}{\LWR@origendtabular}%
\LetLtxMacro\toprule\LWR@origtoprule%
\LetLtxMacro\midrule\LWR@origmidrule%
\LetLtxMacro\cmidrule\LWR@origcmidrule%
\LetLtxMacro\bottomrule\LWR@origbottomrule%
\LetLtxMacro\addlinespace\LWR@origaddlinespace%
\LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
\LetLtxMacro\specialrule\LWR@origspecialrule%
\let\newline\LWR@orignewline%
\LetLtxMacro{\raisebox}{\LWR@origraisebox}%
\LetLtxMacro\includegraphics\LWR@origincludegraphics%
\LetLtxMacro{\scalebox}{\LWR@origscalebox}%
\LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
\let\reflectbox\LWR@origreflectbox%
\LetLtxMacro\resizebox\LWR@origresizebox%
\let\framebox\LWR@origframebox%
\let\makebox\LWR@origmakebox%
\let\fbox\LWRprint@fbox%
\let\fboxBlock\LWRprint@fbox%
\LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
\LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
\LetLtxMacro{\minipage}{\LWR@origminipage}%
\let\endminipage\LWR@origendminipage%
\LetLtxMacro{\parbox}{\LWR@origparbox}%
\let\TeX\LWR@origTeX%
\let\LaTeX\LWR@origLaTeX%
\let\LaTeXe\LWR@origLaTeXe%
\renewcommand*{\Xe}{X\textsubscript{E}}%


\LetLtxMacro\@ensuredmath\LWR@origensuredmath%


\csletcs{equation*}{LWR@origequationstar}%
\csletcs{endequation*}{LWR@origendequationstar}%
%
\LWR@restoreorigaccents%
\LWR@restoreoriglists%
%
\LWR@FBcancel%
```

```
6993 }

6994 \end{warpHTML}
```

69 Math

69.1 Limitations

69.1.1 Rendering tradeoffs

Math rendering

Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.

SVG files

In its current implementation, rendering math as images creates a new SVG file for each expression. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and check-summing may be used to remove the need for duplicate files.

SVG inline

Another approach could be to in-line the SVG files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files

Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML

Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

69.1.2 SVG option

SVG math option

For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[15], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML

[15] See section 295 regarding fonts and fractions.

alt attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size

The size of the math and text used in the SVG image may be adjusted by setting \LateximageFontSizeName to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

SVG math copy/paste

For SVG math, text copy/paste from the HTML <alt> tags lists the equation number or tag for single equations, along with the LaTeX code for the math expression. For $\mathcal{AMS}$ environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred $\mathcal{AMS}$ environment, although the \tag macro will still appear inside the LaTeX math expression.

69.1.3 MathJax option

MathJax math option

Prog MathJax

The popular MathJax alternative (mathjax.org) may be used to display math.

When MathJax is enabled, math is rendered twice:

1. As regular LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of LaTeX, and
2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

MathJax limitations

Prog MathJax

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, \tagged equations have the chapter number prepended in HTML output, unlike LaTeX. \tag* equations (correctly) do not. This may be improved with future versions of the MathJax support script.

  https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY

subequations

- MathJax itself does not support subequations. This may be improved by parsing the LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a lateximage, and therefore also inside a Ti*k*z or picture

environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

- Usage of siunitx inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

⚠ siunitx inside an equation

https://github.com/burnpanck/MathJax-siunitx

Also see section 8.5.5.

tabbing

- A `tabbing` environment is emulated using an HTML `<pre>`.While MathJax is enabled inside `tabbing`, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, bigdelim, units, and nicefrac, along with occasionally-used macros such as `\footnote` and `\relax`.

custom MathJax macros

- MathJax does not automatically support custom LaTeX macros, but they may be created by the user inside a math expression:

```
\begin{document}
(...)
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
    % New macros for MathJax are
    % placed inside a math expression:
    \(
        \newcommand{\expval}[1]{\langle#1\rangle}
        \newcommand{\abs}[1]{\lvert#1\rvert}
    \)
}{}
\end{warpHTML}
```

69.2 Inline and display math

for HTML output:

```
\begin{warpHTML}
```

\$ Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar \$, print it inside a span to avoid it being interpreted by MathJax, unless are inside a lateximage, in which case it will not be seen by MathJax.

```
\let\LWR@origtextdollar\$
```

```
\renewcommand*{\$}{%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{\LWR@origtextdollar}%
{\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
}
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
\newcounter{LWR@externalfilecnt}
```

```
\LetLtxMacro\LWR@origdollar$
\LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting
```

```
\LetLtxMacro\LWR@origopenparen\(
\LetLtxMacro\LWR@origcloseparen\)
\LetLtxMacro\LWR@origopenbracket\[
\LetLtxMacro\LWR@origclosebracket\]
```

`$` `$$` Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:

```
\begingroup
\catcode`\$=\active%
\protected\gdef${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
\protected\gdef\LWR@doubledollar$#1$${
```

If MathJax or formatting for a word processor, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between `\[` and `\]`:

```
{

        \textbackslash[%
        \LWR@HTMLsanitize{#1}%
        \textbackslash]

}% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the LaTeX code:

```
{% not mathjax

```

```
        \begin{lateximage}%
        [\textbackslash{[} \LWR@HTMLsanitize{#1} \textbackslash{]}]%
        \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
        \end{lateximage}%

}% not mathjax
}%
```

\LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
\protected\gdef\LWR@subsingledollar#1{%
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between \(and \):

```
{%
        {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}%
}% mathjax
```

For SVG, print the math inside a lateximage, with an <alt> tag of the LaTeX code:

```
{% not mathjax
        \begin{lateximage}%
        [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
        \LWR@origdollar#1\LWR@origdollar%
        \end{lateximage}%
}%
}

\protected\gdef\LWR@singledollar#1${%
\LWR@subsingledollar{#1}%
}
```

\(\) Redefine to the above dollar macros.

```
\protected\gdef\(#1\){%
$#1$}
\protected\gdef\[#1\]{$$#1$$}

\endgroup
```

\@ensuredmath {⟨*expression*⟩} Not yet adapted to lwarp.

```
\LetLtxMacro\LWR@origensuredmath\@ensuredmath

\renewcommand{\@ensuredmath}[1]{%
\ifmmode%
    \LWR@origensuredmath{#1}%
\else%
    \LWR@subsingledollar{\relax#1}%
\fi%
}
```

Remove the old `math` and `displaymath` environments:

```
\let\math\relax
\let\endmath\relax
\let\displaymath\relax
\let\enddisplaymath\relax
```

Env `math` Set math mode then typeset the body of what was between the begin/end. See the environ package for `\BODY`.

```
\NewEnviron{math}{\expandafter\(\BODY\)}
```

Env `displaymath` Set math mode then typeset the body of what was between the begin/end. See the environ package for `\BODY`.

```
\NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

69.3 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```
\newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside "`\(`" and "`\)`" characters. They are printed to HTML output, not interpreted by LaTeX.

```
\newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
\ifcsdef{thechapter}{
\InlineClass{hidden}{
\textbackslash(
\textbackslash{}seteqsection \{\thechapter\}
\textbackslash)
}
}
{}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
\ifthenelse{\cnttest{\value{equation}}>0}
{
```

Tell MathJax that the next set of equations begins with the current LaTeX equation number, plus one.

```
    \setcounter{LWR@nextequation}{\value{equation}}
    \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside "`\(`" and "`\)`" characters, to be printed to HTML, not interpreted by LaTeX.

```
    \InlineClass{hidden}{
        \textbackslash(
        \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
        \textbackslash)
    }
}{}% not eq > 0
}
```

`\LWR@hidelatexequation` `{`⟨*environment*⟩`}` `{`⟨*contents*⟩`}`

Creates the LaTeX version of the equation inside an HTML comment.

```
\NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
\LWR@stoppars
\LWR@htmlopencomment

```

Start the LaTeX math environment inside the HTML comment:

```
\begingroup
\csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their LaTeX meanings.

```
\LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 69.5.1.

Print the contents of the equation:

```
#2
```

End the LaTeX math environment inside the HTML comment:

```
\csuse{LWR@origend#1}
\endgroup

```

Close the HTML comment and resume HTML paragraph handling:

```
\LWR@htmlclosecomment
\LWR@startpars
}
```

`\LWR@addmathjax` `{`⟨*environment*⟩`}` `{`⟨*contents*⟩`}`

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by LaTeX.

```
\NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed "`\(`" and "`\)`" characters.

```
\LWR@origtilde\LWR@orignewline
\textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
\textbackslash{}end\{#1\}
\LWR@orignewline
}
```

69.4 Equation environment

Remember existing `equation` environment:

```
\let\LWR@origequation\equation
\let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
\let\equation\relax
\let\endequation\relax
```

Env `equation` The new `equation` environment is created with `\NewEnviron` (from the environ package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the LaTeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\label`s. The contents are also printed to the HTML output for processing by the MathJax script.

```
\NewEnviron{equation}{%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

MathJax output:

```
{
```

Print commands to syncronize MathJax's equation number and format to the current LaTeX chapter/section and equation number:

```
    \LWR@syncmathjax
```

Print the LaTeX math inside an HTML comment:

```
    \LWR@hidelatexequation{equation}{\BODY}
}
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the LaTeX equation environment commmands, and the contents of the environment's `\BODY`.

```
{% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
    \begin{lateximage}[(\LWR@equationtag) \textbackslash{begin\{equation\}} %
    \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %
    \textbackslash{end\{equation\}}]% alt tag
```

Create the actual LaTeX-formatted equation inside the `lateximage` using the contents of the environment.

```
    \LWR@origequation
    \BODY% contents collected by NewEnviron
    \LWR@origendequation
    \end{lateximage}%
}% not mathjax

```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
}[%
    \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
    {%
        \LWR@addmathjax{equation}{\BODY}%
    }{}%

]
```

Env `equation*`

```
\csletcs{LWR@origequationstar}{equation*}
\csletcs{LWR@origendequationstar}{endequation*}
\renewenvironment*{equation*}
{\displaymath}
{\enddisplaymath}
```

69.5 AMS Math environments

69.5.1 Support macros

Bool `LWR@amsmultline` True if processing a multline environment.

To compensate for `multline`-spefific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an amsmath environment which is not a `multline` environment and not an `equation`.

```
\newbool{LWR@amsmultline}
\boolfalse{LWR@amsmultline}
```

`\LWR@htmlmathlabel` {⟨*label*⟩}

lwarp points `\ltx@label` here. This is used by `\label` when inside a LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the LaTeX original, modified by lwarp, then by amsmath, then by cleveref.

```
\newcommand*{\LWR@htmlmathlabel}[1]{%
\LWR@traceinfo{LWR@htmlmathlabelb #1}%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

The combined LaTeX & HTML label is printed in a `\text` field:

```
    \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
    \ifbool{LWR@amsmultline}{}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the amsmath package.)

```
    \LWR@htmlclosecomment%
    \LWR@origltx@label{#1}%
    \LWR@htmlopencomment%
    }% text
```

```
}% mathjax
{%
    \LWR@origltx@label{#1}%
}%
}
```

`\LWR@beginhideamsmath` Starts hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@beginhideamsmath}{
\LWR@stoppars
\LWR@origtilde\LWR@orignewline
\LWR@htmlopencomment

\begingroup
\LWR@restoreorigformatting
}
```

`\LWR@endhideamsmath` Ends hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@endhideamsmath}{
\endgroup

\LWR@htmlclosecomment
\LWR@orignewline
\LWR@startpars
}
```

69.5.2 Environment patches

The following amsmath environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an $\mathcal{AMS}$ package, and thus requires special handling.

For SVG math: Each envrionment is encapsulated inside a `lateximage` environment, along with a special optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the $\mathcal{AMS}$ environment.

For MathJax: Each environment is syched with LaTeX's equation numbers, typeset with LaTeX inside an HTML comment, then printed to HTML output for MathJax to process.

Env `eqnarray` This environmnet is not an $\mathcal{AMS}$ environment and thus its body is not automatically captured, so the environ package is used to capture the environment into `\BODY`.

```
\let\LWR@origeqnarray\eqnarray
\let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
\newbool{LWR@numbereqnarray}
\booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from fancybox):

```
\newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

If MathJax, the environment contents (the `\BODY`) are executed in a HTML comment to trigger the correct equation number increment (if not starred), then are included verbatim in the output for MathJax to interpret:

```
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \ifbool{LWR@numbereqnarray}
    {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
        \LWR@beginhideamsmath
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
        \LWR@addmathjax{eqnarray}{\BODY}
    }%
    {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
    \LWR@addmathjax{eqnarray*}{\BODY}
    }% LWR@numbereqnarray
}% mathjax
{% not mathjax
    \ifbool{LWR@numbereqnarray}
```

```
    {
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
        \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
            \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \end{lateximage}
    }%
    {% not LWR@numbereqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```
        \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
        \LWR@origeqnarray
        \BODY
        \nonumber
        \LWR@origendeqnarray
        \end{lateximage}
    }% LWR@numbereqnarray
}% not mathjax
```

Default to number equations in the future:

```
\booltrue{LWR@numbereqnarray}
}
```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```
\RenewEnviron{eqnarray}
{%

\LWR@eqnarrayfactor

}
```

The starred version is patched to turn off the numbering:

```
\csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

The following $\mathcal{AMS}$ environments are more easily patched in-place:

Env `multline`

```
\BeforeBeginEnvironment{multline}{
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
}
}

\AfterEndEnvironment{multline}{
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env `multline*`

```
\BeforeBeginEnvironment{multline*}{
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{multline*}]
}
}

\AfterEndEnvironment{multline*}{
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline*}{\the\@envbody}
```

```
}
{\end{lateximage}}

}
```

Env gather

```
\BeforeBeginEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
}
}

\AfterEndEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{gather}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env gather*

```
\BeforeBeginEnvironment{gather*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbody{gather*}]
}
}

\AfterEndEnvironment{gather*}{
```

```
7313
7314 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7315 {
7316     \LWR@endhideamsmath
7317     \LWR@addmathjax{gather*}{\the\@envbody}
7318 }
7319 {\end{lateximage}}
7320
7321 }
```

Env `align`

```
7322 \BeforeBeginEnvironment{align}{
7323
7324 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7325 {
7326     \LWR@syncmathjax
7327     \boolfalse{LWR@amsmultline}
7328     \LWR@beginhideamsmath
7329 }
7330 {
7331     \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
7332 }
7333 }
7334
7335 \AfterEndEnvironment{align}{
7336
7337 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7338 {
7339     \LWR@endhideamsmath
7340     \LWR@addmathjax{align}{\the\@envbody}
7341 }
7342 {\end{lateximage}}
7343
7344 }
```

Env `align*`

```
7345 \BeforeBeginEnvironment{align*}{
7346
7347 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7348 {
7349     \LWR@syncmathjax
7350     \boolfalse{LWR@amsmultline}
7351     \LWR@beginhideamsmath
7352 }
7353 {
7354     \begin{lateximage}[\LWR@amsmathbody{align*}]
```

```
}
}

\AfterEndEnvironment{align*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{align*}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env `flalign`

```
\BeforeBeginEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
}
}

\AfterEndEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{flalign}{\the\@envbody}
}
{\end{lateximage}}

}
```

Env `flalign*`

```
\BeforeBeginEnvironment{flalign*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
```

```
7397     \LWR@beginhideamsmath
7398 }
7399 {
7400     \begin{lateximage}[\LWR@amsmathbody{flalign*}]
7401 }
7402 }
7403
7404 \AfterEndEnvironment{flalign*}{
7405
7406 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7407 {
7408     \LWR@endhideamsmath
7409     \LWR@addmathjax{flalign*}{\the\@envbody}
7410 }
7411 {\end{lateximage}}
7412
7413 }

7414 \end{warpHTML}
```

70 Lateximages

70.1 Description

Env `lateximage` A `lateximage` is a piece of the document which is typeset in LaTeX then included in the HTML output as an image. This is used for math if SVG math is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 68.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `<span>` is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

SVG image font size

The size of the math and text used in the SVG image may be adjusted by setting \LateximageFontSizeName to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

70.2 Support counters and macros

for HTML output:

```
\begin{warpHTML}
```

Ctr LWR@lateximagenumber

Sequence the images.

```
\newcounter{LWR@lateximagenumber}
\setcounter{LWR@lateximagenumber}{0}
```

Ctr LWR@lateximagedepth

Do not create \lateximage inside of \lateximage.

```
\newcounter{LWR@lateximagedepth}
\setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
\edef\LWR@hashmark{\string#} % for use in \write
\edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr LWR@LIpage

Used to reference the PDF page number of a lateximage to be written into lateximages.txt.

```
\newcounter{LWR@LIpage}
```

```
\end{warpHTML}
```

70.3 Font size

for HTML & PRINT:

```
\begin{warpall}
```

\LateximageFontSizeName

Declares how large to write text in the \lateximage. The .svg file text size should blend well with the surrounding HTML text size.

⚠ no backslash

Do not include the leading backslash in the name.

```
\newcommand*{\LateximageFontSizeName}{large}
```

```
\end{warpall}
```

70.4 Sanitizing math expressions for HTML

for HTML output:

```
\begin{warpHTML}
```

`\LWR@HTMLsanitize` {⟨*text*⟩}

Math expressions are converted to `lateximage`s, and some math environments may contain "&", "<", or ">", which should not be allowed inside an HTML `<alt>` tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way…

```
\newcommand{\LWR@HTMLsanitize}[1]{%
\begingroup%
\LWR@FBcancel%
\protect\StrSubstitute{\detokenize{#1}}%
{\detokenize{&}}%
{\detokenize{&amp;}}[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}%
{\detokenize{&lt;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}%
{\detokenize{&gt;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{##}}%
{\#}%
[\LWR@strresult]%
\LWR@strresult%
\endgroup%
}
```

`\LWR@HTMLsanitizeexpand` {⟨*text*⟩}

This version expands the argument before sanitizing it.

```
\newcommand{\LWR@HTMLsanitizeexpand}[1]{%
\begingroup%
\LWR@FBcancel%
\protect\StrSubstitute{\detokenize\expandafter{#1}}%
{\detokenize{&}}%
{\detokenize{&amp;}}%
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}%
{\detokenize{&lt;}}%
```

```
[\LWR@strresult]%
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}%
{\detokenize{&gt;}}%
[\LWR@strresult]%
\LWR@strresult%
\endgroup%
}
```

70.5 Equation numbers

Ctr `LWR@startingequation` For use with `lateximage` and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```
\newcounter{LWR@startingequation}

\@ifundefined{chapter}
{
\renewcommand{\theLWR@startingequation}{%
\arabic{LWR@startingequation}%
}
}
{% chapter defined
\renewcommand{\theLWR@startingequation}{%
\ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}%
\arabic{LWR@startingequation}%
}
}
```

Bool `LWR@isstartingequation` True for the first equation tag, false for later tags in the same environment.

```
\newbool{LWR@isstartingequation}
```

`\LWR@startingequationtag` Prints the starting equation number or tag.

```
\let\LWR@startingequationtag\theLWR@startingequation
```

`\LWR@equationtag` Prints the ending equation number or tag.

```
\let\LWR@equationtag\theequation
```

Only if SVG math, patch `\tag` after packages have loaded, in case someone else modified `\tag`.

```
\AtBeginDocument{

\ifbool{mathjax}{}{% not mathjax
```

`\LWR@remembertag` {⟨*tag*⟩}

For use inside the math environments while using SVG math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```
\NewDocumentCommand{\LWR@remembertag}{m}{%
\ifbool{LWR@isstartingequation}%
{%
    \global\boolfalse{LWR@isstartingequation}%
    \xdef\LWR@startingequationtag{#1}%
}%
{}%
\xdef\LWR@equationtag{#1}%
}%
```

Patches for $\mathcal{AMS}$ math `\tag` macro to remember the first tag:

```
\LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
\LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@

\renewcommand*{\make@df@tag@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@{#1}%
}

\renewcommand*{\make@df@tag@@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@@{#1}%
}

}% not mathjax
}% AtBeginDocument
```

70.6 HTML <alt> tags

`\LWR@amsmathbody` {⟨*envname*⟩} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the <alt> tag.

```
\newcommand*{\LWR@amsmathbody}[1]
{%
\textbackslash\{begin\}\{#1\} %
```

```
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
\textbackslash\{end\}\{#1\}%
}
```

`\LWR@amsmathbodynumbered` {⟨*envname*⟩} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag, prefixed by the equation numbers.

```
\newcommand*{\LWR@amsmathbodynumbered}[1]
{%
\ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
{(\LWR@equationtag)}%
{(\LWR@startingequationtag--\LWR@equationtag)} %
\LWR@amsmathbody{#1} %
}
```

70.7 `lateximage`

Env `lateximage` [⟨*<alt> tag*⟩]

```
\catcode`\$=\active%


\NewDocumentEnvironment{lateximage}{O{image}}
{%
\LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
```

Nested `lateximage`s remain one large `lateximage`:

```
\ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
{%
    \addtocounter{LWR@lateximagedepth}{1}%
}%
```

Otherwise, this is the outer-most lateximage:

```
{% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
    \setcounter{LWR@startingequation}{\value{equation}}%
    \addtocounter{LWR@startingequation}{1}%
    \booltrue{LWR@isstartingequation}%
    \let\LWR@startingequationtag\theLWR@startingequation%
    \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
    \addtocounter{LWR@lateximagenumber}{1}%
    \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
    \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Next file:

```
    \addtocounter{LWR@externalfilecnt}{1}%
    \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be. `\setcounterpageref` assigns `LWR@LIpage` to the page number for the reference `LWRlateximageXXX`:

```
    \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
    \LWR@traceinfo{lateximage: LWR@LIpage is \arabic{LWR@LIpage}}%
```

Create an HTML span which will hold the comment which contains the `pdftotext` translation of the image's page, and also will hold the link to the `.svg` file:

```
    \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
    class="lateximagesource"} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
    \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
    \immediate\write\LWR@lateximagesfile{|\arabic{LWR@LIpage}|\arabic{LWR@externalfilecnt}|}%
```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by `pdftotext`.

```
    \LWR@traceinfo{lateximage: about to create open comment}%
    \LWR@htmlopencomment%
```

One level deeper:

```
    \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
    \LWR@traceinfo{lateximage: about to create a new page}%
    \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
    \LWR@traceinfo{lateximage: about to create minipage}%
    \LWR@origminipage{6in}%
    \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a lateximage.

```
    \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
    \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
    \def\@mpfn{footnote}%
    \def\thempfn{\thefootnote}%
    \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
    \LWR@traceinfo{lateximage: about to create label}%
    \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
    \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
    \LetLtxMacro$\LWR@origdollar%
    \catcode`\$=3% math shift
    \LetLtxMacro\(\LWR@origopenparen%
    \LetLtxMacro\)\LWR@origcloseparen%
    \LetLtxMacro\[\LWR@origopenbracket%
    \LetLtxMacro\]\LWR@origclosebracket%
}% end of outer-most lateximage
\LWR@traceinfo{lateximage: finished start of environment}%
}% end of \begin{lateximage}
```

`\endlateximage` When the environment closes:

```
{% start of \end{lateximage}
\LWR@traceinfo{lateximage: starting end of environment}%
```

Nested more than one deep?

```
\ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
{\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
{% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
    \LWR@origendminipage%
    \LWR@orignewpage%
    \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```
    \LWR@origvspace*{.5\baselineskip}%
    \LWR@htmlclosecomment{}\LWR@orignewline%
    \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

```
    \LWR@subinlineimage[#1]{lateximage}%
    {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
    \LWR@htmltag{/span}%
    \ifbool{HTMLDebugComments}{%
        \LWR@htmlcomment{End of lateximage}%
    }{}%
% \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
    \addtocounter{LWR@lateximagedepth}{-1}%
}% end of outer-most lateximage
\LWR@traceinfo{lateximage: done}%
}%
\catcode`\$=3% math shift
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
% \newenvironment{lateximage}[1][]{\minipage{\linewidth}}{\endminipage}
\newenvironment{lateximage}[1][]{}{}
\end{warpprint}
```

71 center, flushleft, flushright

for HTML output:

```
\begin{warpHTML}
```

Env center Replace center functionality with css tags:

```
\renewenvironment*{center}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[text-align:center]{center}}
{\BlockClass{center}}
}
{\endBlockClass}
```

Env flushright

```
\renewenvironment*{flushright}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[text-align:right]{flushright}}
{\BlockClass{flushright}}
}
{\endBlockClass}
```

Env flushleft

```
\renewenvironment*{flushleft}
{
\LWR@forcenewpage
\ifbool{FormatWP}
```

```
{\BlockClass[text-align:left]{flushleft}}
{\BlockClass{flushleft}}
}
{\endBlockClass}

\end{warpHTML}
```

72 Pre-loaded packages

for HTML output:

```
\begin{warpHTML}
```

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
\@ifpackageloaded{textcomp}
{
\LWR@origRequirePackage{lwarp-textcomp}
}
{}
```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```
\@ifpackageloaded{graphics}
{
\LWR@origRequirePackage{lwarp-graphics}
}
{}

\end{warpHTML}
```

73 Siunitx

Pkg `siunitx` The lwarp core passes a few options to siunitx.

⚠ per-mode Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ math mode required Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the siunitx extension for MathJax is not currently hosted at any public CDN, thus siunitx is not usable with MathJax unless a local copy of this extension is created first.

for HTML output:

```
\begin{warpHTML}
```

Options for siunitx:

```
\PassOptionsToPackage{
    detect-mode=true,
    per-mode=symbol,% fraction is not seen by pdftotext
%     text-celsius = {\protect\LWRsiunitx@degree{}C},
%     text-degree = {\protect\LWRsiunitx@degree},
}{siunitx}

\end{warpHTML}
```

74 Graphics print-mode modifications

74.1 General limitations

⚠ .pdf image files

⚠ no file extension

For `\includegraphics` with `.pdf` files, the user should provide a `.pdf` image file, and also a `.svg`, `.png`, or `.jpg` version of the same image. These should be referred to without a file extension:

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

For print output, lwarp will automatically choose the `.pdf` if available, other some other format otherwise. For HTML, one of the other formats is used instead.

Prog pdftocairo

To convert a PDF image to SVG, use the utility `pdftocairo`:

```
Enter ⇒   pdftocairo -svg filename.pdf
```

If a `.pdf` file is referred to with its file extension, a link to the `.pdf` file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

Pkg epstopdf

For `.eps` files, use epstopdf to provide a PDF version, and also provide a SVG version as well.

other image files

For `.png`, `.jpg`, or `.gif` image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

⚠ graphics vs. graphicx

If using the older graphics syntax, use both optional arguments for `\includegraphics`.

⚠ viewports

A single optional parameter is interpreted as the newer graphicx syntax. Note that viewports are not supported by warp; the entire image will be shown.

units

For `\includegraphics`, avoid `px` and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use `ex` or `em`. For fixed-sized images, use `cm`, `mm`, `in`, `pt`, or `pc`. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.

options

`\includegraphics` accepts `width` and `height`, `origin`, `rotate` and `scale`, plus a new `class` key.

HTML class

With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

`\rotatebox`

`\rotatebox` accepts the optional `origin` key.

⚠ browser support

`\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

74.2 Print-mode modifications

for PRINT output:

For print output, accept and then discard the new `class` key:

```
\begin{warpprint}
\define@key{Gin}{class}{}
```

Print-mode additions for the overpic package. See section 219 for the HTML version.

```
\AtBeginDocument{
\@ifpackageloaded{overpic}{
\newcommand*{\overpicfontsize}{12}
\newcommand*{\overpicfontskip}{14}
}{}
}
\end{warpprint}
```

75 Xcolor boxes

Pkg `xcolor`

A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of `none`, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT:

```
\begin{warpall}
```

After xparse may have been loaded ...

```
\AtBeginDocument{
```

... and *only* if xcolor was loaded:

```
\@ifpackageloaded{xcolor}{
\LWR@traceinfo{patching xcolor}
```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```
\LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
\warpprintonly{
\LetLtxMacro\LWRprint@colorboxBlock\colorbox
\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
}
```

`\fcolorbox` `[`⟨*framemodel*⟩`]` `{`⟨*framecolor*⟩`}` `[`⟨*boxmodel*⟩`]` `{`⟨*boxcolor*⟩`}` `{`⟨*text*⟩`}`

In print mode, `\fcolorbox` is modified to accept a background color of `none`.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

```
\newsavebox{\LWR@colorminipagebox}

\DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
\LWR@traceinfo{LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
#5%
\end{lrbox}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    \LWR@traceinfo{background is none}%
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
\LWR@traceinfo{background not none}%
\IfValueTF{#1}%
{%
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}%
{% no value #1
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}% no value #1
}% #4 not none
\LWR@traceinfo{LWRprint@fcolorbox done}%
}
```

`\fcolorboxBlock` `[`⟨*framemodel*⟩`]` `{`⟨*framecolor*⟩`}` `[`⟨*boxmodel*⟩`]` `{`⟨*boxcolor*⟩`}` `{`⟨*text*⟩`}`

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
\LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox
```

Env `fcolorminipage` `[`⟨*1:framemodel*⟩`]` `{`⟨*2:framecolor*⟩`}` `[`⟨*3:boxmodel*⟩`]` `{`⟨*4:boxcolor*⟩`}` `[`⟨*5:align*⟩`]` `[`⟨*6:height*⟩`]` `[`⟨*7:inner-align*⟩`]` `{`⟨*8:width*⟩`}`

In print mode, becomes a `\fcolorbox` containing a `minipage`:

```
\NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{} o m}
```

```
{%
\LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
\IfValueTF{#7}%
{\begin{minipage}[#5][#6][#7]{#8}}%
{\begin{minipage}[#5][#6][#5]{#8}}%
}%
{%
\end{minipage}%
\end{lrbox}%
\LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
    \IfValueTF{#1}%
    {%
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }%
    {% no value #1
    \IfValueTF{#3}%
    {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }% no value #1
}% #4 not none
\LWR@traceinfo{*** finished end fcolorminipage}%
}
```

\LWR@restoreorigprintxcolor Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `lateximage`.

```
\newcommand*{\LWR@restoreorigprintxcolor}{%
\LWR@traceinfo{LWR@restoreorigprintxcolor}%
\LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
\LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
\LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
\LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
\LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
}

\appto{\LWR@restoreorigformatting}{%
\LWR@restoreorigprintxcolor%
}
```

If print mode, immediately activate the print-mode enhancements for xcolor:

```
\warpprintonly{\LWR@restoreorigprintxcolor}

\LWR@traceinfo{xcolor patches done}
}{}% xcolor loaded
}% AtBeginDocument

\end{warpall}
```

76 Cleveref

Pkg cleveref cleveref package is used as-is with minor patches.

⚠ cleveref page numbers cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

Ex:

```
    \cpageref{tab:first,tab:second}
 in HTML becomes:
```

"pages **for** table 4.1 and **for** table 4.2"

See `\cpagerefFor` at page 419 to redefine the message which is printed for page number references.

loading order cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires cleveref. The user's document may not require

cleveref, thus the user may never explicitly load it, so during HTML output lwarp loads it last. If the user's document preamble uses cleveref options, or functions such as `\crefname`, then cleveref may be loaded in the user's preamble near the end, and lwarp's additional loading of cleveref will have no effect.

Table 10 on page 354 shows the data structure of the label/reference system as revised by lwarp and cleveref.

A few patches allow cleveref to work as-is:

for HTML output:

```
\begin{warpHTML}
```

`\AtEndPreable` forces cleveref to be loaded last:

```
\AtEndPreamble{
\RequirePackage{cleveref}
}
```

The following patches are applied after cleveref has loaded, and after `\AtBeginDocument`:

```
\AfterEndPreamble{
\LWR@traceinfo{Patching cleveref.}
```

`\@@setcref` {⟨*kindofref*⟩} {⟨*label*⟩}

```
\renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{}{}}
```

`\@@setcrefrange` {⟨*text*⟩} {⟨*label*⟩} {⟨*label*⟩}

```
\renewcommand{\@@setcrefrange}[3]{%
#1{\ref{#2}}{\ref{#3}}{}{}{}{}}
```

`\cpagerefFor` Redefinable word between "page(s)" and the page numbers.

```
\newcommand*{\cpagerefFor}{for}
```

`\@@setcpageref` {⟨*typeofref*⟩} {⟨*label*⟩}, where typeofref is "page" or "pages"

```
\renewcommand*{\@@setcpageref}[2]{%
#1{\cpagerefFor\ \cref{#2}}{}{}%
}
```

```
\renewcommand{\@@setcpagerefrange}[3]{%
#1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}
}% AfterEndPreamble
```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

```
\LetLtxMacro\LWR@origlabel\label
\RenewDocumentCommand{\label}{}{\LWR@newlabel}

\LetLtxMacro\LWR@origref\ref
\RenewDocumentCommand{\ref}{}{\LWR@newref}%

\LetLtxMacro\LWR@origpageref\pageref
\RenewDocumentCommand{\pageref}{}{\LWR@newpageref}

\end{warpHTML}
```

77 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output:

```
\begin{warpHTML}
```

Env `picture`

```
\BeforeBeginEnvironment{picture}{\begin{lateximage}}

\AfterEndEnvironment{picture}{\end{lateximage}}

\end{warpHTML}
```

78 Boxes and Minipages

A css flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

⚠ inline A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.

placement Minipages and parboxes will be placed side-by-side in HTML unless you place a

\newline between them.

side-by-side Side-by-side minipages may be separated by \quad, \qquad, \enskip, \hspace, \hfill, or a \rule. When inside a center environment, the result is similar in print and HTML. Paragraph tags are surpressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span There is limited support for minipages inside an HTML . An HTML cannot appear inside a . While in a , minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an "inline" format, without markup except for HTML breaks. Use \newline or \par for an HTML break.

size When using \linewidth, \textwidth, and \textheight, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages A minipage of width exactly \linewidth is automatically given no HTML width.

full-width minipages A new macro \minipagefullwidth requests that the next minipage be generated without an HTML width attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment Nested minipages adopt their parent's text alignment in HTML, whereas in regular LaTeX PDF output they do not. Use a flushleft or similar environment in the child minipage to force a text alignment.

for HTML output:
```
7795 \begin{warpHTML}
```

78.1 Counters and lengths

Ctr LWR@minipagedepth Used to only reset the line width at the outermost minipage.

```
7796 \newcounter{LWR@minipagedepth}
7797 \setcounter{LWR@minipagedepth}{0}
```

Len \WR@minipagewidth Used to convert the width into printable units.

```
7798 \newlength{\LWR@minipagewidth}
```

Len \WR@minipageheight Used to convert the height into printable units.

```
7799 \newlength{\LWR@minipageheight}
```

78.2 Footnote handling

Also see section 49 for other forms of footnotes. Minipage footnotes are gathered in section 49.5, and then placed into the document in section 78.3.

78.3 Minipage handling

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML `width`?

```
\newbool{LWR@minipagefullwidth}
\boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no `width` tag in HTML:

for HTML output:
```
\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}
\newcommand*{\minipagefullwidth}{}
\end{warpprint}
```

for HTML output:
```
\begin{warpHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
\newbool{LWR@minipagethispar}
\boolfalse{LWR@minipagethispar}
```

Env `minipage` `[`⟨*vert position*⟩`]` `[`⟨*height*⟩`]` `[`⟨*inner vert position*⟩`]` `{`⟨*width*⟩`}`

The vertical positions may be 'c', 't', or 'b'. The inner position may also be 's'.

When using `\linewidth`, `\textwidth`, or `\textheight`, these are scaled proportionally to a 6×9 inch text area.

```
\RenewDocumentEnvironment{minipage}{O{t} o O{t} m}
{%
```

Units for printing dimensions to HTML:

```
\uselengthunit{PT}%
```

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual LaTeX minipage.

```
\begingroup
```

Compute width, adjusted for frames:

```
\setlength{\LWR@minipagewidth}{#4}%
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
```

Only create a new page if not yet nested:

```
    \LWR@orignewpage%
```

Adjust virtual page size:

```
    \addtolength{\LWR@minipagewidth}{3em}% room for frames
    \setlength{\linewidth}{6in}%
    \setlength{\textwidth}{6in}%
    \setlength{\textheight}{9in}%
}{}%
\LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}%
```

Compute height:

```
\setlength{\LWR@minipageheight}{\textheight}% default unless specified
\IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
\addtocounter{LWR@minipagedepth}{1}%
```

LaTeX wants to start a paragraph for the virtual minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars%
```

If `FormatWP`, add a text frame:

```
\ifbool{FormatWP}{%

\addtocounter{LWR@thisautoidWP}{1}%
\LWR@htmltag{div id="autoidWP-\arabic{LWR@thisautoidWP}" class="wpminipage"}%

```

```
}{}%
```

Create the tag with optional alignment style:

```
\LWR@traceinfo{minipage: creating div class}%
\LWR@htmltag{div class="minipage" style="%
\ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
\ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
\ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
\ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
\ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
\ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
\ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```
\LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
\uselengthunit{PT}%
\ifbool{LWR@minipagefullwidth}%
{\boolfalse{LWR@minipagefullwidth}}%
{%
    \ifthenelse{\lengthtest{#4}=\linewidth}%
    {}%
    {width:\rndprintlength{\LWR@minipagewidth} ; }%
}%
\LWR@traceinfo{minipage: about to print the height}%
\IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
"}%
```

Finish with an empty line to start the contents on a new line.

```

% The preceding empty line is required.
```

Set the user-accessible line and text width and height values inside the virtual minipage. These do not affect the actual size of the PDF output, but are used by any reference to \linewidth, etc. inside the virtual minipage being created here.

```
\setlength{\linewidth}{#4}% the original width
\setlength{\textwidth}{6in}%
\setlength{\textheight}{9in}%
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
\LWR@origraggedright%
```

Set minipage footnotes:

```
\def\@mpfn{mpfootnote}%
\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
\let\@footnotetext\@mpfootnotetext%
```

Resume paragraph tag handling for the contents of the minipage:

```
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== begin minipage ===

}{}%
\LWR@traceinfo{minipage: finished starting the minipage}%
}% finished \minipage
{% \endminipage
```

Print pending minipage footnotes:

```
\LWR@printpendingmpfootnotes%
```

End the environment with closing tag:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== end minipage ===

}{}%
\LWR@stoppars%
```

The following used to be an actual LaTeX minipage.

```
\endgroup%

\ifbool{FormatWP}{%

\LWR@htmlelementend{div}%

}{}%
\LWR@htmldivclassend{minipage}%

\LWR@origvspace{1\baselineskip}% required for subcaption
\addtocounter{LWR@minipagedepth}{-1}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
\global\booltrue{LWR@minipagethispar}%
}
```

78.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output:

\parbox [⟨*pos*⟩] [⟨*height*⟩] [⟨*inner-pos*⟩] {⟨*width*⟩} {⟨*text*⟩}

A parbox uses the minipage code:

```
\RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
{
\LWR@traceinfo{parbox of width #4}%
\begin{minipage}[#1][#2][#3]{#4}%
#5
\end{minipage}%
}
```

\makebox (⟨*(*⟩)posn) [⟨*width*⟩] [⟨*pos*⟩] {⟨*text*⟩}

```
\LetLtxMacro\LWR@origmakebox\makebox

\RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
\IfValueTF{#2}%
{%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is `justified` alignment.

```
    {% scope
    \def\LWR@align{center}%
    \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
    \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
    \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the `width` argument:

```
    \setlength{\LWR@tempwidth}{#2}%
```

`inline-block` allows `width` and `text-alignment` to be used in a .

```
    \uselengthunit{PT}%
```

```
    \InlineClass[%
        display:inline-block ; %
        text-align:\LWR@align\ ; %
        width:\rndprintlength{\LWR@tempwidth}%
    ]%
    {makebox}%

    {#4}%
    }% scope
}%
```

Without a width argument, the text is simply used inline:

```
{#4}% no width
}
```

`\framebox` `[`⟨*width*⟩`] [`⟨*pos*⟩`] {`⟨*text*⟩`}`

```
\LetLtxMacro\LWR@origframebox\framebox

\RenewDocumentCommand{\framebox}{o o m}{%
\fbox{\makebox[#1][#2]{#3}}%
}
```

`\LWR@forceminwidth` `{`⟨*legth*⟩`}`

Sets `\LWR@atleastonept` to be at least 1pt.

```
\newlength{\LWR@atleastonept}

\newcommand*{\LWR@forceminwidth}[1]{%
\setlength{\LWR@atleastonept}{#1}%
\ifthenelse{%
    \lengthtest{\LWR@atleastonept>0pt}\AND%
    \lengthtest{\LWR@atleastonept<1pt}%
}%
{\setlength{\LWR@atleastonept}{1pt}}%
{}%
}
```

`\LWR@blackborderpadding` Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
\newcommand*{\LWR@blackborderpadding}{%
\uselengthunit{PT}%
```

```
border:\rndprintlength{\LWR@atleastonept} solid black ; %
padding:\rndprintlength{\fboxsep}%
}
```

`\fbox` `{`⟨*text*⟩`}`

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
\let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
\newcommand{\LWRhtml@fbox}[1]{%
\LWR@traceinfo{HTML fbox}%
\LWR@forceminwidth{\fboxrule}%
\InlineClass[%
\LWR@blackborderpadding%
]{fbox}{#1}
}
```

xcolor `\let`s things to `\fbox` when it is loaded, and this must remain even for HTML output while in a `lateximage`, so `\fbox` is not modified until `\AtBeginDocument`:

```
\AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` `{`⟨*text*⟩`}` Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```
\newcommand{\fboxBlock}[1]{%
\LWR@forceminwidth{\fboxrule}%
\begin{BlockClass}[%
\LWR@blackborderpadding%
]{fboxBlock}
#1
\end{BlockClass}
}
```

Env `fminipage` `[`⟨*align*⟩`]` `[`⟨*height*⟩`]` `[`⟨*align*⟩`]` `{`⟨*width*⟩`}`

Creates a framed HTML `<div>` around its contents.

A print-output version is also supplied below.

```
\NewDocumentEnvironment{fminipage}{O{t} o O{t} m}
```

```
{%
\LWR@traceinfo{fminipage #1 #2 #3 #4}%
\LWR@forceminwidth{\fboxrule}%
\setlength{\LWR@tempwidth}{#4}%
\IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
\begin{BlockClass}[%
\LWR@blackborderpadding ; %
\uselengthunit{PT}%
\IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
width:\rndprintlength{\LWR@tempwidth}%
]{fminipage}%
}
{%
\end{BlockClass}%
\LWR@traceinfo{fminipage done}%
}
```

\raisebox {⟨*raiselen*⟩} [⟨*height*⟩] [⟨*depth*⟩] {⟨*text*⟩}

```
\LetLtxMacro{\LWR@origraisebox}{\raisebox}

\RenewDocumentCommand{\raisebox}{m o o m}{%
#4%
}
```

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

LWRprint@fminipage is defined inside warpall. For print output, it is \let to fminipage. For HTML output, the HTML version of fminipage is used instead, but the print version is still available for use inside a lateximage.

Env LWRprint@fminipage [⟨*1:align*⟩] [⟨*2:height*⟩] [⟨*3:inner-align*⟩] {⟨*4:width*⟩}

Creates a frame around its contents.

```
\newsavebox{\LWR@fminipagebox}

\NewDocumentEnvironment{LWRprint@fminipage}{O{t} o O{t} m}
{%
```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with \fbox.

If the optional inner alignment is not given, use the outer instead:

```
\IfValueTF{#3}%
{\def\LWR@thisalign{#3}}
{\def\LWR@thisalign{#1}}%
```

Form the outer minipage depending on whether a height was given. Make the outer minipage larger to compensate for the frame.

```
\IfValueTF{#2}%
{\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
{\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
\begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
\IfValueTF{#2}%
{\minipage[#1][#2][\LWR@thisalign]{#4}}%
{\minipage[#1]{#4}}%
}
{%
```

Close the inner minipage and the LR box with the contents:

```
\endminipage%
\end{lrbox}%
```

Create a frame around the contents of the environment:

```
\fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
\endminipage%
}
```

```
\end{warpall}
```

for PRINT output:

```
\begin{warpprint}
```

For print output, the following are `\let` to become active.

`\fboxBlock` {⟨*text*⟩}

Creates a framed HTML `<div>` around the text.

```
\let\fboxBlock\fbox
```

Env `fminipage` [⟨*align*⟩] [⟨*height*⟩] [⟨*align*⟩] {⟨*width*⟩}

Creates a frame around its contents.

```
\LetLtxMacro{\fminipage}{\LWRprint@fminipage}
\LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
\end{warpprint}
```

79 Direct formatting

⚠ `\bfseries`, etc. `\textbf`, etc. are supported, but `\bfseries`, etc. are not yet supported.

⚠ HTML special chars `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

For high-level block and inline custom CSS classes, see section 43.8.

for HTML output:

```
\begin{warpHTML}
```

`\LWR@HTMLtextstyle` {⟨*FormatWP style*⟩} {⟨*class*⟩} {⟨*text*⟩}

If `FormatWP`, adds an explicit style to the text span class. This is used by LibreOffice to mark its imported text using the given style.

```
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
\ifbool{FormatWP}%
{\LWR@htmlspanclass[#1]{#2}{#3}}%
{\LWR@htmlspanclass{#2}{#3}}%
}
```

`\emph` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
\DeclareRobustCommand{\LWR@nullemph}[1]{#1}
\LetLtxMacro{\emph}{\LWR@HTMLemph}
```

`\textmd` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
\LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
}
```

```
\DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}

\LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
\DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
\LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
\LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}

\LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
\LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
}
\DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
\LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
\DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
\LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
\LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextup}[1]{#1}

\LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
\DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
\LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
\LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}

\LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
\LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}

\LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

`\textnormal` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
\DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
\LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

\DeclareRobustCommand{\LWR@nullrmfamily}{}
\DeclareRobustCommand{\LWR@nullsffamily}{}
\DeclareRobustCommand{\LWR@nullttfamily}{}
\DeclareRobustCommand{\LWR@nullbfseries}{}
\DeclareRobustCommand{\LWR@nullmdseries}{}
\DeclareRobustCommand{\LWR@nullupshape}{}
\DeclareRobustCommand{\LWR@nullslshape}{}
\DeclareRobustCommand{\LWR@nullscshape}{}
\DeclareRobustCommand{\LWR@nullitshape}{}
\DeclareRobustCommand{\LWR@nullem}[1]{}
\DeclareRobustCommand{\LWR@nullnormalfont}{}
```

`\LWR@nullfonts` Removes formatting during filename operations.

```
\newcommand*{\LWR@nullfonts}{%
\LetLtxMacro{\emph}{\LWR@nullemph}%
```

```
\LetLtxMacro{\textmd}{\LWR@nulltextmd}%
\LetLtxMacro{\textbf}{\LWR@nulltextbf}%
\LetLtxMacro{\textrm}{\LWR@nulltextrm}%
\LetLtxMacro{\textsf}{\LWR@nulltextsf}%
\LetLtxMacro{\texttt}{\LWR@nulltexttt}%
\LetLtxMacro{\textup}{\LWR@nulltextup}%
\LetLtxMacro{\textit}{\LWR@nulltextit}%
\LetLtxMacro{\textsc}{\LWR@nulltextsc}%
\LetLtxMacro{\textsl}{\LWR@nulltextsl}%
\LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
\LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
\LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
\LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
\LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
\LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
\LetLtxMacro{\upshape}{\LWR@nullupshape}%
\LetLtxMacro{\slshape}{\LWR@nullslshape}%
\LetLtxMacro{\scshape}{\LWR@nullscshape}%
\LetLtxMacro{\itshape}{\LWR@nullitshape}%
\LetLtxMacro{\em}{\LWR@nullem}%
\LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
\renewcommand*{\,}{-}%
\renewcommand*{~}{-}%
\renewcommand*{\textellipsis}{-}%
\renewcommand*{\HTMLunicode}[1]{-}%
\renewcommand*{\HTMLentity}[1]{-}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
\renewcommand*{\&}{and}%
\renewcommand{\textsuperscript}[1]{##1}%
\renewcommand{\textsubscript}[1]{##1}%
\LetLtxMacro\underline\LWR@origunderline%
\RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
\DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
\DeclareRobustCommand{\LWR@subsingledollar}[1]{}%
}
```

\mdseries

```
\renewcommand*{\mdseries}{}
```

\bfseries

```
\renewcommand*{\bfseries}{}
```

\rmfamily

```
\renewcommand*{\rmfamily}{}
```

\sffamily

```
\renewcommand*{\sffamily}{}
```

\ttfamily

```
\renewcommand*{\ttfamily}{}
```

\upshape

```
\renewcommand*{\upshape}{}
```

\itshape

```
\renewcommand*{\itshape}{}
```

\scshape

```
\renewcommand*{\scshape}{}
```

\normalfont

```
\renewcommand*{\normalfont}{}
```

\sp {⟨*text*⟩}

For siunitx. Must work in math mode.

```
\renewcommand{\sp}[1]{\text{<sup>#1</sup>}{}}
```

\sb {⟨*text*⟩}

For siunitx. Must work in math mode.

```
\renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}
```

\textsuperscript {⟨*text*⟩}

```
\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\@textsuperscript {⟨*text*⟩}

```
8134 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\textsubscript {⟨*text*⟩}

```
8135 \AtBeginDocument{
8136 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8137 }
```

\@textsubscript {⟨*text*⟩}

```
8138 \AtBeginDocument{
8139 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8140 }
```

\up {⟨*text*⟩} Prints superscript.

This is \let at the beginning of the document in case some other package has changed the definition.

```
8141 \AtBeginDocument{\let\up\textsuperscript}
```

\fup {⟨*text*⟩} Prints superscript.

Supports fmtcount package.

This is \let at the beginning of the document in case some other package has changed the definition.

```
8142 \AtBeginDocument{\let\fup\textsuperscript}
```

\underline {⟨*text*⟩}

```
8143 \renewcommand{\underline}[1]{%
8144 \LWR@HTMLtextstyle%
8145     {text-decoration:underline;text-decoration-skip}%
8146     {underline}{#1}%
8147 }
```

\hfill

```
8148 \renewcommand*{\hfill}{\qquad}
```

\hrulefill

```
\renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

\dotfill

```
\renewcommand*{\dotfill}{\dots}
```

```
\end{warpHTML}
```

80 Skips, spaces, font sizes

for HTML output:

```
\begin{warpHTML}
```

\, must be redefined after \RequirePackage{printlen}

```
\let\LWR@origcomma\,
\let\LWR@origtilde~
\let\LWR@origenskip\enskip
\let\LWR@origquad\quad
\let\LWR@origqquad\qquad
\let\LWR@orighspace\hspace
\let\LWR@origvspace\vspace
\let\LWR@origrule\rule
\let\LWR@origmedskip\medskip
\let\LWR@origbigskip\bigskip
\let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
\renewrobustcmd*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space

\renewrobustcmd*{~}{\HTMLentity{nbsp}}

\renewrobustcmd*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
\let\LWR@orignormalsize\normalsize
\let\LWR@origsmall\small
\let\LWR@origfootnotesize\footnotesize
\let\LWR@origscriptsize\scriptsize
\let\LWR@origtiny\tiny
```

```
\let\LWR@origlarge\large
\let\LWR@origLarge\Large
\let\LWR@origLARGE\LARGE
\let\LWR@orighuge\huge
\let\LWR@origHuge\Huge
\renewcommand*{\normalsize}{}
\renewcommand*{\small}{}
\renewcommand*{\footnotesize}{}
\renewcommand*{\scriptsize}{}
\renewcommand*{\tiny}{}
\renewcommand*{\large}{}
\renewcommand*{\Large}{}
\renewcommand*{\LARGE}{}
\renewcommand*{\huge}{}
\renewcommand*{\Huge}{}

\renewcommand*{\onecolumn}{}

\renewcommand{\twocolumn}[1][]{

#1

}
```

`\newpage`

```
\renewcommand*{\newpage}{

}
```

`\newline` Uses the HTML `<br />` element.

```
\newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
\let\newline\LWR@newlinebr
```

`\\` Redefined to `\LWR@endofline` or `\LWR@tabularendofline`.

`\LWR@endofline` `* [`⟨*len*⟩`]`

`\\` is assigned to `\LWR@endofline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendofline`.

```
\LetLtxMacro\LWR@origendofline\\
\NewDocumentCommand{\LWR@endofline}{s o}
{%
\newline%
}
```

\LWR@minipagestartpars Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a to be inside a p, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, lwarp correctly surpresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. lwarp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are surpressed around \hspace, \enskip, \quad, and \qquad until the end of the paragraph, when the closing p tag is created.

\hspace
\enskip
\quad
\qquad

When a minipage is seen, the boolean LWR@minipagethispar is set, telling the following horizontal whitespace commands to try to surpress their surrounding paragraph tags. LWR@minipagethispar is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before \hspace, \quad, or \qquad's HTML output.

```
\newcommand*{\LWR@minipagestartpars}{%
\ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
}
```

\LWR@minipagestoppars Placed just after \hspace, \quad, or \qquad's HTML output.

```
\newcommand*{\LWR@minipagestoppars}{%
\ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
}
```

\quad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\quad}{%
\LWR@minipagestoppars%
\HTMLunicode{2001}%
\LWR@minipagestartpars%
}
```

\qquad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
\renewcommand*{\enskip}{%
\LWR@minipagestoppars%
\HTMLunicode{2000}%
```

```
\LWR@minipagestartpars%
}
```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

Len \WR@tempheight

Len \WR@tempraise

```
\newlength{\LWR@tempwidth}
\newlength{\LWR@tempheight}
\newlength{\LWR@tempraise}
```

\LWR@hspace * {⟨*length*⟩}

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

\hspace{\fill} is converted to \hspace{2em}, equal to \qquad.

```
\NewDocumentCommand{\LWR@hspace}{s m}{%
\setlength{\LWR@tempwidth}{#2}%
```

If \fill, change to \qquad:

```
\ifnum\gluestretchorder\LWR@tempwidth>0%
\setlength{\LWR@tempwidth}{2em}%
\fi%
```

Only if the width is not zero:

```
\ifdimcomp{\LWR@tempwidth}{=}{0pt}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
    \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
    {%
        \HTMLunicode{2009}% thin breakable space
    }%
```

Print the span with the converted width. Not rounded.

```
    {%
        \uselengthunit{PT}%
        \LWR@htmltagc{%
```

```
            span style="width:\printlength{\LWR@tempwidth}; %
                display:inline-block"%
        }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
        \ifbool{FormatWP}{%
            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
                \quad%
                \addtolength{\LWR@templengthone}{-1em}%
            }%
        }{}%
```

Close the span:

```
        \LWR@htmltagc{/span}%
    }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% width not 0
}
```

`\LWR@nohspace` * {⟨*length*⟩}

Used to disable \hspace while creating description \items.

```
\NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` * {⟨*length*⟩}

Handles special minipage & horizontal space interactions.

```
\LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * {⟨*length*⟩} Nullified vspace.

```
\NewDocumentCommand{\LWR@vspace}{s m}{}
```

`\vspace` * {⟨*length*⟩} Nullified.

```
\let\vspace\LWR@vspace
```

`\linebreak` `[`⟨*num*⟩`]` Inserts an HTML `br` tag.

```
\renewcommand*{\linebreak}[1][]{\newline}
```

`\nolinebreak` `[`⟨*num*⟩`]`

```
\renewcommand*{\nolinebreak}[1][]{}
```

`\pagebreak` `[`⟨*num*⟩`]` Starts a new paragraph.

```
\renewcommand*{\pagebreak}[1][]{

}
```

`\nopagebreak` `[`⟨*num*⟩`]`

```
\renewcommand*{\nopagebreak}[1][]{}
```

`\enlargethispage` `*` `{`⟨*len*⟩`}`

```
\RenewDocumentCommand{\enlargethispage}{s m}{}
```

`\clearpage`
`\cleardoublepage`

```
\renewcommand*{\clearpage}{}
\renewcommand*{\cleardoublepage}{}
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
\newcommand*{\LWR@currenttextcolor}{black}
```

`\LWR@rule` `[`⟨*raise*⟩`]` `{`⟨*width*⟩`}` `{`⟨*height*⟩`}`

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
\NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary LaTeX length, from which comparisons and conversions may be made:

```
\setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
\ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
{}% zero- width
{% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
    \ifthenelse{%
        \lengthtest{\LWR@tempwidth>0pt}\AND%
        \lengthtest{\LWR@tempwidth<1pt}%
    }%
    {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
    \setlength{\LWR@tempheight}{#3}%
    \ifthenelse{%
        \lengthtest{\LWR@tempheight>0pt}\AND%
        \lengthtest{\LWR@tempheight<1pt}%
    }%
    {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in LaTeX code.

```
    \uselengthunit{PT}%
    \LWR@htmltagc{%
    span
    style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
    \ifbool{FormatWP}{}{background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
    width:\printlength{\LWR@tempwidth} ; %
    height:\printlength{\LWR@tempheight} ; %
```

The raise height is converted to a CSS transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a LaTeX length allows a typical LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of LaTeX ex height, then print that to HTML with an ex unit.

```
    \IfValueT{#1}%
    {%
        \setlength{\LWR@tempraise}{0pt-#1}%
        \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
        \LWR@orignewline%
        -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
        \LWR@orignewline%
        -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
        \LWR@orignewline%
        transform: translate(0pt,\printlength{\LWR@tempraise}); %
        \LWR@orignewline%
    }%
```

Display inline-block to place the span inline with the text:

```
    display:inline-block;"%
    }%
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```
        \ifbool{FormatWP}{%
            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
                \_{}%
                \addtolength{\LWR@templengthone}{-1em}%
            }%
        }{}%
```

Close the span:

```
    \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% non-zero width
}
```

`\rule` `[`⟨*raise*⟩`]` `{`⟨*width*⟩`}` `{`⟨*height*⟩`}`

Handles special minipage & horizontal space interactions.

```
\renewcommand{\rule}{\LWR@rule}
```

```
\end{warpHTML}
```

81 \phantomsection

for HTML output:
```
\begin{warpHTML}
```

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
\DeclareDocumentCommand{\phantomsection}{}{%
\section*{}%
}
```

```
\end{warpHTML}
```

82 \LaTeX and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally `\let` in `\LWR@LwarpStart`.

For CSS conversions, see:
http://edward.oconnor.cx/2007/08/tex-poshlet
http://nitens.org/taraborelli/texlogo

82.1 HTML logos

for HTML output:
```
\begin{warpHTML}
```

`\TeX` TEX

`latexlogo` is a CSS class used to properly typeset the E and A in LATEX and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in LaTeX, LuaTeX, ConTeXt, etc.

```
\let\LWR@origTeX\TeX

\newcommand*{\LWR@TeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
            {latexlogo}%
        {T\textsubscript{e}X}%
    }%
}
```

`\LaTeX` `\LaTeXe` LaTeX, LaTeX 2_ε

```
\let\LWR@origLaTeX\LaTeX

\newcommand*{\LWR@LaTeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
            {latexlogo}%
        {L\textsuperscript{a}T\textsubscript{e}X}%
    }%
}

\let\LWR@origLaTeXe\LaTeXe

\renewcommand*{\LaTeXe}
{\LaTeX\InlineClass{latexlogofont}%
{\,2\textsubscript{\textit{\HTMLunicode{3B5}}}}}
```

`\LuaTeX` `\LuaLaTeX` LuaTeX, LuaLaTeX

```
\newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
\newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` `\XeLaTeX` XeTeX, XeLaTeX

`xetexlogo` is a CSS class which aligns the backwards E in XeTeX and spaces TeX appropriately.

xelatexlogo is a css class which aligns the backwards E in XeLaTeX and spaces LaTeX appropriately.

```
\newcommand*{\Xe}
  {X\textsubscript{\HTMLunicode{18e}}}
\newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
\newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

`\ConTeXt` ConTeXt

```
\newcommand*{\LWR@ConTeXt}
{\InlineClass{latexlogofont}{Con}\TeX{}%
\InlineClass{latexlogofont}{t}}
```

`\BibTeX` BibTeX, *MakeIndex*
`\MakeIndex`

```
\providecommand*{\BibTeX}
{\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}

\newcommand*{\MakeIndex}
{\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

`\AmS` $\mathcal{AMS}$

amslogo is a css class used for the $\mathcal{AMS}$logo.

```
\AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
{\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX` MiKTeX

```
\newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` LyX

lyxlogo is a css class used for the LyXlogo.

```
\newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
\end{warpHTML}
```

82.2 Print logos

for PRINT output:

```
\begin{warpprint}
\newcommand*{\XeTeXrevE}
  {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
\providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
\providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
\providecommand*{\AmS}{%
\leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
\hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
\newcommand*{\LyX}{\textsf{LyX}}
\providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
\providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
\providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
\providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
\providecommand*{\ConTeXt}{\mbox{Con\TeX{}t}}
\providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
\end{warpprint}
```

83 \AtBeginDocument, \AtEndDocument

for HTML output:

```
\begin{warpHTML}
```

\LWR@LwarpStart
\LWR@LwarpEnd

Automatically sets up the HTML-related actions for the start and end of the document.

```
\AfterEndPreamble{\LWR@LwarpStart}
\AtEndDocument{\LWR@LwarpEnd}
```

```
\end{warpHTML}
```

84 Koma-script

Load patches to koma-script.

for HTML output:

```
\begin{warpHTML}
```

```
\@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}
```

```
\end{warpHTML}
```

85 Memoir

Load patches to memoir.

for HTML output:

```
\begin{warpHTML}
\@ifclassloaded{memoir}{\RequirePackage{lwarp-patch-memoir}}{}
\end{warpHTML}
```

86 Trademarks

- TEX is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

The following adjustments apply to the lwarp-* package listings:

File 2 `lwarp-a4.sty`

§ 87 Package a4

Pkg `a4` a4 is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a4}
\newcommand*{\WideMargins}{}
```

File 3 `lwarp-a4wide.sty`

§ 88 Package a4wide

Pkg `a4wide` a4wide is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a4wide}
```

File 4 `lwarp-a5comb.sty`

§ 89 Package a5comb

Pkg `a5comb` a5comb is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a5comb}
```

File 5 `lwarp-abstract.sty`

§ 90 Package abstract

(*Emulates or patches code by* Peter Wilson.)

Pkg `abstract` abstract is supported and patched by lwarp.

⚠ missing TOC If using the `number` option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output:

memoir provides an `abstract` environment even though it is not an article or report class. Meanwhile, lwarp loads book to emulate memoir, but book does not have an `abstract` environment, so when the abstract package is loaded for emulation there is no pre-existing `abstract` to redefine, which would cause an error. Thus, a null `abstract` is provide here:

```
\ProvideDocumentEnvironment{abstract}{}{}{}
```

Accept all options for lwarp-abstract:

```
\LWR@ProvidesPackagePass{abstract}

\AtBeginDocument{
\BeforeBeginEnvironment{abstract}{
\LWR@forcenewpage
\BlockClass{abstract}
}
\AfterEndEnvironment{abstract}{\endBlockClass}
}

\renewcommand{\@bsrunintitle}{%
\hspace*{\abstitleskip}%
{\abstractnamefont%
\InlineClass{abstractrunintitle}{\abstractname}%
\@bslabeldelim}%
}

\@ifclassloaded{memoir}
{
  \renewenvironment{abstract}{%
%     \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
          \begin{\absnamepos}%
  \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
            \@endparpenalty\@M
          \end\absnamepos%
%%        \vspace{\abstitleskip}%
```

```
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
    }
}{% not memoir
\if@titlepage
  \renewenvironment{abstract}{%
%     \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
          \begin{\absnamepos}%
  \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
            \@endparpenalty\@M
          \end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
    }
\else
  \renewenvironment{abstract}{%
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
\begin{\absnamepos}%
\abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
\end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
```

```
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}}
\fi
}% not memoir
```

File 6 `lwarp-adjmulticol.sty`

§ 91 Package adjmulticol

(*Emulates or patches code by* Boris Veytsman.)

Pkg `adjmulticol` adjmulticol is emulated.

Emulation similar to multicols is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, lwarp allows a variable number of columns up to three.

for HTML output:

```
\LWR@ProvidesPackageDrop{adjmulticol}

\RequirePackage{multicol}
```

`adjmulticols` * {⟨*numcols*⟩} {⟨*left margi*⟩} {⟨*right margin*⟩}

```
\NewDocumentEnvironment{adjmulticols}{s m m m}
{%
```

Compute the margins, and limit to positive only:

```
\setlength{\LWR@templengthone}{#3}%
\ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
\setlength{\LWR@templengthtwo}{#4}
\ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%
```

If one column is specified, use a of class `singlecolumn`, else use `multicols`:

```
\newcommand*{\LWR@mcolstype}{multicols}%
\ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{singlecolumn}}{}%
```

Help avoid page overflow:

```
\LWR@forcenewpage%
```

Print margins in points:

```
\uselengthunit{PT}%
```

Create the with the given margin and class:

```
\BlockClass[%
margin-left:\rndprintlength{\LWR@templengthone} ; %
margin-right:\rndprintlength{\LWR@templengthtwo}%
]{\LWR@mcolstype}%
}
{\endBlockClass}
```

File 7 `lwarp-addlines.sty`

§ 92 Package addlines

(*Emulates or patches code by* Will Robertson.)

Pkg `addlines` addlines is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{addlines}
```

```
\newcommand*\addlines[1][1]{}
\let\addline\addlines
\newcommand*\removelines[1][1]{}
\let\removeline\removelines
```

File 8 `lwarp-afterpage.sty`

§ 93 Package afterpage

(*Emulates or patches code by* David Carlisle.)

Pkg `afterpage` Emulated.

for HTML output: Discard all options for lwarp-afterpage:

```
\LWR@ProvidesPackageDrop{afterpage}
```

```
\newcommand{\afterpage}[1]{#1}
```

File 9 `lwarp-algorithmicx.sty`

§ 94 Package

algorithmicx

(*Emulates or patches code by* Szász János.)

Pkg `algorithmicx` algorithmicx is supported with minor adjustments.

for HTML output:

```
\LWR@ProvidesPackagePass{algorithmicx}
```

Inside the `algorithmic` environment, level indenting is converted to a `<span>` of the required length, and comments are placed inside a `<span>` which is floated right.

⚠ package conflicts If using `\newfloat`, trivfloat, and/or algorithmicx together, see section 285.1.

for HTML output:

```
\begin{warpHTML}

\AtBeginEnvironment{algorithmic}{%
%
\let\origALG@doentity\ALG@doentity%
%
\renewcommand*{\ALG@doentity}{%
\origALG@doentity%
\uselengthunit{PT}%
\LWR@htmltagc{%
span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
}%
\ifbool{FormatWP}{%
\setlength{\LWR@templengthone}{\the\ALG@thistlm}%
\whiledo{\lengthtest{\LWR@templengthone>1em}}{%
\quad%
\addtolength{\LWR@templengthone}{-1em}%
}%
}{}%
\LWR@htmltagc{/span}%
}%

\let\LWR@origComment\Comment%

\renewcommand{\Comment}[1]{%
    \InlineClass{floatright}{\LWR@origComment{#1}}%
}%
}

\renewcommand\algorithmiccomment[1]{%
\hfill\HTMLunicode{25B7} #1% white right triangle
```

```
}%
\end{warpHTML}
```

File 10 `lwarp-alltt.sty`

§ 95 Package alltt

(*Emulates or patches code by* Johannes Braams.)

Pkg `alltt` alltt is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{alltt}

\AfterEndPreamble{
\LWR@traceinfo{Patching alltt.}
\AtBeginEnvironment{alltt}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
}
```

File 11 `lwarp-amsthm.sty`

§ 96 Package amsthm

(*Emulates or patches code by* Publications Technical Group — American Mathematical Society.)

The original source code is located in `amsclass.dtx`, and printed in `amsclass.pdf`.

Pkg `amsthm` amsthm is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{amsthm}
```

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\renewcommand{\theoremstyle}[1]{%
  \@ifundefined{th@#1}{%
```

Table 12: AMSthm package — CSS styling of theorems and proofs

Theorem: of class amsthmbody<theoremstyle>

Theorem Name: of class amsthmname<theoremtyle>

Theorem Number: of class amsthmnumber<theoremstyle>

Theorem Note: of class amsthmnote<theoremstyle>

Proof: of class amsthmproof

Proof Name: of class amsthmproofname

where <theoremstyle> is plain, definition, etc.

```
    \PackageWarning{amsthm}{Unknown theoremstyle `#1'}%
    \thm@style{plain}%
    \renewcommand{\LWR@newtheoremstyle}{plain}% new
  }{%
    \thm@style{#1}%
    \renewcommand{\LWR@newtheoremstyle}{#1}% new
  }%
}
```

Patched to remember the style for this theorem type:

```
\def\@xnthm#1#2{%
  \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
  \let\@tempa\relax
  \@xp\@ifdefinable\csname #2\endcsname{%
    \global\@xp\let\csname end#2\endcsname\@endtheorem
    \ifx *#1% unnumbered, need to get one more mandatory arg
      \edef\@tempa##1{%
        \gdef\@xp\@nx\csname#2\endcsname{%
          \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
            {}{##1}}}%
    \else % numbered theorem, need to check for optional arg
      \def\@tempa{\@oparg{\@ynthm{#2}}[]}%
    \fi
    \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
  }%
  \@tempa
}
```

Patched to enclose with css:

```
\newcommand{\LWR@haveamsthmname}{
```

```
\renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
}

\newcommand{\LWR@haveamsthmnumber}{
\renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
}

\newcommand{\LWR@haveamsthmnote}{
\renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
}

\LWR@haveamsthmname
\LWR@haveamsthmnumber
\LWR@haveamsthmnote
```

Patches for css:

```
\def\@begintheorem#1#2[#3]{%
    \item[
%   \deferred@thm@head{
%     \the\thm@headfont \thm@indent
    \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
    \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
    \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
    \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
    \the\thm@headpunct~
    \thmheadnl % possibly a newline.
    \hskip\thm@headsep
%   }%
    ]
  \ignorespaces}
```

Patched for css:

```
\def\@thm#1#2#3{%
  \ifhmode\unskip\unskip\par\fi
  \normalfont
  \LWR@forcenewpage% new
  \BlockClass{amsthmbody\LWR@thisthmstyle}% new
  \trivlist
  \let\thmheadnl\relax
  \let\thm@swap\@gobble
  \thm@notefont{\fontseries\mddefault\upshape}%
  \thm@headpunct{.}% add period after heading
  \thm@headsep 5\p@ plus\p@ minus\p@\relax
  \thm@space@setup
  #1% style overrides
  \@topsep \thm@preskip               % used by thm head
  \@topsepadd \thm@postskip           % used by \@endparenv
```

```
  \def\@tempa{#2}\ifx\@empty\@tempa
    \def\@tempa{\@oparg{\@begintheorem{#3}{}}[]}%
  \else
    \refstepcounter{#2}%
    \def\@tempa{\@oparg{\@begintheorem{#3}{\csname the#2\endcsname}}[]}%
  \fi
  \@tempa
}
```

cleveref patches `\@thm` to do `\cref@thmoptarg` if an optional argument is given. lwarp then patches `\cref@thmoptarg` `\AtBeginDocument`.

```
\AtBeginDocument{
\def\cref@thmoptarg[#1]#2#3#4{%
    \ifhmode\unskip\unskip\par\fi%
    \normalfont%
    \LWR@forcenewpage% new
    \BlockClass{amsthmbody\LWR@thisthmstyle}% new
    \trivlist%
    \let\thmheadnl\relax%
    \let\thm@swap\@gobble%
    \thm@notefont{\fontseries\mddefault\upshape}%
    \thm@headpunct{.}% add period after heading
    \thm@headsep 5\p@ plus\p@ minus\p@\relax%
    \thm@space@setup%
    #2% style overrides
    \@topsep \thm@preskip               % used by thm head
    \@topsepadd \thm@postskip           % used by \@endparenv
    \def\@tempa{#3}\ifx\@empty\@tempa%
        \def\@tempa{\@oparg{\@begintheorem{#4}{}}[]}%
    \else%
        \refstepcounter[#1]{#3}%  <<< cleveref modification
        \def\@tempa{\@oparg{\@begintheorem{#4}{\csname the#3\endcsname}}[]}%
    \fi%
    \@tempa
}%
}% AtBeginDocument

\def\@endtheorem{\endtrivlist\endBlockClass\@endpefalse }
```

Proof QED symbol:

```
\AtBeginDocument{
\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
}
```

Patched for CSS:

```
\renewenvironment{proof}[1][\proofname]{\par
\LWR@forcenewpage% new
    \BlockClass{amsthmproof}% new
  \pushQED{\qed}%
  \normalfont \topsep6\p@\@plus6\p@\relax
  \trivlist
  \item[
        \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
}{%
  \InlineClass{theoremendmark}{\popQED}\endtrivlist%
  \endBlockClass% new
  \@endpefalse
}
```

File 12 `lwarp-anonchap.sty`

§ 97 Package anonchap

(*Emulates or patches code by* Peter Wilson.)

Pkg anonchap

anonchap is emulated.

⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

The code is shared by tocbibind.

for HTML output:

```
\LWR@ProvidesPackageDrop{anonchap}

\newcommand{\simplechapter}[1][\@empty]{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\newcommand{\restorechapter}{%
\let\@chapcntformat\@seccntformat%
}
```

File 13 `lwarp-anysize.sty`

§ 98 Package anysize

(*Emulates or patches code by* Michael Salzenberg, Thomas Esser.)

Pkg `anysize` anysize is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{anysize}

\def\papersize#1#2{}
\def\marginsize#1#2#3#4{}
```

File 14 `lwarp-appendix.sty`

§ 99 Package appendix

(*Emulates or patches code by* Peter Wilson.)

Pkg `appendix` appendix is patched for use by lwarp.

⚠ incorrect TOC link During HTML conversion, the option `toc` without the option `page` results in a TOC link to whichever section was before the `appendices` environment. It is recommended to use both `toc` and also `page` at the same time.

for HTML output:

```
\LWR@ProvidesPackagePass{appendix}

\renewcommand*{\@chap@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}

\renewcommand*{\@sec@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}
```

File 15 `lwarp-arabicfront.sty`

§ 100 Package arabicfront

Pkg `arabicfront` arabicfront is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{arabicfront}
```

File 16 `lwarp-array.sty`

§ 101 Package array

Pkg `array` array is used as-is for print output, and emulated for HTML.

for HTML output:

```
\LWR@ProvidesPackagePass{array}

\let\LWR@origfirsthline\firsthline
\let\LWR@origlasthline\lasthline

\appto{\LWR@restoreorigformatting}{%
\let\firsthline\LWR@origfirsthline%
\let\lasthline\LWR@origlasthline%
}

\renewcommand*{\firsthline}{\LWR@hline}%
\renewcommand*{\lasthline}{\LWR@hline}%
```

File 17 `lwarp-atbegshi.sty`

§ 102 Package atbegshi

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `atbegshi` Emulated.

for HTML output: Discard all options for lwarp-atbegshi:

```
\LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]
```

```
\newcommand*{\AtBeginShipout}[1]{}
\newbox\AtBeginShipoutBox
\newcommand*{\AtBeginShipoutNext}[1]{}
\newcommand*{\AtBeginShipoutFirst}[1]{}
\newcommand*{\AtBeginShipoutDiscard}{}
\newcommand*{\AtBeginShipoutInit}{}
\newcommand*{\AtBeginShipoutAddToBox}[1]{}
\newcommand*{\AtBeginShipoutAddToBoxForeground}[1]{}
\newcommand*{\AtBeginShipoutUpperLeft}[1]{}
\newcommand*{\AtBeginShipoutUpperLeftForeground}[1]{}
\newcommand*{\AtBeginShipoutOriginalShipout}[1]{}
\def\AtBeginShipoutBoxWidth{0pt}
\def\AtBeginShipoutBoxHeight{0pt}
\def\AtBeginShipoutBoxDepth{0pt}

```

File 18 **lwarp-authblk.sty**

§ 103 Package

authblk

(*Emulates or patches code by* Patrick W. Daly.)

Pkg authblk

authblk is patched for HTML.

package support

⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

\published and \subtitle

If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 56.7.

(*Emulates or patches code by* Patrick W. Daly.)

for HTML output:

Require that authblk be loaded before titling:

```
\@ifpackageloaded{titling}{
\PackageError{lwarp-authblk}
{Package authblk must be loaded before titling}
{Titling appends authblk's author macro, so authblk must be loaded first.}
}
{}
```

Load authblk:

```
\LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```
\LetLtxMacro\LWRAB@affil\affil

\renewcommand{\affil}[2][]{%
\LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
}
```

Create an HTML break for an `\authorcr`:

```
\renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 19 **`lwarp-backref.sty`**

§ 104 Package backref

(*Emulates or patches code by* David Carlisle and Sebastian Rahtz.)

Pkg `backref` backref is patched for use by lwarp.

⚠ loading Note that backref must be explicitly loaded, and is not automatically loaded by hyperref when generating HTML output.

for HTML output:

```
\LWR@ProvidesPackagePass{backref}
```

Force the `hyperref` option:

```
\def\backref{}\let\backrefxxx\hyper@section@backref
```

File 20 **`lwarp-balance.sty`**

§ 105 Package balance

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `balance` Emulated.

for HTML output: Discard all options for lwarp-balance:

```
\LWR@ProvidesPackageDrop{balance}

\newcommand*{\balance}{}
\newcommand*{\nobalance}{}
```

File 21 `lwarp-bigdelim.sty`

§ 106 Package

bigdelim

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `bigdelim` bigdelim is used as-is for print or `lateximage`, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `<span>` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`.

⚠ use `\mrowcell` `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

	a	b
left {	c	d
	e	f
	g	h

for HTML output: First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case bigdelim was not loaded:

```
\let\ldelim\relax
\let\rdelim\relax
```

Next, load the package's new definitions:

```
\LWR@ProvidesPackagePass{bigdelim}
```

Remember the print-mode versions:

```
\LetLtxMacro\LWR@origldelim\ldelim
\LetLtxMacro\LWR@origrdelim\rdelim
```

`\ldelim` {⟨*1:delimiter*⟩} {⟨*2:#rows*⟩} {⟨*3:width*⟩} [⟨*4:text*⟩]
`\rdelim`

```
\RenewDocumentCommand{\ldelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{right}%
\multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
}

\RenewDocumentCommand{\rdelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{left}%
\multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
}
```

When entering a `lateximage`, restore the print-mode versions:

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro{\ldelim}{\LWR@origldelim}%
\LetLtxMacro{\rdelim}{\LWR@origrdelim}%
}
```

File 22 `lwarp-bigstrut.sty`

§ 107 Package bigstrut

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `bigstrut` bigstrut is used as-is for print or `lateximage`, and patched for HTML.

for HTML output:

```
\LWR@ProvidesPackagePass{bigstrut}

\LetLtxMacro\LWR@origbigstrut\bigstrut

\renewcommand\bigstrut[1][x]{}

\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
}
```

File 23 `lwarp-bookmark.sty`

§ 108 Package bookmark

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `bookmark` bookmark is emulated.

for HTML output: Discard all options for lwarp-bookmark:

```
\LWR@ProvidesPackageDrop{bookmark}

\newcommand*{\bookmarksetup}[1]{}
\newcommand*{\bookmarksetupnext}[1]{}
\newcommand*{\bookmark}[2][]{}
\newcommand*{\bookmarkdefinestyle}[2]{}
\newcommand*{\bookmarkget}[1]{}
\newcommand{\BookmarkAtEnd}[1]{}
```

File 24 `lwarp-booktabs.sty`

§ 109 Package booktabs

(*Emulates or patches code by* Simon Fear.)

Pkg `booktabs` booktabs is emulated during HTML output, and used as-is during print output and inside an HTML `lateximage`.

for HTML output:

```
\LWR@ProvidesPackagePass{booktabs}
```

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML `lateximage`s. The HTML versions temporarily overwrite these print versions when `tabular` is started.

```
\LetLtxMacro\LWR@origtoprule\toprule
\LetLtxMacro\LWR@origmidrule\midrule
\LetLtxMacro\LWR@origcmidrule\cmidrule
\LetLtxMacro\LWR@origbottomrule\bottomrule
\LetLtxMacro\LWR@origaddlinespace\addlinespace
\LetLtxMacro\LWR@origmorecmidrules\morecmidrules
\LetLtxMacro\LWR@origspecialrule\specialrule
```

File 25 `lwarp-boxedminipage.sty`

§ 110 Package boxedminipage

Pkg `boxedminipage` boxedminipage is superceded by boxedminipage2e.

for HTML output:

```
\LWR@loadnever{boxedminipage}{boxedminipage2e}
```

File 26 `lwarp-boxedminipage2e.sty`

§ 111 Package boxedminipage2e

(*Emulates or patches code by* Scott Pakin.)

Pkg `boxedminipage2e` boxedminipage2e is emulated.

for HTML output: Discard all options for lwarp-boxedminipage2e:

```
\LWR@ProvidesPackageDrop{boxedminipage2e}

\newenvironment{boxedminipage}{%
\begin{BlockClass}{framebox}%
\minipage%
}
{
\endminipage%
\end{BlockClass}
}
```

File 27 `lwarp-breakurl.sty`

§ 112 Package breakurl

(*Emulates or patches code by* Vilar Camara Neto.)

Pkg `breakurl` breakurl is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{breakurl}

\LetLtxMacro\burl\url

\NewDocumentCommand{\burlalt}{O{} +m m}{%
\LWR@ensuredoingapar%
\def\LWR@templink{#2}%
\@onelevel@sanitize\LWR@templink%
\def\LWR@templinktwo{#3}%
\@onelevel@sanitize\LWR@templinktwo%
\LWR@subhyperref{\LWR@templink}{\LWR@templinktwo}%
\LWR@ensuredoingapar%
}

```

```
\LetLtxMacro\urlalt\burlalt
```

File 28 `lwarp-cancel.sty`

§ 113 Package

cancel

Pkg `cancel` cancel is used as-is for SVG math, and emulated for HTML text output.

for HTML output:

```
\LWR@ProvidesPackagePass{cancel}
```

`\cancelto` is math-only, so is used as-is.

```
\LetLtxMacro\LWR@origcancel\cancel
\LetLtxMacro\LWR@origbcancel\bcancel
\LetLtxMacro\LWR@origxcancel\xcancel

\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\cancel\LWR@origcancel%
\LetLtxMacro\bcancel\LWR@origbcancel%
\LetLtxMacro\xcancel\LWR@origxcancel%
}
```

`\LWR@cancelcolor` {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
\newcommand{\LWR@cancelcolor}[5]{%
\ifcsempty{#2}%
{\LWR@HTMLtextstyle{#5}{#3}{#1}}%
{\LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}}%
}
```

`\cancel` {⟨*text*⟩}

```
\DeclareRobustCommand{\cancel}[1]{%
\begingroup%
\CancelColor%
\LWR@findcurrenttextcolor%
\color{black}%
\LWR@cancelcolor{#1}{LWR@tempcolor}{sout}{text-decoration-color}%
    {text-decoration:line-through}%
\endgroup%
}

\LetLtxMacro\bcancel\cancel
\LetLtxMacro\xcancel\cancel
```

File 29 `lwarp-caption2.sty`

§ 114 Package caption2

Pkg `caption2` caption2 is not used. The user is recommended to use caption instead.

for HTML output:

```
\LWR@loadnever{caption2}{caption}
```

File 30 `lwarp-ccaption.sty`

§ 115 Package ccaption

Pkg `ccaption` ccaption is not used. The user is recommended to use caption instead.

for HTML output:

```
\LWR@loadnever{ccaption}{caption}
```

File 31 `lwarp-changebar.sty`

§ 116 Package changebar

Pkg `changebar` changebar is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{changebar}

\newcommand*{\cbstart}{}
\newcommand*{\cbend}{}
\newenvironment*{\changebar}{}{}
\newcommand*{\cbdelete}{}
\newcommand*{\nochnagebars}{}
\newcommand*{\cbcolor}[1]{}
\newlength{\changebarwidth}
\newlength{\deletebarwidth}
\newlength{\changebarsep}
\newcounter{changebargrey}
```

File 32 `lwarp-changepage.sty`

§ 117 Package changepage

(*Emulates or patches code by* Peter Wilson.)

Pkg `changepage` changepage is emulated.

for HTML output: Discard all options for lwarp-changepage:

```
\LWR@ProvidesPackageDrop{changepage}

\newif\ifoddpage
\DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
\DeclareRobustCommand{\changetext}[5]{}
\DeclareRobustCommand{\changepage}[9]{}

\@ifundefined{adjustwidth}{
\newenvironment{adjustwidth}[2]{}{}
\newenvironment{adjustwidth*}[2]{}{}
}{
\renewenvironment{adjustwidth}[2]{}{}
\renewenvironment{adjustwidth*}[2]{}{}
}

\DeclareDocumentCommand{\strictpagecheck}{}
\DeclareDocumentCommand{\easypagecheck}{}
```

File 33 `lwarp-chngpage.sty`

§ 118 Package chngpage

Pkg `chngpage` chngpage is superceded by changepage.

for HTML output:

```
\LWR@loadnever{chngpage}{changepage}
```

File 34 `lwarp-chappg.sty`

§ 119 Package chappg

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `chappg` chappg is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{chappg}

\renewcommand{\pagenumbering}[2][]{}
\providecommand{\chappgsep}{--}
```

File 35 `lwarp-cite.sty`

§ 120 Package cite

(*Emulates or patches code by* Donald Arseneau.)

Pkg `cite` cite is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{cite}
```

For the [super] option, the `\kern` must be removed:

```
\def\LWRCT@biblabel#1{\@citess{#1}\kern-\labelsep\,}

\ifdefstrequal{\@biblabel}{\LWRCT@biblabel}
{
    \def\@biblabel#1{\@citess{#1}}
}{}
```

For the [super] option, `\textsuperscript` is used instead of math superscript:

```
\def\@citess#1{\textsuperscript{#1}}

\DeclareDocumentCommand\citepunct{}{,\,\relax}
```

File 36 `lwarp-color.sty`

§ 121 Package color

Pkg `color` Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

⚠ missing colors It should be sufficient for the user's document to load color then load xcolor as well.

for HTML output:

```
\LWR@ProvidesPackagePass{color}
\RequirePackage{xcolor}
```

File 37 `lwarp-continue.sty`

§ 122 Package continue

Pkg `continue` continue is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{continue}

\newcommand*{\flagcont}{}
\newcommand*{\flagend}{}
\newcommand*{\flagword}{}
\newcommand*{\preflagword}{}
\newcommand*{\postflagword}{}
\newlength\contsep
\newlength\contdrop
```

File 38 `lwarp-crop.sty`

§ 123 Package crop

(*Emulates or patches code by* Melchior FRANZ.)

Pkg `crop` Emulated.

for HTML output: Discard all options for lwarp-crop:

```
\LWR@ProvidesPackageDrop{crop}
```

```
\newcommand*{\crop}[1][]{}
\newcommand*{\cropdef}[6][]{}
```

File 39 `lwarp-cuted.sty`

§ 124 Package cuted

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `cuted` cuted is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{cuted}
```

```
\newenvironment{strip}{}{}
\newskip\stripsep
\def\oldcolsbreak#1{}
```

File 40 `lwarp-cutwin.sty`

§ 125 Package cutwin

(*Emulates or patches code by* Peter Wilson and Alan Hoenig.)

Pkg `cutwin` Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```
\LWR@ProvidesPackageDrop{cutwin}
```

```
\newcommand*{\opencutleft}{}
\newcommand*{\opencutright}{}
\newcommand*{\opencutcenter}{}
\newcommand*{\cutfuzz}{}

\newenvironment{cutout}[4]
{\marginpar{\windowpagestuff}}
{}

\newcommand*{\windowpagestuff}{}

\newcommand*{\pageinwindow}{%
% \begin{minipage}{.3\linewidth}
\windowpagestuff
% \end{minipage}
```

```
}

\newenvironment{shapedcutout}[3]
{\marginpar{\picinwindow}}
{}

\newcommand*{\putstuffinpic}{}

\newcommand*{\picinwindow}{%
\begin{picture}(0,0)
\putstuffinpic
\end{picture}}
```

File 41 `lwarp-dblfnote.sty`

§ 126 Package dblfnote

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `dblfnote` dblfnote is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{dblfnote}

\newcounter{DFNsloppiness}
\newdimen\DFNcolumnsep
\newdimen\DFNcolumnwidth
\def\DFNallowcbreak{}
\def\DFNinhibitcbreak{}
\def\DFNtrysingle{}
\def\DFNalwaysdouble{}
\def\DFNruleboth{}
\def\DFNruleleft{}
```

File 42 `lwarp-dcolumn.sty`

§ 127 Package dcolumn

Pkg `dcolumn` dcolumn is emulated by the lwarp core.

```
\LWR@ProvidesPackageDrop{dcolumn}
```

File 43 `lwarp-draftwatermark.sty`

§ 128 Package draftwatermark

(*Emulates or patches code by* Sergio Callegari.)

Pkg `draftwatermark` draftwatermark is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{draftwatermark}

\newcommand{\SetWatermarkAngle}[1]{}
\newcommand{\SetWatermarkColor}[1]{}
\newcommand{\SetWatermarkLightness}[1]{}
\newcommand{\SetWatermarkFontSize}[1]{}
\newcommand{\SetWatermarkScale}[1]{}
\newcommand{\SetWatermarkHorCenter}[1]{}
\newcommand{\SetWatermarkVertCenter}[1]{}
\newcommand{\SetWatermarkText}[1]{}
```

File 44 `lwarp-easy-todo.sty`

§ 129 Package easy-todo

(*Emulates or patches code by* Juan Rada-Vilela.)

Pkg `easy-todo` easy-todo is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{easy-todo}
```

`\listoftodos` Modified to correct buggy use of `\flushright`.

```
\let\LWR@origlistoftodos\listoftodos

\renewcommand{\listoftodos}{%
\begingroup
\renewcommand{\flushright}{}
\LWR@origlistoftodos
\endgroup
}
```

`\todoii` Modified to use `\textcolor` instead of `\color`.

```
\renewcommand{\todoii}[2]{%
\ifthenelse{\equal{\@todoobeyfinal}{true}}{%
    \ifoptionfinal{\todoenable{false}}{\todoenable{true}}%
}{}%
\ifthenelse{\equal{\@todoenable}{true}}{%
\refstepcounter{todos}%
\noindent{%
    \todocolor%
    \LWR@textcurrentcolor{%
        \normalfont\scriptsize{\bfseries{\thetodos.#1}}%
    }%
}%
\addcontentsline{lod}{todos}{\protect{\thetodos. }#2}%
}{}%
}
```

File 45 `lwarp-ebook.sty`

§ 130 Package ebook

(*Emulates or patches code by* Jørgen Steensgaard.)

Pkg `ebook` ebook is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{ebook}

\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{2}

\providecommand{\pagefill}[1][0.001mm]{\noindent}

\providecommand{\ebook}{
\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{2}
}
```

File 46 `lwarp-ellipsis.sty`

§ 131 Package ellipsis

(*Emulates or patches code by* Peter J. Heslin.)

Pkg `ellipsis` ellipsis is emulated.

```
\LWR@ProvidesPackageDrop{ellipsis}

\newcommand{\ellipsisgap}{0.1em}
```

File 47 `lwarp-emptypage.sty`

§ 132 Package emptypage

Pkg `emptypage` emptypage is ignored.

for HTML output: Discard all options for lwarp-emptypage:

```
\LWR@ProvidesPackageDrop{emptypage}
```

File 48 `lwarp-endfloat.sty`

§ 133 Package endfloat

Pkg `endfloat` endfloat is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{endfloat}

\newcommand\figureplace{}
\newcommand\tableplace{}
\newcommand\floatplace[1]{}
\newcounter{posttable}
\newcounter{postfigure}
\newcommand*{\theposttbl}{}
\newcommand*{\thepostfig}{}
\newcommand{\AtBeginFigures}[1]{}
\newcommand{\AtBeginTables}[1]{}
\newcommand{\AtBeginDelayedFloats}[1]{}
\newcommand*{\processdelayedfloats}{}
\newcommand*{\efloatseparator}{}
```

File 49 `lwarp-endnotes.sty`

§ 134 Package endnotes

(*Emulates or patches code by* John Lavagnino.)

Pkg endnotes Used as-is.

table of contents To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\enoteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

for HTML output:

```
\LWR@ProvidesPackagePass{endnotes}

\def\enoteformat{%
% \rightskip\z@ \leftskip\z@ \parindent=1.8em
\leavevmode
% \llap{
\makeenmark
% }
}

\def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
\def\makeenmark{\@makeenmark}
```

File 50 **lwarp-enumerate.sty**

§ 135 Package enumerate

Pkg enumerate enumerate is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

for HTML output:

```
\LWR@ProvidesPackagePass{enumerate}
```

File 51 **lwarp-enumitem.sty**

§ 136 Package enumitem

(*Emulates or patches code by* JAVIER BEZOS.)

Pkg enumitem enumitem is supported with minor adjustments.

for HTML output:
```
\LWR@ProvidesPackagePass{enumitem}
```

for HTML output:
```
\begin{warpHTML}
```

\newlist {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}
\renewlist {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}

For enumitem lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```
\let\LWR@orignewlist\newlist

\renewcommand*{\newlist}[3]{%
\LWR@orignewlist{#1}{#2}{#3}%
\AtBeginEnvironment{#1}{\csuse{LWR@#2start}}%
\AtEndEnvironment{#1}{\csuse{LWR@#2end}}%
}

\end{warpHTML}
```

File 52 `lwarp-epigraph.sty`

§ 137 Package epigraph

(*Emulates or patches code by* Peter Wilson.)

Pkg epigraph epigraph is emulated.

for HTML output:
```
\LWR@ProvidesPackageDrop{epigraph}

\DeclareDocumentCommand{\qitem}{m m}
{
\begin{BlockClass}{qitem}
#1
\ifbool{FormatWP}
{\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
{\begin{BlockClass}{epigraphsource}}
#2
\end{BlockClass}
\end{BlockClass}
}

\DeclareDocumentCommand{\epigraph}{m m}
```

```
{
\begin{LWR@BlockClassWP}{text-align:right}{}{epigraph}
\qitem{#1}{#2}
\end{LWR@BlockClassWP}
}

\DeclareDocumentEnvironment{epigraphs}{}
{\LWR@BlockClassWP{text-align:right}{}{epigraph}}
{\endLWR@BlockClassWP}
```

Use css to format epigraphs.

The following are null commands for source compatibility:

```
\newenvironment*{flushepinormal}{}{}


\@ifclassloaded{memoir}{
\setlength{\epigraphwidth}{.5\linewidth}
\renewcommand{\textflush}{flushepinormal}
\renewcommand{\epigraphhead}[2][0]{#2}
\renewcommand{\dropchapter}[1]{}
\renewcommand*{\undodrop}{}
}{% not memoir
\newlength{\epigraphwidth}
\setlength{\epigraphwidth}{.5\linewidth}
\newcommand{\textflush}{flushepinormal}
\newcommand{\epigraphflush}{flushright}
\newcommand{\sourceflush}{flushright}
\newcommand*{\epigraphsize}{\small}
\newlength{\epigraphrule}
\newlength{\beforeepigraphskip}
\newlength{\afterepigraphskip}
\newcommand{\epigraphhead}[2][0]{#2}
\newcommand{\dropchapter}[1]{}
\newcommand*{\undodrop}{}
}% not memoir

\let\cleartoevenpage\relax% also in nextpage
\newcommand{\cleartoevenpage}[1][]{}
```

File 53 `lwarp-eso-pic.sty`

§ 138 Package eso-pic

(*Emulates or patches code by* Rolf Niepraschk.)

Pkg `eso-pic` eso-pic is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{eso-pic}

\newcommand*{\LenToUnit}{}
\newcommand{\AtPageUpperLeft}[1]{}
\newcommand{\AtPageLowerLeft}[1]{}
\newcommand{\AtPageCenter}[1]{}
\newcommand{\AtStockLowerLeft}[1]{}
\newcommand{\AtStockUpperLeft}[1]{}
\newcommand{\AtStockCenter}[1]{}
\newcommand{\AtTextUpperLeft}[1]{}
\newcommand{\AtTextLowerLeft}[1]{}
\newcommand{\AtTextCenter}[1]{}
\NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
\renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
\NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
\newcommand*{\ClearShipoutPictureBG}{}
\newcommand*{\ClearShipoutPicture}{}
\newcommand*{\ClearShipoutPictureFG}{}
\newcommand{\gridSetup}[6][]{}
```

File 54 `lwarp-everypage.sty`

§ 139 Package everypage

(*Emulates or patches code by* Sergio Callegari.)

Pkg `everypage` everypage is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{everypage}

\newcommand*{\AddEverypageHook}[1]{}
\newcommand*{\AddThispageHook}[1]{}
```

File 55 `lwarp-everyshi.sty`

§ 140 Package everyshi

(*Emulates or patches code by* Martin Schröder.)

Pkg `everyshi` Emulated.

for HTML output: Discard all options for lwarp-everyshi:

```
\LWR@ProvidesPackageDrop{everyshi}
```

```
\newcommand*{\EveryShipout}[1]{}
\newcommand*{\AtNextShipout}[1]{}
```

File 56 `lwarp-extramarks.sty`

§ 141 Package extramarks

(*Emulates or patches code by* Piet van Oostrum.)

Pkg `extramarks` extramarks is emulated.

for HTML output: Discard all options for lwarp-extramarks:

```
\LWR@ProvidesPackageDrop{extramarks}
```

```
\newcommand*{\extramarks}[2]{}
\newcommand*{\firstleftxmark}{}
\newcommand*{\lastleftxmark}{}
\newcommand*{\firstrightxmark}{}
\newcommand*{\lastrightxmark}{}
\newcommand*{\firstxmark}{}
\newcommand*{\lastxmark}{}
\newcommand*{\topxmark}{}
\newcommand*{\topleftxmark}{}
\newcommand*{\firstleftmark}{}
\newcommand*{\lastrightmark}{}
```

File 57 `lwarp-fancybox.sty`

§ 142 Package fancybox

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg `fancybox` fancybox is supported with some patches.

framed equation example

fancybox's documentation has an example `FramedEqn` environment which combines math, `\Sbox`, a minipage, and an `\fbox`. This combination requires that the entire environment be enclosed inside a `lateximage`, which is done by adding `\lateximage` at the very start of `FramedEqn`'s beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the HTML `alt` attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives

`\fbox` works with fancybox. Also see lwarp's `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

framed table example

The fancybox documentation's example framed table using an `\fbox` containing a `tabular` does not work with lwarp, but the `FramedTable` environment does work if `\fbox` is replaced by `\fboxBlock`. This method loses HTML formatting. A better method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim

lwarp does not support the `verbatim` environment inside a span, box, or fancybox's `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The fancybox documentation's example `FramedVerb` may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed `\VerbBox`

fancybox's `\VerbBox` may be used inside `\fbox`.

indented alignment

`LVerbatim`, `\LVerbatimInput`, and `\LUseVerbatim` indent with horizontal space which may not line up exactly with what `pdftotext` detects. Some lines may be off slightly in their left edge.

for HTML output:

```
\begin{warpHTML}
```

```
\LWR@ProvidesPackagePass{fancybox}

\renewcommand*{\@shadowbox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black]{shadowbox}{#1}}%
{\InlineClass{shadowbox}{#1}}%
}

\renewcommand*{\@doublebox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px double black]{doublebox}{#1}}%
{\InlineClass{doublebox}{#1}}%
}

\renewcommand*{\@ovalbox}[2]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
{%
    \ifthenelse{\isequivalentto{#1}{\thinlines}}%
    {\InlineClass{ovalbox}{#2}}%
    {\InlineClass{Ovalbox}{#2}}%
}%
}
```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```
\let\LWR@origSbox\Sbox

\def\Sbox{\LWR@origSbox\LWR@nestspan}


\let\LWR@origendSbox\endSbox

\def\endSbox{\endLWR@nestspan\LWR@origendSbox}
```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```
\RenewEnviron{Beqnarray}
{\LWR@eqnarrayfactor}

\csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

`\GenericCaption` is enclosed in an HTML block:

```
\renewcommand{\GenericCaption}[1]{%
\LWR@figcaption%
#1%
```

```
\endLWR@figcaption%
}
```

`Btrivlist` is enclosed in an HTML block:

```
\RenewDocumentEnvironment{Btrivlist}{m o}
{\begin{BlockClass}{Btrivlist}\tabular{#1}}
{\endtabular\end{BlockClass}}
```

`Btrivlist` is also neutralized when used inside a span:

```
\AtBeginEnvironment{LWR@nestspan}{%
\RenewDocumentEnvironment{Btrivlist}{m o}{}{}%
\RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
}
```

lwarp's handling of `\item` is patched to accept fancybox's optional arguments:

```
\let\LWRFB@origitemizeitem\LWR@itemizeitem
\let\LWRFB@origdescitem\LWR@descitem
\LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}

\RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
\RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}
```

The various boxed lists become regular lists:

```
\renewenvironment{Bitemize}[1][]{\begin{itemize}}{\end{itemize}}
\renewenvironment{Benumerate}[1][]{\begin{enumerate}}{\end{enumerate}}
\renewenvironment{Bdescription}[1][]{\begin{description}}{\end{description}}
```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```
\RenewDocumentCommand{\boxput}{s d() m m}{%
\IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
}
```

Neutralized commands:

```
\RenewDocumentCommand{\fancyput}{s d() m}{}
\RenewDocumentCommand{\thisfancyput}{s d() m}{}

\RenewDocumentCommand{\fancypage}{m m}{}
\RenewDocumentCommand{\thisfancypage}{m m}{}

```

```
\def\LandScape#1{}
\def\endLandScape{}
\def\@Landscape#1#2#3{}
\def\endLandscape{}
```

Low-level patches for `UseVerbatim` and friends:

```
\let\LWRFB@UseVerbatim\UseVerbatim
\renewcommand*{\UseVerbatim}[1]{%
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
\LWRFB@UseVerbatim{#1}%
\LWR@afterendverbatim%
}

\let\LWRFB@LUseVerbatim\LUseVerbatim

\renewcommand*{\LUseVerbatim}[1]{%
\LWR@atbeginverbatim{LVerbatim}%\unskip\LWR@origvspace*{-\baselineskip}%%
\noindent%
\LWRFB@LUseVerbatim{#1}%
\LWR@afterendverbatim%
}

\def\@BUseVerbatim[#1]#2{%
\LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
\LWRFB@UseVerbatim{#2}%
\LWR@afterendverbatim%
}

\end{warpHTML}
```

File 58 `lwarp-fancyhdr.sty`

§ 143 Package fancyhdr

(*Emulates or patches code by* Piet van Oostrum.)

Pkg `fancyhdr` fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```
\LWR@ProvidesPackageDrop{fancyhdr}

\newcommand*{\fancyhead}[2][]{}
\newcommand*{\fancyfoot}[2][]{}
\newcommand*{\fancyhf}[2][]{}
```

```
\newcommand*{\fancypagestyle}[2]{}
\newcommand*{\lhead}[2][]{}
\newcommand*{\chead}[2][]{}
\newcommand*{\rhead}[2][]{}
\newcommand*{\lfoot}[2][]{}
\newcommand*{\cfoot}[2][]{}
\newcommand*{\rfoot}[2][]{}
\newcommand*{\headrulewidth}{}
\newcommand*{\footrulewidth}{}
\newcommand*{\fancyheadoffset}[2][]{}
\newcommand*{\fancyfootoffset}[2][]{}
\newcommand*{\fancyhfoffset}[2][]{}
\newcommand*{\iffloatpage}[2]{#2}
\newcommand*{\ifftopfloat}[2]{#2}
\newcommand*{\iffbotfloat}[2]{#2}
```

File 59 `lwarp-fancyref.sty`

§ 144 Package fancyref

Pkg `fancyref` fancyref is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{fancyref}

\newcommand*{\fancyrefhook}[1]{#1}
\newcommand*{\fref}[2][]{\cref{#2}}
\newcommand*{\Fref}[2][]{\Cref{#2}}
\newcommand*{\fancyrefchaplabelprefix}{chap}
\newcommand*{\fancyrefenumlabelprefix}{enum}
\newcommand*{\fancyrefeqlabelprefix}{eq}
\newcommand*{\fancyreffiglabelprefix}{fig}
\newcommand*{\fancyreffnlabelprefix}{fn}
\newcommand*{\fancyrefseclabelprefix}{sec}
\newcommand*{\fancyreftablabelprefix}{tab}
\newcommand*{\fancyrefchangeprefix}[2]{}
\newcommand*{\fancyrefargdelim}{:}
\newcommand*{\fancyrefloosespacing}{~}
\newcommand*{\fancyreftightspacing}{\,}
\newcommand*{\fancyrefdefaultspacing}{\fancyrefloosespacing}
\providecommand*{\chaptername}{Chapter}
\providecommand*{\figurename}{Figure}
\providecommand*{\pagename}{Page}
\providecommand*{\tablename}{Table}
\renewcommand*{\Frefchapname}{\chaptername}
\renewcommand*{\Frefenumname}{Item}
\renewcommand*{\Frefeqname}{Equation}
\renewcommand*{\Freffigname}{\figurename}
```

```
\renewcommand*{\Freffnname}{Footnote}
\renewcommand*{\Frefonname}{On}
\renewcommand*{\Frefpgname}{\pagename}
\renewcommand*{\Frefsecname}{Section}
\renewcommand*{\Frefseename}{See}
\renewcommand*{\Freftabname}{\tablename}
\renewcommand*{\Freffigshortname}{Fig.}
\renewcommand*{\Frefpgshortname}{P.}
\renewcommand*{\Freftabshortname}{Tab.}
\renewcommand*{\frefchapname}{\MakeLowercase{\Frefchapname}}
\renewcommand*{\frefenumname}{\MakeLowercase{\Frefenumname}}
\renewcommand*{\frefeqname}{\MakeLowercase{\Frefeqname}}
\renewcommand*{\freffigname}{\MakeLowercase{\Freffigname}}
\renewcommand*{\freffnname}{\MakeLowercase{\Freffnname}}
\renewcommand*{\frefonname}{\MakeLowercase{\Frefonname}}
\renewcommand*{\frefpgname}{\MakeLowercase{\Frefpgname}}
\renewcommand*{\frefsecname}{\MakeLowercase{\Frefsecname}}
\renewcommand*{\frefseename}{\MakeLowercase{\Frefseename}}
\renewcommand*{\freftabname}{\MakeLowercase{\Freftabname}}
\renewcommand*{\freffigshortname}{\MakeLowercase{\Freffigshortname}}
\renewcommand*{\frefpgshortname}{\MakeLowercase{\Frefpgshortname}}
\renewcommand*{\freftabshortname}{\MakeLowercase{\Freftabshortname}}
\newcommand*{\fancyrefaddcaptions}[2]{}
\newcommand*{\fancyrefdefaultformat}{\@empty}
\newcommand{\frefformat}[3]{}
\newcommand{\Frefformat}[3]{}
```

File 60 `lwarp-fancyvrb.sty`

§ 145 Package fancyvrb

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg `fancyvrb` fancyvrb is supported with some patches.

for HTML output:

```
\RequirePackage{xcolor}% for \convertcolorspec
\LWR@ProvidesPackagePass{fancyvrb}
```

for HTML output:

```
\begin{warpHTML}
```

Initial default patch for fancyvrb:

```
\fvset{frame=none}%
```

For `\VerbatimFootnotes`:

```
\renewcommand{\VerbatimFootnotes}{
```

```
\PackageError{lwarp}
{Verbatim footnotes are not yet supported by lwarp.}
{This may be improved some day.}
}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
\AfterEndPreamble{
\LWR@traceinfo{Patching Verbatim.}

\preto\FVB@Verbatim{\LWR@forcenewpage}
\preto\FVB@LVerbatim{\LWR@forcenewpage}
% \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.
```

Simplified to remove PDF formatting:

```
\def\FV@BeginListFrame@Single{%
  \FV@SingleFrameLine{\z@}%
}

\def\FV@EndListFrame@Single{%
  \FV@SingleFrameLine{\@ne}%
}

\def\FV@BeginListFrame@Lines{%
  \FV@SingleFrameLine{\z@}%
}

\def\FV@EndListFrame@Lines{%
    \FV@SingleFrameLine{\@ne}%
}

\renewcommand*{\FV@SingleFrameSep}{}
```

Adds HTML formatting:

```
\def\FV@BUseVerbatim#1{%
    \LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
    \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
    \LWR@afterendverbatim%
}
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
\newcommand*{\LWR@FVstyle}{}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the fancyvrb package.

```
\newcommand*{\LWR@fvstartnone}{%
\LWR@traceinfo{fvstartnone}%
% \hbox to\z@{
\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
% }%
}

\newcommand*{\LWR@fvendnone}{%
\LWR@traceinfo{fvendnone}%
% \hbox to\z@{
\LWR@afterendverbatim%
% }%
}

\newcommand*{\LWR@fvstartsingle}{%
\LWR@traceinfo{fvstartsingle}%
\LWR@fvstartnone%
\FV@BeginListFrame@Single%
}

\newcommand*{\LWR@fvendsingle}{%
\LWR@traceinfo{fvendsingle}%
\FV@EndListFrame@Single%
\LWR@fvendnone%
}

\newcommand*{\LWR@fvstartline}{%
\LWR@traceinfo{fvstartline}%
\LWR@fvstartnone%
% \setlength{\LWR@templengthone}{\baselineskip}%
\FV@BeginListFrame@Lines%
% \setlength{\baselineskip}{\LWR@templengthone}%
% \setlength{\baselineskip}{5pt}%
}

\newcommand*{\LWR@fvendline}{%
\LWR@traceinfo{fvendline}%
\FV@EndListFrame@Lines%
\LWR@fvendnone%
}
```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the fancyvrb package.

```
\newcommand*{\LWR@FVfindbordercolor}{%
```

```
\FancyVerbRuleColor%
\LWR@findcurrenttextcolor%
\color{black}%
}

% border width of \FV@FrameRule
\newcommand*{\LWR@FVborderstyle}[1]{%
padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
\LWR@FVfindbordercolor %
border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ; %
}

\def\FV@Frame@none{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\FV@Frame@none% default values

\def\FV@Frame@single{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
\let\FV@BeginListFrame\LWR@fvstartsingle%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\FV@RightListFrame@Single%
\let\FV@EndListFrame\LWR@fvendsingle}

\def\FV@Frame@lines{%
\renewcommand*{\LWR@FVstyle}{%
    \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendline}

\def\FV@Frame@topline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\def\FV@Frame@bottomline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendline}
```

```
\def\FV@Frame@leftline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%
% To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
\ifx\FancyVerbFillColor\relax%
\let\FV@FrameFillLine\relax%
\else%
\@tempdima\FV@FrameRule\relax%
\multiply\@tempdima-\tw@%
\edef\FV@FrameFillLine{%
{\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
\kern-\number\@tempdima sp}}%
\fi%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}
```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```
\def\FV@SingleFrameLine#1{%
%   \hbox to\z@{%
%     \kern\leftmargin
    \ifnum#1=\z@\relax
      \let\FV@Label\FV@LabelBegin
    \else
      \let\FV@Label\FV@LabelEnd
    \fi
    \ifx\FV@Label\relax
%       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
    \else
      \ifnum#1=\z@
%         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
        \ifx\FV@LabelPositionTopLine\relax
        \else
        \LWR@FVfindbordercolor
        \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
        \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
        \LWR@htmltagc{/div}
        \fi
      \else
%         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
        \ifx\FV@LabelPositionBottomLine\relax
        \else
        \LWR@FVfindbordercolor

        \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
        \LWR@origtextrm{\FV@LabelEnd}
```

```
        \LWR@htmltagc{/div}
        \fi
      \fi
    \fi
%     \hss
%     }
}
```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```
\def\FV@ListProcessLine#1{%
  \hbox to \hsize{%
%      \kern\leftmargin
      \hbox to \VerbatimHTMLWidth {%
       \ifcsvoid{FV@LeftListNumber}{}{\kern 2.5em}%
        \FV@LeftListNumber%
%       \FV@LeftListFrame
      \FancyVerbFormatLine{#1}%
      \hss%
%       \FV@RightListFrame
      \FV@RightListNumber%
    }%
      \hss% required to avoid underfull hboxes
}
}
```

Env `BVerbatim`

```
\AtBeginEnvironment{BVerbatim}
{%
\LWR@forcenewpage% instead of \preto
\LWR@atbeginverbatim{bverbatim}%
}

\AfterEndEnvironment{BVerbatim}
{%
\leavevmode\par\LWR@origvspace{-\baselineskip}%
\LWR@afterendverbatim%
}
```

End of the modifications to make at the end of the preamble:

```
} % \AfterEndPreamble

\end{warpHTML}
```

File 61 `lwarp-figcaps.sty`

§ 146 Package figcaps

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `figcaps` Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```
\LWR@ProvidesPackageDrop{figcaps}

\newcommand*{\figcapson}{}
\newcommand*{\figcapsoff}{}
\newcommand*{\printfigures}{}
\newcommand*{\figmarkon}{}
\newcommand*{\figmarkoff}{}
\def\figurecapname{Figure Captions}
\def\tablepagename{Tables}
\def\figurepagename{Figures}
```

File 62 `lwarp-fix2col.sty`

§ 147 Package fix2col

Pkg `fix2col` fix2col is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fix2col}
```

File 63 `lwarp-fixme.sty`

§ 148 Package fixme

(*Emulates or patches code by* Didier Verna.)

Pkg `fixme` fixme is patched for use by lwarp.

⚠ external layouts External layouts (`\fxloadlayouts`) are not supported.

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after fixme is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

for HTML output:

```
\LWR@ProvidesPackagePass{fixme}
```

Restore lwarp’s version of `\@wrindex`, ignoring the fixme package’s `target` option:

```
\let\@wrindex\LWR@wrindex
```

Float-related macros required by lwarp:

```
\newcommand{\ext@fixme}{lox}

\renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}
```

Other modifications:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}

\renewcommand*\FXLayoutInline[3]{ %
\InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
    {\@fxtextstd{#1}{#2}{#3}}%
}

\def\FXFaceEnvHTMLStyle{font-weight:bold}

\renewcommand*\FXEnvLayoutPlainBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\ignorespaces#2 \fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutPlainEnd[2]{\endBlockClass}

\renewcommand*\FXEnvLayoutSignatureBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutSignatureEnd[2]{\@fxsignature{#2}\endBlockClass}

\def\FXFaceSignatureHTMLStyle{font-style:italic}

\DeclareRobustCommand*\@fxsignature[1]{%
\ifthenelse{\equal{#1}{}}%
{}%
{ -- {\InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{#1}}}%
```

```
}


\def\FXFaceTargetHTMLStyle{font-style:italic}

\renewcommand\FXTargetLayoutPlain[2]{%
    \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
}
```

File 64 `lwarp-fixmetodonotes.sty`

§ 149 Package fixmetodonotes

(*Emulates or patches code by* Gioele Barabucci.)

Pkg `fixmetodonotes` fixmetodonotes is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{fixmetodonotes}

\renewcommand{\NOTES@addtolist}[2]{%
    \refstepcounter{NOTES@note}%
%   \phantomsection% REMOVED
    \addcontentsline{notes}{NOTES@note}{%
        \protect\numberline{\theNOTES@note}{{#1}: {#2}}%
    }%
}

\renewcommand{\NOTES@marker}[2]{\fbox{%
    \textcolor{#2}{% WAS \color
        \textbf{#1}}%
    }}

\renewcommand{\NOTES@colorline}[2]{%
  \bgroup%
    \ULon{\LWR@backgroundcolor{#1}{#2}}%
}
```

File 65 `lwarp-flafter.sty`

§ 150 Package flafter

Pkg `flafter` flafter is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{flafter}
\providecommand\fl@trace[1]{}
```

File 66 `lwarp-float.sty`

§ 151 Package

float and \newfloat

(*Emulates or patches code by* Anselm Lingnau.)

Pkg `float` float is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{float}[2016/03/04]
```

See section 65.2 for the `\listof` command.

`\newfloat` {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the `\newfloat` command from the float package.

"placement" is ignored.

```
\NewDocumentCommand{\newfloat}{m m m o}{%
\IfValueTF{#4}%
{\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}%
{\DeclareFloatingEnvironment[fileext=#3]{#1}}%
```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
\cslet{listof#1s}\relax%
\cslet{listof#1es}\relax%
}
```

`\floatname` {⟨*type*⟩} {⟨*name*⟩}

Sets the text name of the float, such as "Figure".

```
\NewDocumentCommand{\floatname}{m +m}{%
\SetupFloatingEnvironment{#1}{name=#2}%
}
```

`\floatplacement` {⟨*type*⟩} {⟨*placement*⟩}

Float placement is ignored.

```
\newcommand*{\floatplacement}[2]{%
\SetupFloatingEnvironment{#1}{placement=#2}%
}
```

`\floatstyle` {⟨*style*⟩}

Float styles are ignored.

```
\newcommand{\floatstyle}[1]{%
}
```

`\restylefloat` `*` {⟨*style*⟩}

Float styles are ignored.

```
\NewDocumentCommand{\restylefloat}{s m}{%
}
```

File 67 **lwarp-floatflt.sty**

§ 152 Package floatflt

(*Emulates or patches code by* Mats Dahlgren.)

Pkg `floatflt` Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
\LWR@ProvidesPackageDrop{floatflt}
```

Env `[⟨⟩]` offset {⟨*type*⟩} {⟨*width*⟩} Borrowed from the lwarp version of keyfloat:

```
\NewDocumentEnvironment{KFLTfloatflt@marginfloat}{O{-1.2ex} m m}
{%
\setlength{\LWR@templengthone}{#3}%
\uselengthunit{PT}%
\LWR@BlockClassWP{%
    float:right; %
    width:\rndprintlength{\LWR@templengthone}; %
    margin:10pt%
}{%
    width:\rndprintlength{\LWR@templengthone}%
}%
{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}
```

Env `floatingfigure` [⟨*placement*⟩] {⟨*width*⟩}

```
\DeclareDocumentEnvironment{floatingfigure}{o m}
  {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
```

```
21  {\end{KFLTfloatflt@marginfloat}}
```

Env `floatingtable` [⟨*placement*⟩]

```
22 \DeclareDocumentEnvironment{floatingtable}{o}
23  {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
24  {\end{KFLTfloatflt@marginfloat}}
```

File 68 `lwarp-floatpag.sty`

§ 153 Package floatpag

(*Emulates or patches code by* Vytas Statulevičius and Sigitas Tolušis.)

Pkg `floatpag` Emulated.

for HTML output: Discard all options for lwarp-floatpag:

```
1 \LWR@ProvidesPackageDrop{floatpag}
```

```
2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}
```

File 69 `lwarp-floatrow.sty`

§ 154 Package floatrow

(*Emulates or patches code by* Olga Lapko.)

Pkg `floatrow` floatrow is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{floatrow}
```

⚠ misplaced alignment
alignment tab character &

Use `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 8.8.

⚠ subfig package

When combined with the subfig package, while inside a `subfloatrow` `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.

⚠ `\FBwidth`, `\FBheight`

The emulation of floatrow does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:

- Use fixed lengths. lwarp will scale the HTML lengths appropriately.

- Use warpprint and warpHTML environments to select appropriate values for each case.
- Inside a warpHTML environment, manually change \FBwidth or \FBheight before the \ffigbox or \ttabbox. Use \FBwidth or \FBheight normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that subfig is used instead:

```
\newbool{LWR@subcaptionloaded}

\AtBeginDocument{
\@ifpackageloaded{subcaption}
{\booltrue{LWR@subcaptionloaded}}
{\boolfalse{LWR@subcaptionloaded}}
}
```

\floatbox [⟨*1 preamble*⟩] {⟨*2 captype*⟩} [⟨*3 width*⟩] [⟨*4 height*⟩] [⟨*5 vert pos*⟩] {⟨*6 caption*⟩} {⟨*7 object*⟩}

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of subcaption or subfig.

```
\NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
\ifbool{LWR@subcaptionloaded}%
{% subcaption
```

For subcaption:

```
    \ifbool{LWR@insubfloatrow}%
    {% subcaption in a subfloatrow
```

subfigure and subtable environments take width as an argument.

```
        \IfValueTF{#3}%
        {\@nameuse{sub#2}{#3}}%
        {\@nameuse{sub#2}{\linewidth}}%
    }% subcaption in a subfloatrow
    {% subcaption not in subfloatrow
```

figure and table environments do not take a width argument.

```
        \@nameuse{#2}%
    }% subcaption not in subfloatrow
    #6

    #7
```

End the environments:

```
    \ifbool{LWR@insubfloatrow}%
    {\@nameuse{endsub#2}}%
    {\@nameuse{end#2}}%
}% subcaption
{% assume subfig
```

For subfig:

```
\ifbool{LWR@insubfloatrow}%
{% subfig in a subfloatrow
```

`\subfloat` is a macro, not an environment.

Package subfig's `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```
    \begingroup
    \let\caption\@firstofone
    \subfloat[#6]{#7}
    \endgroup
}% subfig in a subfloatrow
{% subfig package, but not a subfig
```

`figure` and `table` are environments:

```
\@nameuse{#2}
#6

#7
\@nameuse{end#2}
}% subfig package, but not a subfig
}% assume subfig
}
```

Not used:

```
\newcommand*{\nocapbeside}{}
\newcommand*{\capbeside}{}
\newcommand*{\captop}{}
\newlength{\FBwidth}
\setlength{\FBwidth}{.3\linewidth}
\newlength{\FBheight}
\setlength{\FBheight}{2in}
\newcommand*{\useFCwidth}{}
\newcommand{\floatsetup}[2][]{}
\newcommand{\thisfloatsetup}[1]{}
\newcommand{\clearfloatsetup}[1]{}
\newcommand*{\killfloatstyle}{}
```

\newfloatcommand {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\newfloatcommand}{m m o o}{%
\@namedef{#1}{
\floatbox{#2}
}
}
```

\renewfloatcommand {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\renewfloatcommand}{m m o o}{%
\@namedef{#1}{%
\floatbox{#2}
}
}
```

\ffigbox [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ffigbox}{figure}[\nocapbeside][]
```

\ttabbox [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

\fcapside [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{fcapside}{figure}[\capbeside][]
```

Env floatrow [⟨*numfloats*⟩]

The row of floats is placed into a `<div>` of class `floatrow`.

```
\newenvironment*{floatrow}[1][2]
{
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, divide the `\linewidth` by the number of floats.

```
\booltrue{LWR@infloatrow}
\setlength{\linewidth}{6in/#1}
}
{
\boolfalse{LWR@infloatrow}
\endBlockClass
}
```

Keys for \DeclareNewFloatType:

```
\newcommand*{\LWR@frowkeyplacement}{}
\newcommand*{\LWR@frowkeyname}{}
\newcommand*{\LWR@frowkeyfileext}{}
\newcommand*{\LWR@frowkeywithin}{}
\newcommand*{\LWR@frowkeycapstyle}{}

\define@key{frowkeys}{placement}{}%
\define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
\define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
\define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
\define@key{frowkeys}{relatedcapstyle}{}%
```

\DeclareNewFloatType {⟨*type*⟩} {⟨*options*⟩}

Use \listof{type}{Title} to print a list of the floats.

```
\newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
\renewcommand*{\LWR@frowkeyplacement}{}%
\renewcommand*{\LWR@frowkeyname}{}%
\renewcommand*{\LWR@frowkeyfileext}{}%
\renewcommand*{\LWR@frowkeywithin}{}%
\renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
\LWR@traceinfo{about to setkeys frowkeys}%
\setkeys{frowkeys}{#2}%
\LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
\ifthenelse{\equal{\LWR@frowkeywithin}{}}%
{%
    \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ %
        \LWR@frowkeyfileext}%
    \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
}%
{%
    \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ %
        \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
    \newfloat{#1}{\LWR@frowkeyplacement}%
    {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
    \LWR@traceinfo{finished newfloat #1}%
}%
```

Rename the float if a name was given:

```
\ifthenelse{\equal{\LWR@frowkeyname}{}}%
```

```
{}%
{\floatname{#1}{\LWR@frowkeyname}}%
}
```

Not used:

```
\newcommand{\buildFBBOX}[2]{}
\newcommand*{\CenterFloatBoxes}{}
\newcommand*{\TopFloatBoxes}{}
\newcommand*{\BottomFloatBoxes}{}
\newcommand*{\PlainFloatBoxes}{}

\newcommand{\capsubrowsettings}{}

\NewDocumentCommand{\RawFloats}{o o}{}
```

`\RawCaption` `{`⟨*text*⟩`}`

To be used inside a minipage or parbox.

```
\newcommand{\RawCaption}[1]{#1}
```

`\floatfoot` `{`⟨*text*⟩`}`

Places additional text inside a float, inside a css `<div>` of class `floatfoot`.

```
\NewDocumentCommand{\floatfoot}{s +m}{%
\begin{BlockClass}{floatfoot}
#2
\end{BlockClass}
}
```

Used to compute `\linewidth`.

```
\newbool{LWR@insubfloatrow}
\boolfalse{LWR@insubfloatrow}
```

Env `subfloatrow` `[`⟨*num_floats*⟩`]`

```
\newenvironment*{subfloatrow}[1][2]
{
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
\begingroup
```

```
\booltrue{LWR@insubfloatrow}
}
{
\endgroup
\endBlockClass
\boolfalse{LWR@insubfloatrow}
}
```

File 70 `lwarp-fltrace.sty`

§ 155 Package fltrace

Pkg `fltrace` fltrace is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fltrace}

\def\tracefloats{}
\def\tracefloatsoff{}
\def\tracefloatvals{}
```

File 71 `lwarp-flushend.sty`

§ 156 Package flushend

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `flushend` Emulated.

for HTML output: Discard all options for lwarp-flushend:

```
\LWR@ProvidesPackageDrop{flushend}
%    \end{ma-crocode}
%
%    \begin{macrocode}
\newcommand*{\flushend}{}
\newcommand*{\raggedend}{}
\newcommand*{\flushcolsend}{}
\newcommand*{\raggedcolsend}{}
\newcommand*{\atColsBreak}[1]{}
\newcommand*{\atColsEnd}[1]{}
\newcommand*{\showcolsendrule}{}
```

File 72 `lwarp-fncychap.sty`

§ 157 Package fncychap

(*Emulates or patches code by* Ulf A. Lindgren.)

Pkg `fncychap` fncychap is emulated.

for HTML output: Discard all options for lwarp-fncychap:

```
\LWR@ProvidesPackageDrop{fncychap}

\def\mghrulefill#1{}
\def\ChNameLowerCase{}
\def\ChNameUpperCase{}
\def\ChNameAsIs{}
\def\ChTitleLowerCase{}
\def\ChTitleUpperCase{}
\def\ChTitleAsIs{}
\newcommand{\ChRuleWidth}[1]{}
\newcommand{\ChNameVar}[1]{}
\newcommand{\ChNumVar}[1]{}
\newcommand{\ChTitleVar}[1]{}
\newcommand{\TheAlphaChapter}{}
\newcommand{\DOCH}{}
\newcommand{\DOTI}[1]{}
\newcommand{\DOTIS}[1]{}
\newlength{\mylen}
\newlength{\myhi}
\newlength{\px}
\newlength{\py}
\newlength{\pyy}
\newlength{\pxx}
\newlength{\RW}
\newcommand{\FmN}[1]{#1}
\newcommand{\FmTi}[1]{#1}
```

File 73 `lwarp-fnpos.sty`

§ 158 Package fnpos

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `fnpos` fnpos is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{fnpos}

\newcommand*{\makeFNbottom}{}
\newcommand*{\makeFNmid}{}
\newcommand*{\makeFNbelow}{}
\newcommand*{\makeFNabove}{}
```

File 74 `lwarp-fontenc.sty`

§ 159 Package fontenc

Pkg `fontenc` If using pdfLaTeX, lwarp used to require fontspec be loaded before lwarp, but now lwarp itself loads `\fontspec` with T1 encoding, which lwarp requires. fontspec is now allowed to be loaded with another encoding after lwarp.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

for HTML output:

```
\LWR@ProvidesPackagePass{fontenc}
```

File 75 `lwarp-fontspec.sty`

§ 160 Package fontspec

Pkg `fontspec` Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output:

```
\LWR@ProvidesPackageDrop{fontspec}

\LWR@loadbefore{fontspec}
```

File 76 `lwarp-footmisc.sty`

§ 161 Package footmisc

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `footmisc` footmisc is emulated.

```
\LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
\newcommand{\footnotelayout}{}
\newcommand{\setfnsymbol}[1]{}
\NewDocumentCommand{\DefineFNsymbols}{s m o m}{}

\newdimen\footnotemargin
\footnotemargin1.8em\relax

\newcommand*\hangfootparskip{0.5\baselineskip}
\newcommand*\hangfootparindent{0em}%

\let\pagefootnoterule\footnoterule
\let\mpfootnoterule\footnoterule
\def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}

\providecommand*{\multiplefootnotemarker}{3sp}
\providecommand*{\multfootsep}{,}
```

Using cleveref:

```
\providecommand*{\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
\newcommand\mpfootnotemark{%
  \@ifnextchar[%
    \@xmpfootnotemark
    {%
      \stepcounter\@mpfn
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark
    }%
}
\def\@xmpfootnotemark[#1]{%
  \begingroup
    \csname c@\@mpfn\endcsname #1\relax
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotemark
}
```

File 77 `lwarp-footnote.sty`

§ 162 Package footnote

(*Emulates or patches code by* Mark Wooding.)

Pkg `footnote` footnote is used with minor patches.

for HTML output:

```
\LWR@ProvidesPackagePass{footnote}
```

Removed print-version formatting:

```
\def\fn@startnote{%
%   \@parboxrestore%
  \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
%   \color@begingroup% *** conflicts with lwarp
}

% \let\fn@endnote\color@endgroup% *** conflicts with lwarp
\def\fn@endnote{%
\LWR@htmltagc{/\LWR@tagregularparagraph}%
\LWR@orignewline%
}
```

Removed print-version formatting:

```
\def\fn@startfntext{%
  \setbox\z@\vbox\bgroup%
    \fn@startnote%
    \fn@prefntext%
    \ignorespaces%
}
```

Removed print-version formatting, added closing paragraph tag:

```
\def\fn@endfntext{%
    \LWR@htmltagc{/\LWR@tagregularparagraph}%
    \LWR@orignewline%
    \fn@postfntext%
  \egroup%
  \begingroup%
    \let\@makefntext\@empty%
    \let\@finalstrut\@gobble%
    \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
    \@footnotetext{\unvbox\z@}%
  \endgroup%
}
```

These have been redefined, so re-`\let` them again:

```
\let\endfootnote\fn@endfntext
\let\endfootnotetext\endfootnote
```

File 78 `lwarp-footnotehyper.sty`

§ 163 Package footnotehyper

Pkg `footnotehyper` footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```
\RequirePackage{footnote}
\LWR@ProvidesPackageDrop{footnotehyper}
```

File 79 `lwarp-footnpag.sty`

§ 164 Package footnpag

Pkg `footnpag` footnpag is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{footnpag}
```

File 80 `lwarp-framed.sty`

§ 165 Package framed

(*Emulates or patches code by* Donald Arseneau.)

Pkg `framed` framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```
\LWR@ProvidesPackagePass{framed}
\RequirePackage{xcolor}% for \convertcolorspec

\renewenvironment{framed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}

```

```
\renewenvironment{oframed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}


\renewenvironment{shaded}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{shaded}%
}
{\endBlockClass}

\renewenvironment{shaded*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{shaded}%
}
{\endBlockClass}


\renewenvironment{leftbar}{%
\LWR@forcenewpage
    \BlockClass{framedleftbar}
  \def\FrameCommand{}%
  \MakeFramed {}
}%
 {\endMakeFramed\endBlockClass}


\renewenvironment{snugshade}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{snugframed}%
}
{\endBlockClass}

\renewenvironment{snugshade*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \#\LWR@tempcolor]{snugframed}%
}
{\endBlockClass}

\let\oframed\framed
\let\endoframed\endframed


\RenewEnviron{titled-frame}[1]{%
```

```
\CustomFBox{#1}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
}
```

\CustomFBox {⟨*toptitle*⟩} {⟨*bottitle*⟩} {⟨*thicknesstop*⟩} {⟨*bottom*⟩} {⟨*left*⟩} {⟨*right*⟩} {⟨*text contents*⟩}

```
\renewcommand{\CustomFBox}[7]{%
\convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
\ifthenelse{\isempty{#1}}{}{% not empty
    \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#1}}%
    \end{BlockClass}
}% not empty

#7

\ifthenelse{\isempty{#2}}{}{% not empty
    \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
    \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#2}}%
    \end{BlockClass}
}% not empty
\end{BlockClass}
}
```

\TitleBarFrame [⟨*marker*⟩] {⟨*title*⟩} {⟨*contents*⟩}

```
\renewcommand\TitleBarFrame[3][]{
\CustomFBox
    {#2}{}%
    \fboxrule\fboxrule\fboxrule\fboxrule
    {#3}%
}

\renewcommand{\TF@Title}[1]{#1}
```

MakeFramed {⟨*settings*⟩}

```
\let\MakeFramed\relax
\let\endMakeFramed\relax

\NewEnviron{MakeFramed}[1]{%
\FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
}
```

`\fb@put@frame` {⟨*frame cmd no split*⟩} {⟨*frame cmd split*⟩}

```
\renewcommand*{\fb@put@frame}[2]{%
\relax%
\@tempboxa%
}
```

File 81 `lwarp-ftnright.sty`

§ 166 Package ftnright

Pkg `ftnright` ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
\LWR@ProvidesPackageDrop{ftnright}
```

File 82 `lwarp-fullpage.sty`

§ 167 Package fullpage

Pkg `fullpage` fullpage is ignored.

for HTML output: Discard all options for lwarp-fullpage:

```
\LWR@ProvidesPackageDrop{fullpage}
```

File 83 `lwarp-fullwidth.sty`

§ 168 Package fullwidth

(*Emulates or patches code by* Marco Daniel.)

Pkg `fullwidth` fullwidth is emulated.

A minipage is used, of no HTML width.

for HTML output:

```
\LWR@ProvidesPackageDrop{fullwidth}
```

```
\newenvironment*{fullwidth}[1][]{%
\minipagefullwidth%
\minipage{\linewidth}%
```

```
}
{%
\endminipage%
}
```

File 84 `lwarp-fwlw.sty`

§ 169 Package fwlw

Pkg `fwlw` fwlw is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fwlw}

\newbox\FirstWordBox      \global\setbox\FirstWordBox\hbox{}
\newbox\NextWordBox       \global\setbox\NextWordBox\hbox{}
\newbox\LastWordBox       \global\setbox\LastWordBox\hbox{}
\def\ps@fwlwhead{}
\def\ps@NextWordFoot{}
```

File 85 `lwarp-geometry.sty`

§ 170 Package geometry

(*Emulates or patches code by* Hideo Umeki.)

Pkg `geometry` geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for lwarp-geometry:

```
\LWR@ProvidesPackageDrop{geometry}

\renewcommand*{\geometry}[1]{}
\renewcommand*{\newgeometry}[1]{}
\renewcommand*{\restoregeometry}{}
\renewcommand*{\savegeometry}[1]{}
\renewcommand*{\loadgeometry}[1]{}
```

File 86 `lwarp-glossaries.sty`

§ 171 Package

glossaries

(*Emulates or patches code by* Nicola L.C. Talbot.)

Pkg `glossaries`

`xindy` is required for glossaries.

The default `style=item` option for glossaries conflicts with lwarp, so the style is forced to `index` instead.

The page number list in the printed form would become `\namerefs` in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

placement and TOC options

The glossaries may be placed in a numbered or unnumbered section, given a TOC entry, and placed inline or on their own HTML page:

Numbered section, on its own HTML page:

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

Unnumbered section, inline with the current HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

Unnumbered section, on its own HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

Opt `IndexLanguage`

The lwarp package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.

Opt `lwarpmk printglossary`
Opt `lwarpmk htmlglossary`

`lwarpmk` has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary` to process the glossaries created by glossaries using `xindy`.

for HTML output:

```
\PassOptionsToPackage{xindy}{glossaries}
\LWR@ProvidesPackagePass{glossaries}
```

```
\setupglossaries{nonumberlist}
\setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
\renewcommand*{\@p@glossarysection}[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the TOC entry is made after the section is created:

```
      \csname\@@glossarysec\endcsname*{#2}%
    \@gls@toc{#1}{\@@glossarysec}% Moved after the previous line.
  }%
  \@@glossaryseclabel
}
```

File 87 `lwarp-graphics.sty`

§ 172 Package graphics

(*Emulates or patches code by* D. P. CARLISLE.)

Pkg graphics graphics is emulated.

for HTML output:

```
\LWR@ProvidesPackagePass{graphics}
```

§ 172.1 Graphics extensions

`\DeclareGraphicsExtensions` {⟨*list*⟩}

`\AtBeginDocument` allow SVG files instead of PDF:

```
\AtBeginDocument{
\DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
\DeclareGraphicsRule{.svg}{svg}{.svg}{}
\DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
}
```

Inside a `lateximage`, allow PDF instead of SVG:

```
\appto\LWR@restoreorigformatting{%
\DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
}
```

§ 172.2 Length conversions and graphics options

⚠ whitespace

A scaled image in LaTeX by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

```
\AtBeginDocument{
\renewcommand*{\unitspace}{}
}
```

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyx` option.

```
\newlength{\LWR@igwidth}
\newlength{\LWR@igheight}
\newcommand*{\LWR@igwidthstyle}{}
\newcommand*{\LWR@igheightstyle}{}
\newcommand*{\LWR@igorigin}{}
\newcommand*{\LWR@igangle}{}
\newcommand*{\LWR@igxscale}{1}
\newcommand*{\LWR@igyscale}{1}
\newcommand*{\LWR@igclass}{inlineimage}
```

Set the actions of each of the key/value combinations for `\includegraphics`. Many are ignored.

If an optional width was given, set an HTML style:

```
\define@key{igraph}{width}{%
\setlength{\LWR@igwidth}{#1}%
\ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \uselengthunit{PT}%
    \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
}
```

If an optional height was given, set an HTML style:

```
\define@key{igraph}{height}{%
\setlength{\LWR@igheight}{#1}%
\ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \uselengthunit{PT}%
    \renewcommand*{\LWR@igheightstyle}{%
    height:\rndprintlength{\LWR@igheight} %
    }%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
}
```

Handle origin key:

```
\define@key{igraph}{origin}{%
```

```
\renewcommand*{\LWR@igorigin}{#1}%
}
```

Handle angle key:

```
\define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
\define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}

```

It appears that graphicx does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
\define@key{igraph}{scale}{%
\renewcommand*{\LWR@igxscale}{#1}%
\renewcommand*{\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
\define@key{igraph}{bb}{}
\define@key{igraph}{bbllx}{}
\define@key{igraph}{bblly}{}
\define@key{igraph}{bburx}{}
\define@key{igraph}{bbury}{}
\define@key{igraph}{natwidth}{}
\define@key{igraph}{natheight}{}
\define@key{igraph}{hiresbb}{}
\define@key{igraph}{viewport}{}
\define@key{igraph}{trim}{}
\define@key{igraph}{totalheight}{}
\define@key{igraph}{keepaspectratio}{}
\define@key{igraph}{clip}{}
\define@key{igraph}{draft}{}
\define@key{igraph}{type}{}
\define@key{igraph}{ext}{}
\define@key{igraph}{read}{}
\define@key{igraph}{command}{}
```

§ 172.3 Printing HTML styles

`\LWR@rotstyle` {⟨*prefix*⟩} {⟨*degrees*⟩}

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
\newcommand*{\LWR@rotstyle}[2]{%
 #1transform:rotate(-#2deg);
}
```

`\LWR@scalestyle` {⟨*prefix*⟩} {⟨*xscale*⟩} {⟨*yscale*⟩}

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```
\newcommand*{\LWR@scalestyle}[3]{%
 #1transform:scale(#2,#3);
}
```

§ 172.4 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
\newbool{LWR@infloatrow}
\boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
\def\LWR@opacity{1}
```

Used to determine the actual image size if needed:

```
\newsavebox{\LWR@imagesizebox}
```

```
\let\LWR@origGin@setfile\Gin@setfile
```

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
\AtBeginDocument{
\define@key{Gin}{class}{}
```

```
}
```

`\LWR@includegraphicsb` * [⟨*2: options*⟩] [⟨*3: options*⟩] {⟨*4: filename*⟩}

graphics syntax is `\includegraphics` * [⟨*llx,lly*⟩] [⟨*urx,ury*⟩] {⟨*file*⟩}

graphicx syntax is `\includegraphics` [⟨*key values*⟩] {⟨*file*⟩}

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
\NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
{%
\LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
\LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
{%
    \ifbool{LWR@infloatrow}%
    {}
    {% not in a minipage or a floatrow:
        \setlength{\linewidth}{6in}%
        \setlength{\textwidth}{6in}%
        \setlength{\textheight}{9in}%
    }%
}{}%

\begingroup%
\renewcommand*{\Gin@setfile}[3]{%
\LWR@traceinfo{Gin@setfile ##3}%
\xdef\LWR@parsedfilename{##3}%
}%
\Ginclude@graphics{\detokenize\expandafter{#4}}%
\endgroup%
\filename@parse{\LWR@parsedfilename}%
\LWR@traceinfo{LWR@parsedfilename is \LWR@parsedfilename}%
% \LWR@sanitize{\LWR@parsedfilename}%
```

For correct em sizing during the width and height conversions:

```
\large%
```

Reset some defaults, possibly will be changed below if options were given:

```
\setlength{\LWR@igwidth}{0pt}%
\setlength{\LWR@igheight}{0pt}%
\renewcommand*{\LWR@igwidthstyle}{}%
\renewcommand*{\LWR@igheightstyle}{}%
\renewcommand*{\LWR@igorigin}{}%
\renewcommand*{\LWR@igangle}{}%
\renewcommand*{\LWR@igxscale}{1}%
\renewcommand*{\LWR@igyscale}{1}%
\renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```
\IfValueF{#3}{%
\IfValueTF{#2}%
{\setkeys{igraph}{#2}}%
{\setkeys{igraph}{}}%
}%
```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```
\ifbool{FormatWP}{%
    \begingroup%
    \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
    \define@key{Gin}{angle}{}%
    \IfBooleanTF{#1}%
    {% starred
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*{#4}}%
            }%
        }%
    }% starred
    {% not starred
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
```

```
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics{#4}}%
            }%
        }%
    }% not starred
    \endgroup%
    \uselengthunit{PT}%
    \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
    \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igheightstyle}{height:\rndprintlength{\LWR@igheight}}%
}{}%
```

Create the HTML reference with the graphicspath, filename, extension, alt tag, style, and class.

The `\LWR@origtilde` adds space between tags in case this is being done inside a `\savebox` where `\newline` has no effect.

```
\LWR@traceinfo{LWR@includegraphicsb: about to create href}%
\href{\LWR@parsedfilename}%
{% start of href
\LWR@traceinfo{LWR@includegraphicsb: about to LWR@htmltag}%
\LWR@htmltag{% start of image tags
% img src="\LWR@sanitized" \LWR@orignewline%
img src="%
\begingroup\@sanitize\LWR@parsedfilename\endgroup%
" \LWR@orignewline%
% \LWR@origtilde{} alt="[\filename@base]" \LWR@orignewline%
```

Only include a style tag if a width, height, angle, or scale was given:

```
\ifthenelse{
    \NOT\equal{\LWR@igwidthstyle}{} \OR
    \NOT\equal{\LWR@igheightstyle}{} \OR
    \NOT\equal{\LWR@igorigin}{} \OR
    \NOT\equal{\LWR@igangle}{} \OR
    \NOT\equal{\LWR@igxscale}{1} \OR
    \NOT\equal{\LWR@igyscale}{1}
}%
{\LWR@origtilde{} style="%
\ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
{\LWR@igwidthstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
{\LWR@igheightstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
{\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
```

```
\ifthenelse{\NOT\equal{\LWR@igangle}{}}%
{%
\LWR@rotstyle{-ms-}{\LWR@igangle}%
\LWR@rotstyle{-webkit-}{\LWR@igangle}%
\LWR@rotstyle{}{\LWR@igangle%
}}{}%
\ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
\NOT\equal{\LWR@igyscale}{1}}%
{\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
\LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
\LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}{}%
%
\ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
{opacity:\LWR@opacity;}%
{}%
%
" \LWR@orignewline}{}%
```

Set the class:

```
\LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
}% end of image tags
}% end of href
```

Return to small-sized output:

```
\LWR@origscriptsize
\LWR@traceinfo{LWR@includegraphicsb done}%
}
```

`\includegraphics` `[`⟨*key=val*⟩`]` `{`⟨*filename*⟩`}`

Handles width and height, converted to fixed width and heights.

Converts any `.pdf` references to `.svg` for HTML

The user should always refer to `.pdf` in the document source.

```
\AtBeginDocument{

\LWR@traceinfo{Patching includegraphics.}

\LetLtxMacro\LWR@origincludegraphics\includegraphics

\renewcommand*{\includegraphics}
{%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
\LWR@traceinfo{includegraphics}%
\LWR@ensuredoingapar%
\LWR@includegraphicsb%
}% includegraphics
}% AtBeginDocument
```

§ 172.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```
\newcommand*{\LWR@rotboxorigin}{}
```

`\LWR@originname` {⟨*letter*⟩}

Given one LaTeX origin key value, translate into an HTML origin word:

```
\newcommand*{\LWR@originname}[1]{%
\ifthenelse{\equal{#1}{t}}{top}{}%
\ifthenelse{\equal{#1}{b}}{bottom}{}%
\ifthenelse{\equal{#1}{c}}{center}{}%
\ifthenelse{\equal{#1}{l}}{left}{}%
\ifthenelse{\equal{#1}{r}}{right}{}%
}
```

`\LWR@originnames` {⟨*letters*⟩}

Given one- or two-letter LaTeX origin key values, translate into HTML origin words:

```
\newcommand*{\LWR@originnames}[1]{%
\StrChar{#1}{1}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
\StrChar{#1}{2}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
}
```

Handle the origin key for `\rotatebox`:

```
\define@key{krotbox}{origin}{%
\renewcommand*{\LWR@rotboxorigin}{#1}%
}
```

These keys are ignored:

```
\define@key{krotbox}{x}{}
\define@key{krotbox}{y}{}
\define@key{krotbox}{units}{}
```

`\rotatebox` `[`⟨*keyval list*⟩`]` `{`⟨*angle*⟩`}` `{`⟨*text*⟩`}`

```
\LetLtxMacro\LWR@origrotatebox\rotatebox

\AtBeginDocument{
\RenewDocumentCommand{\rotatebox}{O{} m +m}{%
```

Reset the origin to "none-given":

```
\renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
\setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
\ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
{transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
\LWR@rotstyle{-ms-}{#2} %
\LWR@rotstyle{-webkit-}{#2} %
\LWR@rotstyle{}{#2} %
"}\LWR@orignewline%
```

Print the text to be rotated:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
}% AtBeginDocument
```

\scalebox {⟨*h-scale*⟩} [⟨*v-scale*⟩] {⟨*text*⟩}

```
\LetLtxMacro\LWR@origscalebox\scalebox

\AtBeginDocument{
\RenewDocumentCommand{\scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
\LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
\LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
\LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
"}%
```

Print the text to be scaled:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
}% AtBeginDocument
```

\reflectbox {⟨*text*⟩}

```
\let\LWR@origreflectbox\reflectbox

\AtBeginDocument{
\renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
}
```

\resizebox {⟨*h-length*⟩} {⟨*v-length*⟩} {⟨*text*⟩}

Simply prints its text argument.

```
\LetLtxMacro\LWR@origresizebox\resizebox

\AtBeginDocument{
\renewcommand{\resizebox}[3]{#3}
}
```

File 88 `lwarp-graphicx.sty`

§ 173 Package graphicx

Pkg graphicx graphicx is emulated.

graphicx loads graphics, which also loads lwarp-graphics, which remembers the original graphics definitions for use inside a `lateximage`, and then patches them `\AtBeginDocument` for HTML output.

lwarp-graphics handles the syntax of either graphics or graphicx.

for HTML output:

```
1 \LWR@ProvidesPackagePass{graphicx}
```

File 89 `lwarp-grffile.sty`

§ 174 Package grffile

Pkg grffile grffile is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

⚠ matching PDF and SVG

lwarp-grffile now exists as a placeholder since grffile used to be emulated by lwarp, and thus older versions of lwarp-grffile may exist and should be overwritten by this newer version.

for HTML output:

```
1 \LWR@ProvidesPackagePass{grffile}
```

File 90 `lwarp-hang.sty`

§ 175 Package hang

(*Emulates or patches code by* Andreas Nolda.)

Pkg hang hang is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{hang}
2 \newlength{\hangingindent}
```

```
\setlength{\hangingindent}{1em}
\newlength{\hangingleftmargin}
\setlength{\hangingleftmargin}{0em}

\newcommand*{\LWR@findhangingleftmargin}{%
\setlength{\LWR@templengthone}{\hangingleftmargin}%
\addtolength{\LWR@templengthone}{\hangingindent}%
}

\newenvironment{hangingpar}
{
    \LWR@findhangingleftmargin%
    \uselengthunit{PT}%
    \BlockClass[%
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}%
    ]%
    {hangingpar}%
}
{\endBlockClass}

\newenvironment{hanginglist}
{%
    \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
    \renewcommand*{\LWR@printopenlist}{%
        \LWR@findhangingleftmargin%
        \uselengthunit{PT}%
        ul style="list-style-type:none; %
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}"%
    }%
    \let\item\LWR@itemizeitem%
    \list{}{}%
}
{\endlist}

\newenvironment{compacthang}
{\hanginglist}
{\endhanginglist}

\newlength{\labeledleftmargin}
\setlength{\labeledleftmargin}{0em}

\newenvironment{labeledpar}[2]
{%
    \BlockClass[%
        \LWR@findhangingleftmargin%
        \uselengthunit{PT}%
        margin-left:\rndprintlength{\LWR@templengthone}; %
        text-indent:-\rndprintlength{\hangingindent}%
```

```
    ]{labeledpar}#2%
}
{\endBlockClass}

\newenvironment{labeledlist}[1]
{\hanginglist}
{\endhanginglist}

\newenvironment{compactlabel}[1]
{\hanginglist}
{\endhanginglist}
```

File 91 `lwarp-hanging.sty`

§ 176 Package hanging

Pkg `hanging` hanging is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{hanging}

\@ifclassloaded{memoir}{
\let\hangpara\relax
\let\hangparas\relax
\let\endhangparas\relax
\let\hangpunct\relax
\let\endhangpunct\relax
}{}
```

`\hangpara` {⟨*indent*⟩} {⟨*afternum*⟩}

Use `hangparas` instead.

```
\newcommand*{\hangpara}[2]{}
```

Env `hangparas` {⟨*indent*⟩} {⟨*afternum*⟩}

```
\newenvironment*{hangparas}[2]
{%
    \uselengthunit{PT}%
    \BlockClass[%
        margin-left:\rndprintlength{#1}; %
        text-indent:-\rndprintlength{#1}%
    ]%
    {hangingpar}%
}
{\endBlockClass}
```

Env `hangpunct`

```
\newenvironment*{hangpunct}
{\BlockClass{hangpunct}}
{\endBlockClass}

\newcommand{\nhpt}{.}
\newcommand{\nhlq}{‘}
\newcommand{\nhrq}{’}
```

File 92 `lwarp-hyperref.sty`

§ 177 Package hyperref

(*Emulates or patches code by* Sebastian Rahtz, Heiko Oberdiek.)

Pkg `hyperref` hyperref is emulated.

for HTML output:

```
% \LWR@ProvidesPackageDrop{hyperref}
\typeout{Using the lwarp html version of package ‘hyperref’ -- discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘hyperref’.}
% \ProvidesPackage{lwarp-#1-#2}
\DeclareOption*{}
\ProcessOptions\relax

\newcommand*{\hypersetup}[1]{}
\newcommand*{\hyperbaseurl}[1]{}
```

`\hyperimage` {⟨*url*⟩} {⟨*alt text*⟩}

Insert an image with alt text:

```
\NewDocumentCommand{\hyperimage}{m +m}{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\LWR@htmltag{img src="\LWR@templink" alt="#2" class="hyperimage"}%
\LWR@ensuredoingapar%
}
```

`\hyperdef` {⟨*1: category*⟩} {⟨*2: name*⟩} {⟨*3: text*⟩}

Creates an HTML anchor to `category.name` with the given text.

```
\NewDocumentCommand{\hyperdef}{m m +m}{%
```

```
\LWR@ensuredoingapar%
\LWR@sublabel{#1.#2}%
#3%
}
```

`\LWR@hyperrefb` {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩}

Creates an HTML link to URL#category.name with the given text.

```
\NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\def\LWR@templinktwo{#2}%
\@onelevel@sanitize\LWR@templinktwo%
\def\LWR@templinkthree{#3}%
\@onelevel@sanitize\LWR@templinkthree%
\LWR@htmltag{a href="\LWR@templink\LWR@hashmark%
    \LWR@templinktwo.\LWR@templinkthree"%
}%
#4%
\LWR@htmltag{/a}%
}
```

`\LWR@hyperrefc` [⟨*label*⟩] {⟨*text*⟩}

Creates text as an HTML link to the LaTeX label.

```
\NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
\LWR@startref{#1}%
#2%
\LWR@htmltag{/a}%
}
```

`\hyperref` {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩} —or— [⟨*1: label*⟩] {⟨*2: text*⟩}

```
\DeclareRobustCommand*{\hyperref}{%
\LWR@ensuredoingapar%
\@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
}
```

`\hypertarget` {⟨*name*⟩} {⟨*text*⟩}

Creates an anchor to name with the given text.

```
\NewDocumentCommand{\hypertarget}{m +m}{%
\label{#1}%
#2%
}
```

`\hyperlink` {⟨*name*⟩} {⟨*text*⟩}

Creates a link to the anchor created by hypertarget, with the given link text.

Declared because also defined by memoir.

```
\DeclareDocumentCommand{\hyperlink}{m +m}{%
\hyperref[#1]{#2}%
}
```

`\autoref` * {⟨*label*⟩}

For HTML, `\cleveref` is used instead.

```
\NewDocumentCommand{\autoref}{s m}{%
\IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
}
```

`\autopageref` {⟨*label*⟩}

For HTML, `\cleveref` is used instead.

```
\NewDocumentCommand{\autopageref}{s m}{%
\IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
}
```

`\pdfstringdef` {⟨*macroname*⟩} {⟨*TEXstring*⟩}

```
\newcommand{\pdfstringdef}[2]{}
```

`\pdfbookmark` [⟨*level*⟩] {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\pdfbookmark}[3][]{}
```

`\currentpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\currentpdfbookmark}[2]{}
```

`\subpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\subpdfbookmark}[2]{}
```

`\belowpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\belowpdfbookmark}[2]{}
```

`\texorpdfstring` {⟨*TEXstring*⟩} {⟨*PDFstring*⟩}

```
\newcommand{\texorpdfstring}[2]{#2}
```

`\hypercalcbp` {⟨*dimen*⟩} From hyperref.

```
\def\hypercalcbp#1{%
\strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
}%
```

`\Acrobatmenu` {⟨*menuoption*⟩} {⟨*text*⟩}

```
\newcommand{\Acrobatmenu}[2]{}
```

`\TextField` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\TextField}[2][]{}
```

`\CheckBox` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\CheckBox}[2][]{}
```

`\ChoiceMenu` [⟨*parameters*⟩] {⟨*label*⟩} {⟨*choices*⟩}

```
\DeclareRobustCommand{\ChoiceMenu}[3][]{}
```

`\PushButton` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\PushButton}[2][]{}
```

`\Submit` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Submit}[2][]{}
```

`\Reset` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Reset}[2][]{}
```

`\Gauge` [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Gauge}[2][]{}
```

`\LayoutTextField` {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutTextField}[2]{}
```

`\LayoutChoiceField` {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutChoiceField}[2]{}
```

`\LayoutCheckField` {⟨*label*⟩} {⟨*field*⟩}

```
76 \newcommand*{\LayoutCheckField}[2]{}
```

`\MakeRadioField` {⟨*width*⟩} {⟨*height*⟩}

```
77 \newcommand*{\MakeRadioField}[2]{}
```

`\MakeCheckField` {⟨*width*⟩} {⟨*height*⟩}

```
78 \newcommand*{\MakeCheckField}[2]{}
```

`\MakeTextField` {⟨*width*⟩} {⟨*height*⟩}

```
79 \newcommand*{\MakeTextField}[2]{}
```

`\MakeChoiceField` {⟨*width*⟩} {⟨*height*⟩}

```
80 \newcommand*{\MakeChoiceField}[2]{}
```

`\MakeFieldButton` {⟨*text*⟩}

```
81 \newcommand{\MakeFieldButton}[1]{}
```

File 93 **`lwarp-hyperxmp.sty`**

§ 178 Package hyperxmp

Pkg `hyperxmp` Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 94 **`lwarp-idxlayout.sty`**

§ 179 Package idxlayout

(*Emulates or patches code by* Thomas Titz.)

Pkg `idxlayout` Emulated.

for HTML output: Discard all options for lwarp-idxlayout:

```
\LWR@ProvidesPackageDrop{idxlayout}

\newcommand{\LWR@indexprenote}{}

\renewcommand*{\printindex}
{
\LWR@startpars

\LWR@indexprenote

\LWR@origprintindex
}

\newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
\newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{}}

\newcommand{\idxlayout}[1]{}
\newcommand*{\indexfont}{}
\newcommand*{\indexjustific}{}
\newcommand*{\indexsubsdelim}{}
\newcommand*{\indexstheadcase}{}
```

File 95 `lwarp-ifoddpage.sty`

§ 180 Package ifoddpage

(*Emulates or patches code by* Martin Scharrer.)

Pkg `ifoddpage` ifoddpage is emulated.

for HTML output: Discard all options for lwarp-ifoddpage:

```
\LWR@ProvidesPackageDrop{ifoddpage}

\newif\ifoddpage

\newif\ifoddpageoroneside

\DeclareRobustCommand{\checkoddpage}{\oddpagetrue\oddpageoronesidetrue}

\def\oddpage@page{1}

\def\@ifoddpage{%
        \expandafter\@firstoftwo
}
```

```
13
14 \def\@ifoddpageoroneside{%
15         \expandafter\@firstoftwo
16 }
```

File 96 `lwarp-indentfirst.sty`

§ 181 Package indentfirst

Pkg `indentfirst` indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{indentfirst}
```

File 97 `lwarp-inputenc.sty`

§ 182 Package inputenc

Pkg `inputenc` Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{inputenc}
2 \LWR@loadbefore{inputenc}
```

File 98 `lwarp-keyfloat.sty`

§ 183 Package keyfloat

(*Emulates or patches code by* Brian Dunn.)

Pkg `keyfloat` keyfloat is supported with minor adjustments.

⚠ `keywrap` If placing a `\keyfig`[H] inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

for HTML output:

```
1 \LWR@ProvidesPackagePass{keyfloat}
```

After keyfloat has loaded:

```
\AtBeginDocument{

\let\KFLT@boxinner\relax
\let\endKFLT@boxinner\relax

\NewEnviron{KFLT@boxinner}
{%
\LWR@traceinfo{KFLT@boxinner}%
\LWR@stoppars%
\KFLT@frame{\BODY}%
\LWR@startpars%
\LWR@traceinfo{KFLT@boxinner: done}%
}


\DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
{%
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}

\DeclareDocumentEnvironment{marginfigure}{o}
  {\begin{KFLT@marginfloat}{figure}}
  {\end{KFLT@marginfloat}}

\DeclareDocumentEnvironment{margintable}{o}
  {\begin{KFLT@marginfloat}{table}}
  {\end{KFLT@marginfloat}}

\DeclareDocumentEnvironment{keywrap}{m +m}
{%
\LWR@ensuredoingapar%
\setlength{\LWR@templengthone}{#1}%
\uselengthunit{PT}%
\begin{LWR@BlockClassWP}{%
    float:right; width:\rndprintlength{\LWR@templengthone}; %
    margin:10pt%
}%
{%
    width:\rndprintlength{\LWR@templengthone}%
}%
{marginblock}%
\setlength{\linewidth}{.95\LWR@templengthone}%
#2%
```

```
\end{LWR@BlockClassWP}%
}
{%
}

}% AtBeginDocument
```

File 99 `lwarp-layout.sty`

§ 184 Package layout

(*Emulates or patches code by* Kent McPherson, Johannes Braams, Hideo Umeki.)

Pkg `layout` layout is emulated.

for HTML output: Discard all options for lwarp-layout:

```
\LWR@ProvidesPackageDrop{layout}

\NewDocumentCommand{\layout}{s}{}
```

File 100 `lwarp-letterspace.sty`

§ 185 Package letterspace

(*Emulates or patches code by* R Schlicht.)

Pkg `letterspace` letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
\LWR@ProvidesPackageDrop{letterspace}

\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
```

File 101 `lwarp-lettrine.sty`

§ 186 Package lettrine

(*Emulates or patches code by* Daniel Flipo.)

Pkg `lettrine` Emulated.

for HTML output: Discard all options for lwarp-lettrine:

```
\LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `<span>` of class `lettrine`, and the following text is in a `<span>` of class `lettrinetext`. `\lettrine [`⟨*keys*⟩`] {`⟨*letter*⟩`} {`⟨*additional text*⟩`}`

```
\DeclareDocumentCommand{\lettrine}{o m m}{%
\InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
}

\newcounter{DefaultLines}
\setcounter{DefaultLines}{2}
\newcounter{DefaultDepth}
\newcommand*{\DefaultOptionsFile}{\relax}
\newcommand*{\DefaultLoversize}{0}
\newcommand*{\DefaultLraise}{0}
\newcommand*{\DefaultLhang}{0}
\newdimen\DefaultFindent
\setlength{\DefaultFindent}{\z@}
\newdimen\DefaultNindent
\setlength{\DefaultNindent}{0.5em}
\newdimen\DefaultSlope
\setlength{\DefaultSlope}{\z@}
\newdimen\DiscardVskip
\setlength{\DiscardVskip}{0.2\p@}
\newif\ifLettrineImage
\newif\ifLettrineOnGrid
\newif\ifLettrineRealHeight

\newcommand*{\LettrineTextFont}{\scshape}
\newcommand*{\LettrineFontHook}{}
\newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
\newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

File 102 `lwarp-lips.sty`

§ 187 Package lips

(*Emulates or patches code by* Matt Swift.)

Pkg `lips` lips is emulated.

```
% \LWR@ProvidesPackageDrop{lips}
\PackageInfo{lwarp}{Using the lwarp version of package ‘lips’.}%
\ProvidesPackage{lwarp-lips}

\NewDocumentCommand{\Lips}{}{\textellipsis}

\NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}

\let\lips\Lips
\let\olips\lips

\DeclareOption*{}
\DeclareOption{mla}{
\let\lips\BracketedLips
}
\ProcessOptions\relax

\newcommand \LPNobreakList {}
```

File 103 `lwarp-listings.sty`

§ 188 Package listings

(*Emulates or patches code by* Carsten Heinz, Brooks Moses, Jobst Hoffmann.)

Pkg `listings` listings is supported with some limitations. Text formatting is not yet supported.

for HTML output:

```
\begin{warpHTML}

\LWR@ProvidesPackagePass{listings}
```

Patches to embed listings inside `pre` tags:

```
\let\LWR@origlst@Init\lst@Init
```

```
\let\LWR@origlst@DeInit\lst@DeInit

\let\LWR@origlsthkEveryPar\lsthk@EveryPar

\renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{lol}{#1}{#2}}
```

`\lst@Init` `{`⟨*backslash-processing*⟩`}` Done at the start of a listing.

```
\renewcommand{\lst@Init}[1]{%
```

First, perform the listings initialization:

```
\LWR@traceinfo{lst@Init}%
\renewcommand*{\@captype}{lstlisting}%
\LWR@origlst@Init{#1}%
\LWR@traceinfo{finished origlst@Init}%
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```
\LWR@traceinfo{About to create verbatim.}%
\let\lsthk@EveryPar\relax%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}%

\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so open a :

```
\ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
\fi%
}
```

`\lst@DeInit` Done at the end of a listing.

```
\renewcommand*{\lst@DeInit}{%
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```
\let\lsthk@EveryPar\relax%

\LWR@afterendverbatim%
\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so create the closing :

```
\ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
\fi%
```

Final listings deinit:

```
\LWR@origlst@DeInit%
}
```

`\lst@MakeCaption` `{⟨t/b⟩}`

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
\def\lst@MakeCaption#1{%
\LWR@traceinfo{MAKING CAPTION at #1}%
  \lst@ifdisplaystyle
\LWR@traceinfo{making a listings display caption}%
    \ifx #1t%
         \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
                                  \expandafter\refstepcounter
         \fi {lstlisting}%
\LWR@traceinfo{About to assign label: !\lst@label!}%
%         \ifx\lst@label\@empty\else
% \label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
        \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
        \global\let\lst@name\lst@arg \global\let\lstname\lst@name
        \lst@ifnolol\else
            \ifx\lst@@caption\@empty
                \ifx\lst@caption\@empty
                    \ifx\lst@intname\@empty \else \def\lst@temp{ }%
                    \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
\LWR@traceinfo{addcontents lst@name: -\lst@name-}%
%                             \addcontentsline{lol}{lstlisting}{\lst@name}
                    \fi\fi
                \fi
            \else
```

This would have to be modified for lwarp:

```
\LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
                  \addcontentsline{lol}{lstlisting}%
{\protect\numberline{\thelstlisting}%
{\protect\ignorespaces \lst@@caption \protect\relax}}%
            \fi
         \fi
     \fi
    \ifx\lst@caption\@empty\else
\LWR@traceinfo{lst@caption not empty-}%
        \lst@IfSubstring #1\lst@captionpos
            {\begingroup
```

```
\LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
%               \let\@@vskip\vskip
%               \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
%               \def\lst@vskip{\nobreak\@@vskip\@tempskipa\nobreak}%
%               \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
%               \ifx #1t\allowbreak \fi
              \ifx\lst@title\@empty
```

New lwarp code to create a caption:

```
                  \lst@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}
              \else
```

New lwarp code to create a title:

```
%                   \lst@maketitle\lst@title % (AS)
\LWR@traceinfo{Making title: \lst@title}%
\begin{BlockClass}{lstlistingtitle}% lwarp
\lst@maketitle\lst@title% lwarp
\end{BlockClass}% lwarp
              \fi
\LWR@traceinfo{About to assign label: !\lst@label!}%
        \ifx\lst@label\@empty\else
\leavevmode% gets rid of bad space factor error
\GetTitleStringExpand{\lst@caption}%
\edef\LWR@lntemp{\GetTitleStringResult}%
\edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
\label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
```

Not needed for lwarp:

```
%               \ifx #1b\allowbreak \fi
              \endgroup}{}%
    \fi
\LWR@traceinfo{end of making a listings display caption}%
  \else
\LWR@traceinfo{INLINE}%
  \fi
\LWR@traceinfo{DONE WITH CAPTION at #1}%
}
```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field `\VerbatimHTMLWidth` wide.

```
\lst@Key{numbers}{none}{%
    \let\lst@PlaceNumber\@empty
    \lstKV@SwitchCases{#1}%
    {none&\\%
     left&\def\lst@PlaceNumber{%
```

```
% \llap{
\LWR@orignormalfont%
\lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
% }
}
\\%
     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
                \kern\VerbatimHTMLWidth \kern\lst@numbersep
                \lst@numberstyle{\thelstnumber}}}%
    }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

\end{warpHTML}
```

File 104 `lwarp-longtable.sty`

§ 189 Package longtable

(*Emulates or patches code by* David Carlisle.)

Pkg `longtable` longtable is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{longtable}
```

⚠ Longtable `\endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid "Misplaced `\noalign`." when using `\begin`{warpprint}.

Keep the `\endfirsthead` row, which is still relevent to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside `\warpingprintonly`.

See:
http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable

Env `longtable` * [⟨*horizalignment*⟩] {⟨*colspec*⟩} Emulates the `longtable` environment.

Per the caption package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```
\newenvironment{longtable*}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\booltrue{LWR@starredlongtable}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

\newenvironment{longtable}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\refstepcounter{\LTcaptype}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

```

Provided for compatibility, but ignored:

```
\newcounter{LTchunksize}
\def\endhead{\LWR@tabularendofline}% throws away options //[dim] and //*
\def\endfirsthead{\LWR@tabularendofline}
\def\endfoot{\LWR@tabularendofline}
\def\endlastfoot{\LWR@tabularendofline}
\newcommand\tabularnewline{\LWR@tabularendofline}
\newcommand{\setlongtables}{}% Obsolete command, does nothing.
\newlength{\LTleft}
\newlength{\LTright}
\newlength{\LTpre}
\newlength{\LTpost}
\newlength{\LTcapwidth}

\renewcommand*{\kill}{\LWR@tabularendofline}
```

File 105 `lwarp-lscape.sty`

§ 190 Package lscape

(*Emulates or patches code by* D. P. Carlisle.)

Pkg `lscape` lscape is emulated.

for HTML output: Discard all options for lwarp-lscape.

```
\LWR@ProvidesPackageDrop{lscape}
\newenvironment*{landscape}{}{}
```

File 106 `lwarp-ltcaption.sty`

§ 191 Package ltcaption

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg `ltcaption` ltcaption is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{ltcaption}
```

`\LTcaptype` is already defined by lwarp.

`longtable*` is already defined by lwarp-longtable.

```
\newlength{\LTcapskip}
\newlength{\LTcapleft}
\newlength{\LTcapright}
\newcommand*{\LTcapmarginsfalse}{}
```

File 107 `lwarp-ltxtable.sty`

§ 192 Package ltxtable

Pkg `ltxtable` ltxtable is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{ltxtable}
```

\LTXtable {⟨*width*⟩} {⟨*file*⟩}

```
\newcommand*{\LTXtable}[2]{%
\input{#2}%
}
```

File 108 **lwarp-luatodonotes.sty**

§ 193 Package luatodonotes

(*Emulates or patches code by* Fabian Lipp.)

Pkg luatodonotes luatodonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

for HTML output:

```
\LWR@ProvidesPackagePass{luatodonotes}
```

Nullify options:

```
\@todonotes@additionalMarginEnabledfalse

\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}


\renewcommand{\@todonotes@drawMarginNoteWithLine}{%
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
```

```
        {\@todonotes@author:\,}%
        \fi%
        \@todonotes@text%
    }%
}

\newcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

\LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon

\RenewDocumentCommand{\@todocommon}{m m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\LWRTODONOTES@orig@todocommon{#1}{#2}%
\endgroup%
}

\renewcommand{\@todoarea}[3][]{%
    \@todonotes@areaselectedtrue%
    \@todocommon{#1}{#2}%
    \todonotes@textmark@highlight{#3}%
    \zref@label{@todonotes@\arabic{@todonotes@numberoftodonotes}@end}%
}%


\DeclareDocumentCommand{\todonotes@textmark@highlight}{m}{%
```

```
\InlineClass[background:\#B3FFB3]{highlight}{#1}%
}

\fi% \if@todonotes@disabled
```

File 109 `lwarp-marginfit.sty`

§ 194 Package marginfit

Pkg `marginfit` marginfit is ignored.

for HTML output: Discard all options for lwarp-marginfit:

```
\LWR@ProvidesPackageDrop{marginfit}
```

File 110 `lwarp-marginfix.sty`

§ 195 Package marginfix

(*Emulates or patches code by* Stephen Hicks.)

Pkg `marginfix` Emulated.

for HTML output: Discard all options for lwarp-marginfix:

```
\LWR@ProvidesPackageDrop{marginfix}

\newcommand*{\marginskip}[1]{}
\newcommand*{\clearmargin}{}
\newcommand*{\softclearmargin}{}
\newcommand*{\extendmargin}[1]{}
\newcommand*{\mparshift}[1]{}
\newdimen\marginheightadjustment
\newdimen\marginposadjustment
\newcommand*{\blockmargin}[1][]{}
\newcommand*{\unblockmargin}[1][]{}
\newcommand*{\marginphantom}[2][]{}
```

File 111 `lwarp-marginnote.sty`

§ 196 Package marginnote

(*Emulates or patches code by* Markus Kohm.)

Pkg `marginnote` Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```
\LWR@ProvidesPackageDrop{marginnote}

\NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
\newcommand*{\marginnoteleftadjust}{}
\newcommand*{\marginnoterightadjust}{}
\newcommand*{\marginnotetextwidth}{}
\let\marginnotetextwidth\textwidth
\newcommand*{\marginnotevadjust}{}
\newcommand*{\marginfont}{}
\newcommand*{\raggedleftmarginnote}{}
\newcommand*{\raggedrightmarginnote}{}
```

File 112 `lwarp-mcaption.sty`

§ 197 Package mcaption

(*Emulates or patches code by* Stephan Hennig.)

Pkg `mcaption` mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```
\LWR@ProvidesPackageDrop{mcaption}

\newenvironment{margincap}{}{}
\newcommand*{\margincapalign}{}
\newlength{\margincapsep}
```

File 113 `lwarp-mdframed.sty`

§ 198 Package mdframed

(*Emulates or patches code by* Marco Daniel, Elke Schubert.)

Pkg `mdframed` mdframed is loaded with options forced to `framemethod=none`.

§ 198.1 Limitations

support Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for mdframed environments and frame titles.

⚠ loading When used, lwarp loads mdframed in HTML with `framemethod=none`.

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the mdframed source). Since lwarp does not support `\bfseries` and friends, only one font selection may be made at a time.

theoremtitlefont `theoremtitlefont` is not supported, since the following text is not in braces in the mdframed source.

footnotes Footnotes are currently placed at the bottom of the HTML page.

ignored options `userdefinedwidth` and `align` are currently ignored.

CSS classes Environments created or encapsulated by mdframed are enclosed in a `<div>` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `<span>` of class `mdframedtitle`. Subtitles are in a `<span>` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

§ 198.2 Package loading

for HTML output:

```
\RequirePackage{xcolor}% for \convertcolorspec
\LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before mdframed

```
\LWR@origRequirePackage{amsthm}
```

Do not require Ti*k*z or pstricks:

```
\LWR@origRequirePackage[framemethod=none]{mdframed}
```

§ 198.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
\AtBeginDocument{
\def\mdf@trivlist#1{%
  \edef\mdf@temp{%
%     \topsep=\the\topsep\relax%
%     \partopsep=\the\partopsep\relax%
%     \parsep=\the\parsep\relax%
  }%
%   \setlength{\topsep}{#1}%
%   \topskip\z@%
%   \partopsep\z@%
%   \parsep\z@%
%   \@nmbrlistfalse%
%   \@trivlist%
%   \labelwidth\z@%
%   \leftmargin\z@%
%   \itemindent\z@%
  \let\@itemlabel\@empty%
  \def\makelabel##1{##1}%
%   \item\relax\mdf@temp\relax%
}

\renewcommand*{\endmdf@trivlist}{%
\LWR@traceinfo{endmdf@trivlist}%
% \endtrivlist%
\LWR@listend%
}
}% AtBeginDocument
```

§ 198.4 Initial setup

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
\mdfsetup{
startcode={\LWR@mdframedstart\LWR@origraggedright},
endcode={\LWR@mdframedend},
startinnercode={\LWR@startpars\LWR@origraggedright},
endinnercode={\LWR@stoppars},
}
```

§ 198.5 Color and length HTML conversion

`\LWR@mdfprintcolor` `{`⟨*mdfcolorkey*⟩`}`

Given the mdframed key, print the color.

```
\newcommand*{\LWR@mdfprintcolor}[1]{%
\convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
\#\LWR@tempcolor
}
```

`\LWR@mdfprintlength` `{`⟨*mdflengthkey*⟩`}`

Given the mdframed key, print the length.

```
\newcommand*{\LWR@mdfprintlength}[1]{%
\rndprintlength{\csuse{mdf@#1@length}}
}
```

§ 198.6 Environment encapsulation

`\LWR@mdframedstart` Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired `class`.

```
\newcommand*{\LWR@mdframedstart}{%
\LWR@traceinfo{LWR@mdframedstart start}%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
\LWR@stoppars%
```

Below, print HTML pt units:

```
\uselengthunit{PT}%
```

Open a `<div>` and with custom `class` and custom `style`:

```
\LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the background color:

```
background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
border: \LWR@mdfprintlength{linewidth} solid
\LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
\ifbool{mdf@shadow}{%
    box-shadow:
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintcolor{shadowcolor} ;
}
{box-shadow: none ;}
\LWR@orignewline

"}
% \LWR@htmldivclass{\LWR@mdthisenv}
```

`mdframed` environment may not work with the HTML versions of the following, so restore them to their originals while inside `mdframed`:

```
\LetLtxMacro{\hspace}{\LWR@orighspace}%
\LetLtxMacro\rule\LWR@origrule%
\LetLtxMacro\makebox\LWR@origmakebox%
\LWR@startpars%
\LWR@traceinfo{LWR@mdframedstart done}%
}
```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
\newcommand*{\LWR@mdframedend}{
\LWR@traceinfo{LWR@mdframedend start}%
```

Close the custom `<div>`:

```
\LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
\gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
\LWR@startpars%
\LWR@traceinfo{LWR@mdframedend done}%
}
```

§ 198.7 Mdframed environment

```
\renewenvironment{mdframed}[1][]{%
 \color@begingroup%
   \mdfsetup{userdefinedwidth=\linewidth,#1}%
   \mdf@startcode%
   \mdf@preenvsetting%
   \ifdefempty{\mdf@firstframetitle}{}%
            {\let\mdf@frametitlesave\mdf@frametitle%
             \let\mdf@frametitle\mdf@firstframetitle%
            }%
   \ifvmode\nointerlineskip\fi%
        \ifdefempty{\mdf@frametitle}{}%
            {\mdfframedtitleenv{\mdf@frametitle}%
%             \mdf@@frametitle@use%
            }%
   \mdf@trivlist{\mdf@skipabove@length}%%
   \mdf@settings%
%    \mdf@lrbox{\mdf@splitbox@one}%
%    \mdf@startinnercode%
  }%
  {%
%    \mdf@@ignorelastdescenders%
   \par%
%     \unskip\ifvmode\nointerlineskip\hrule \@height\z@ \@width\hsize\fi%%
   \ifmdf@footnoteinside%
      \def\mdf@reserveda{%
        \mdf@footnoteoutput%
%         \mdf@endinnercode%
%         \endmdf@lrbox%
%         \ifdefempty{\mdf@frametitle}{}%
%             {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
%         \detected@mdf@put@frame
      }%
   \else%
      \def\mdf@reserveda{%
%         \mdf@endinnercode%
%         \endmdf@lrbox%
%         \ifdefempty{\mdf@frametitle}{}%
%             {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
%         \detected@mdf@put@frame%
        \mdf@footnoteoutput%
        }%
   \fi%
   \mdf@reserveda%
  \aftergroup\endmdf@trivlist%
 \color@endgroup%
 \mdf@endcode%
}
```

\mdf@footnoteoutput

```
\renewrobustcmd*\mdf@footnoteoutput{%
    \LWR@printpendingmpfootnotes%
}
```

§ 198.8 Titles and subtitles

\mdfframedtitleenv {⟨*title*⟩}

Encapsulation of the original which places the title inside a of class mdframedtitle:

```
\LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv

\newlength{\LWR@titleroundcorner}

\renewrobustcmd\mdfframedtitleenv[1]{%
\LWR@traceinfo{LWR@mdframedtitleenv start}%
% \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
\uselengthunit{PT}%
```

Open a with a custom class and custom style:

```
\LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the title background color:

```
background:
\LWR@mdfprintcolor{frametitlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the title rule:

```
\ifbool{mdf@frametitlerule}{%
    border-bottom:
    \LWR@mdfprintlength{frametitlerulewidth}
    solid
    \LWR@mdfprintcolor{frametitlerulecolor}
    ; \LWR@orignewline
}{}%
```

The title's top border radius is adjusted for the line width:

```
border-radius:
\setlength{\LWR@titleroundcorner}
    {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
    \rndprintlength{\LWR@titleroundcorner}
    \rndprintlength{\LWR@titleroundcorner}
    0pt 0pt
    \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
" \LWR@orignewline
}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Print the title inside the span:

```
#1%
```

Closee the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
% }
\LWR@traceinfo{LWR@mdframedtitleenv end}%
}
```

`\LWR@mdfsubtitlecommon` {⟨*sub -or- subsub*⟩} [⟨*options*⟩] {⟨*title*⟩}

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `<span>` of class `mdframedsubtitle`:

```
\NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
{% the following empty line is required

\LWR@traceinfo{LWR@mdframedsubtitlecommon start}%
```

Special handling for mdframed: Subtitles have `\par`s around them, so temporarily disable them here.

```
\let\par\LWR@origpar%
```

Open a `<span>` with a custom `class` and custom `style`:

```
\LWR@htmltagc{span class="mdframed#1title"
style=" \LWR@orignewline
```

Convert and print the background color:

```
background:
\LWR@mdfprintcolor{#1titlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the above line:

```
\ifbool{mdf@#1titleaboveline}{%
    border-top:
    \LWR@mdfprintlength{#1titleabovelinewidth}
    solid
    \LWR@mdfprintcolor{#1titleabovelinecolor}
    ; \LWR@orignewline
}{}%
```

Convert and print the below line:

```
\ifbool{mdf@#1titlebelowline}{%
    border-bottom:
    \LWR@mdfprintlength{#1titlebelowlinewidth}
    solid
    \LWR@mdfprintcolor{#1titlebelowlinecolor}
    ; \LWR@orignewline
}{}%
```

Finish the custom style and the opening span tag:

```
"}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
\IfNoValueTF{#2}
{\csuse{LWR@origmdf#1title}{#3}}%
{\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}% the following empty line is required
\end{LWR@nestspan}% must follow the /span or an extra <p> appears

\LWR@traceinfo{LWR@mdframedsubtitlecommon end}%
}
```

\LWR@mdfsubtitle [⟨*options*⟩] {⟨*title*⟩}

```
\newcommand*{\LWR@mdfsubtitle}{%
\LWR@mdfsubtitlecommon{sub}%
}
\let\mdfsubtitle\LWR@mdfsubtitle
```

\LWR@mdfsubsubtitle [⟨*options*⟩] {⟨*title*⟩}

```
\newcommand*{\LWR@mdfsubsubtitle}{%
\LWR@mdfsubtitlecommon{subsub}%
}
\let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

§ 198.9 New environments

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```
\newcommand*{\LWR@mdthisenv}{mdframed}
```

`\newmdenv` `[`⟨*options*⟩`] {`⟨*env-name*⟩`}`

Modified from the original to remember the environment.

```
\renewrobustcmd*\newmdenv[2][]{%
\newenvironment{#2}%
{%
\mdfsetup{#1}%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}%
}
{\end{mdframed}}%
}
```

`\surroundwithmdframed` `[`⟨*options*⟩`] {`⟨*environment*⟩`}`

Modified from the original to remember the environment.

```
\renewrobustcmd*{\surroundwithmdframed}[2][]{%
\BeforeBeginEnvironment{#2}{%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{\end{mdframed}}%
}
```

`\mdtheorem` `[`⟨*mdframed-options*⟩`] {`⟨*envname*⟩`} [`⟨*numberedlike*⟩`] {`⟨*caption*⟩`} [`⟨*within*⟩`]`

Modified from the original to remember the environment.

```
\DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
 {\ifcsdef{#2}%
   {\mdf@PackageWarning{Environment #2 already exits\MessageBreak}}%
   {%
    \IfNoValueTF {#3}%
     {%#3 not given -- number relationship
      \IfNoValueTF {#5}%
        {%#3+#5 not given
        \@definecounter{#2}%
        \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#2}%
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                             \mdf@theoremspace%
```

```
                              \mdf@theoremtitlefont%
                              ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }%
          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
          {\end{mdframed}}%
        \newenvironment{#2*}[1][]{%
          \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
        }%
        {%#5 given -- reset counter
        \@definecounter{#2}\@newctr{#2}[#5]%
        \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
        \expandafter\xdef\csname the#2\endcsname{%
               \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
                  \@thmcounter{#2}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#2}%
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                              \mdf@theoremspace%
                              \mdf@theoremtitlefont%
                              ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }
          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
          {\end{mdframed}}%
        \newenvironment{#2*}[1][]{%
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                              \mdf@theoremspace%
                              \mdf@theoremtitlefont%
                              ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
        }%
      }%
      {%#3 given -- number relationship
        \global\@namedef{the#2}{\@nameuse{the#3}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#3}%
```

```
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                             \mdf@theoremspace%
                             \mdf@theoremtitlefont%
                             ##1}%
             \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
             }
          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
          {\end{mdframed}}%
        \newenvironment{#2*}[1][]{%
          \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
     }%
    \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
    \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
   }%
}
```

`\newmdtheoremenv` `[`⟨*mdframed-options*⟩`]` `{`⟨*envname*⟩`}` `[`⟨*numberedlike*⟩`]` `{`⟨*caption*⟩`}` `[`⟨*within*⟩`]`

Modified from the original to remember the environment.

```
\DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
    {\newtheorem{#2}{#4}}%
    {%
     \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
     \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
    }%
\BeforeBeginEnvironment{#2}{%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{%
\end{mdframed}}%
}
```

File 114 **lwarp-memhfixc.sty**

§ 199 Package memhfixc

Pkg `memhfixc` memhfixc is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{memhfixc}
```

File 115 `lwarp-metalogo.sty`

§ 200 Package metalogo

(*Emulates or patches code by* Andrew Gilbert Moschou.)

Pkg `metalogo` metalogo is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{metalogo}

\newcommand\setlogokern[2]{}
\newcommand\setlogodrop[2][XeTeX]{}
\newcommand\setLaTeXa[1]{}
\newcommand\setLaTeXee[1]{}
\newcommand\seteverylogo[1]{}
\newcommand\everylogo[1]{}
```

File 116 `lwarp-microtype.sty`

§ 201 Package microtype

(*Emulates or patches code by* R Schlicht.)

Pkg `microtype` microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```
\LWR@ProvidesPackageDrop{microtype}

\DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
\DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
\DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
\DeclareDocumentCommand{\SetProtrusion}{o m m}{}
\DeclareDocumentCommand{\SetExpansion}{o m m}{}
\DeclareDocumentCommand{\SetTracking}{o m m}{}
\DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
\DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
\DeclareDocumentCommand{\DisableLigatures}{o m}{}
\DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
\DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
\DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
```

```
\DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
\DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
\DeclareDocumentCommand{\microtypesetup}{m}{}
\DeclareDocumentCommand{\microtypecontext}{m}{}
\DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
\DeclareDocumentCommand{\lsstyle}{}{}
\DeclareDocumentCommand{\textls}{o +m}{}
\DeclareDocumentCommand{\lslig}{m}{#1}
}
\def\DeclareMicrotypeSet#1#{\@gobbletwo}
\def\DeclareMicrotypeVariants#1#{\@gobble}
\@onlypreamble\DeclareMicrotypeSet
\@onlypreamble\UseMicrotypeSet
\@onlypreamble\DeclareMicrotypeSetDefault
\@onlypreamble\DisableLigatures
\@onlypreamble\DeclareMicrotypeVariants
\@onlypreamble\DeclareMicrotypeBabelHook
```

File 117 `lwarp-midfloat.sty`

§ 202 Package midfloat

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `midfloat` midfloat is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{midfloat}

\newenvironment{strip}[1][]{}{}
\newskip\stripsep
```

File 118 `lwarp-midpage.sty`

§ 203 Package midpage

Pkg `midpage` midpage is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{midpage}

\newenvironment{midpage}
{\begin{BlockClass}[margin-top:6ex ; margin-bottom:6ex]{midpage}}
{\end{BlockClass}}
```

File 119 `lwarp-moreverb.sty`

§ 204 Package

moreverb

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `moreverb` moreverb is supported with some patches.

for HTML output:

```
\begin{warpHTML}

\LWR@ProvidesPackagePass{moreverb}

\BeforeBeginEnvironment{verbatimtab}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
}
\AfterEndEnvironment{verbatimtab}{%
\LWR@afterendverbatim%
}


\LetLtxMacro\LWRMV@orig@verbatimtabinput\@verbatimtabinput

\renewcommand{\@verbatimtabinput}[2][]{%
\LWR@forcenewpage
\LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
\LWRMV@orig@verbatimtabinput[#1]{#2}%
\LWR@afterendverbatim%
}

\BeforeBeginEnvironment{listing}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
}

\AfterEndEnvironment{listing}{%
% \unskip\LWR@origvspace*{-\baselineskip}%
\LWR@afterendverbatim%
}

\BeforeBeginEnvironment{listingcont}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
}

```

```
\AfterEndEnvironment{listingcont}{%
% \unskip\LWR@origvspace*{-\baselineskip}%
\LWR@afterendverbatim%
}

\LetLtxMacro\LWRMV@@listinginput\@listinginput

\renewcommand{\@listinginput}[3][]{
\LWR@forcenewpage
\LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
\LWRMV@@listinginput[#1]{#2}{#3}
\LWR@afterendverbatim%
}


\renewenvironment*{boxedverbatim}
{
\LWR@forcenewpage
\LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-2\baselineskip}%
\verbatim%
}
{
\endverbatim%
\unskip%
\LWR@afterendverbatim%
}


\end{warpHTML}
```

File 120 `lwarp-morewrites.sty`

§ 205 Package

morewrites

Pkg `morewrites`

Error if morewrites is loaded after lwarp.

Discard all options for lwarp-morewrites:

for HTML output:

```
\LWR@ProvidesPackageDrop{morewrites}

\LWR@loadbefore{morewrites}
```

File 121 `lwarp-mparhack.sty`

§ 206 Package mparhack

Pkg `mparhack` Ignored.

for HTML output: Discard all options for lwarp-mparhack:

```
\LWR@ProvidesPackageDrop{mparhack}
```

File 122 `lwarp-multicol.sty`

§ 207 Package multicol

(*Emulates or patches code by* Frank Mittelbach.)

Pkg `multicol` multicol is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{multicol}[2015/09/13]
```

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a `<div>` of class `multicolsheading`.

The content is placed inside a `<div>` of class `multicols`.

```
\begin{warpHTML}
```

Env `multicols` * {⟨*numcols*⟩} [⟨*heading*⟩]

```
\NewDocumentEnvironment{multicols}{s m o}
```

HTML `<div>` class to contain everything:

```
{
\LWR@forcenewpage
\BlockClass{multicols}
```

Optional HTML `<div>` class for the heading:

```
\IfValueT{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}}
```

When done with the environment, close the `<div>`:

```
{\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
\newcommand*{\columnbreak}{}
\newcommand*{\RLmulticolcolumns}{}
\newcommand*{\LRmulticolcolumns}{}

\newlength{\premulticols}
\newlength{\postmulticols}
\newlength{\multicolsep}
\newlength{\multicolbaselineskip}
\newlength{\multicoltolerance}
\newlength{\multicolpretolerance}
\newcommand*{\columnseprulecolor}{\normalcolor}
\newcounter{columnbadness}
\newcounter{finalcolumnbadness}
\newcounter{collectmore}
\newcounter{unbalance}
\newlength{\multicolovershoot}
\newlength{\multicolundershoot}

\end{warpHTML}
```

File 123 `lwarp-multirow.sty`

§ 208 Package

multirow

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `multirow` multirow is emulated during HTML output, and used as-is while inside a `lateximage`.

In a `lateximage`, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 62.17 for the print-mode versions.

for HTML output:

```
\LWR@ProvidesPackagePass{multirow}
```

Remember the print-mode version:

```
\LetLtxMacro\LWR@origmultirow\multirow
```

`\LWR@multirowborder` Set to `left` or `right` to create a thick border for the cell, for use by bigdelim:

```
\newcommand{\LWR@multirowborder}{}
```

§ 208.1 Multirow

\multirow [⟨*vpos*⟩] {⟨*numrows*⟩} [⟨*bigstruts*⟩] {⟨*width*⟩} [⟨*fixup*⟩] {⟨*text*⟩}

```
\RenewDocumentCommand{\multirow}{O{c} m o m o +m}%
{%
\LWR@traceinfo{*** multirow #1 #2 #4}%
\LWR@maybenewtablerow%
\LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
\LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
\IfValueT{#1}{%
\ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
\ifstrequal{#1}{t}{style="vertical-align:top" }{}%
}%
```

The left/right border, if given:

```
\ifdefvoid{\LWR@multirowborder}{}{%
style="border-\LWR@multirowborder: 2px dotted black ; %
padding-\LWR@multirowborder: 2px" %
}%
```

A class adds the column spec and the rule:

```
class="td%
```

Append this column's spec:

```
\StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add the vertical bar class.

```
\LWR@addcmidruletrim%
\LWR@addleftmostbartag%
\LWR@printbartag{\arabic{LWR@tablecolindex}}%
"%


\LWR@tdstartstyles%
\LWR@addcmidrulewidth%
\LWR@tdendstyles%
}%
```

The column's < spec:

```
\LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tablecolindex}}%
```

While printing the text, redefine \\ to generate a new line

```
\begingroup\LetLtxMacro{\\}{\LWR@endofline}#6\endgroup%
\LWR@stoppars%
\global\boolfalse{LWR@intabularmetadata}%
\renewcommand{\LWR@multirowborder}{}%
\LWR@traceinfo{*** multirow done}%
}%
```

§ 208.2 Combined multicolumn and multirow

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

\multicolumnrow {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

```
\NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```
\begingroup%
```

```
\LetLtxMacro{\\}{\LWR@endofline}%
\LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
\endgroup%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tablecolindex}{#1}%
\addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing `@` or `!` columns for this cell:

```
\booltrue{LWR@skipatbang}%
}
```

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\multirow\LWR@origmultirow%
\renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
}
```

File 124 `lwarp-nameref.sty`

§ 209 Package nameref

Pkg `nameref` nameref is emulated by lwarp.

for HTML output: Discard all options for lwarp-nameref:

```
\typeout{Using the lwarp html version of package ‘nameref’ -- discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘nameref’.}
\DeclareOption*{}
\ProcessOptions\relax
```

File 125 `lwarp-natbib.sty`

§ 210 Package natbib

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `natbib` natbib is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{natbib}
```

Replace math < and > with \textless and \textgreater:

A macro to compare:

```
\newcommand{\LWRNB@NAT@open}{$<$}
```

To patch \NAT@open and \NAT@close

```
\newcommand{\LWRNB@patchnatbibopenclose}{
\ifdefstrequal{\NAT@open}{\LWRNB@NAT@open}
{
    \renewcommand{\NAT@open}{\textless}
    \renewcommand{\NAT@close}{\textgreater}
}{}
}
```

Do it now in case angle was selected as an option:

```
\LWRNB@patchnatbibopenclose
```

Also patch \setcitestyle to patch after settings are made:

```
\let\LWRNB@origsetcitestyle\setcitestyle

\renewcommand{\setcitestyle}[1]{%
\LWRNB@origsetcitestyle{#1}%
\LWRNB@patchnatbibopenclose%
}
```

File 126 **lwarp-needspace.sty**

§ 211 Package

needspace

(*Emulates or patches code by* Peter Wilson.)

Pkg needspace needspace is not used during HTML conversion.

for HTML output: Discard all options for lwarp-needspace:

```
\LWR@ProvidesPackageDrop{needspace}

\DeclareDocumentCommand{\needspace}{m}{}
\DeclareDocumentCommand{\Needspace}{s m}{}
```

File 127 `lwarp-newclude.sty`

§ 212 Package newclude

Pkg `newclude` Error if newclude is loaded after lwarp.

Discard all options for lwarp-newclude:

for HTML output:

```
\LWR@ProvidesPackageDrop{newclude}
\LWR@loadbefore{newclude}
```

File 128 `lwarp-newunicodechar.sty`

§ 213 Package newunicodechar

Pkg `newunicodechar` Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

for HTML output:

```
\LWR@ProvidesPackageDrop{newunicodechar}
\LWR@loadbefore{newunicodechar}
```

File 129 `lwarp-nextpage.sty`

§ 214 Package nextpage

(*Emulates or patches code by* Peter Wilson.)

Pkg `nextpage` nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

```
\LWR@ProvidesPackageDrop{nextpage}

\DeclareDocumentCommand{\cleartoevenpage}{o}{}
\DeclareDocumentCommand{\movetoevenpage}{o}{}
\DeclareDocumentCommand{\cleartooddpage}{o}{}
\DeclareDocumentCommand{\movetooddpage}{o}{}
```

File 130 `lwarp-nonumonpart.sty`

§ 215 Package nonumonpart

Pkg `nonumonpart` nonumonpart is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nonumonpart}
```

File 131 `lwarp-nopageno.sty`

§ 216 Package nopageno

Pkg `nopageno` nopageno is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nopageno}
```

File 132 `lwarp-nowidow.sty`

§ 217 Package nowidow

(*Emulates or patches code by* Raphaël Pinson.)

Pkg `nowidow` nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}
```

`\nowidow` [⟨*lines*⟩]

`\setnowidow` [⟨*lines*⟩]

```
2 \newcommand*{\nowidow}[1][]{}
3 \newcommand*{\setnowidow}[1][]{}
```

`\noclub` [⟨*lines*⟩]

`\setnoclub` [⟨*lines*⟩]

```
4 \newcommand*{\noclub}[1][]{}
5 \newcommand*{\setnoclub}[1][]{}
```

File 133 `lwarp-ntheorem.sty`

§ 218 Package

ntheorem

(*Emulates or patches code by* Wolfgang May, Andreas Schedler.)

Pkg `ntheorem` ntheorem is patched for use by lwarp.

Table 13: Ntheorem package — CSS styling of theorems and proofs

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `<span>` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

§ 218.1 Limitations

⚠ Font control This conversion is not total. Font control is via css, and the custom LaTeX font settings are ignored.

⚠ Equation numbering ntheorem has a bug with equation numbering in $\mathcal{AMS}$ environments when the option `thref` is used. lwarp does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarp's HTML output, but not with the print output. It is recommended to use cleveref instead of ntheorem's `thref` option.

§ 218.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ Disabled options The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because cleveref functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because lwarp emulated hyperref.

for HTML output: Some disabled options:

```
\DeclareOption{thref}{}

```

```
\newbool{LWR@ntheoremmarks}
\boolfalse{LWR@ntheoremmarks}

\DeclareOption{thmmarks}{
\booltrue{LWR@ntheoremmarks}
\newif\ifsetendmark\setendmarktrue
}


\newbool{LWR@ntheoremamsthm}
\boolfalse{LWR@ntheoremamsthm}

\DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}


\DeclareOption{amsmath}{}
\DeclareOption{hyperref}{}


\LWR@ProvidesPackagePass{ntheorem}
```

§ 218.3 Remembering the theorem style

Storage for the style being used for new theorems.

```
\newcommand{\LWR@newtheoremstyle}{plain}

\AtBeginDocument{
\@ifpackageloaded{cleveref}{
\gdef\@thm#1#2#3{%
   \if@thmmarks
     \stepcounter{end\InTheoType ctr}%
   \fi
   \renewcommand{\InTheoType}{#1}%
   \if@thmmarks
     \stepcounter{curr#1ctr}%
     \setcounter{end#1ctr}{0}%
   \fi
   \refstepcounter[#1]{#2}%  <<< cleveref modification
   \theorem@prework
    \LWR@forcenewpage% new
    \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
   \ifuse@newframeskips  % cf. latex.ltx for topsepadd: \@trivlist
     \ifthm@inframe
       \thm@topsep\theoreminframepreskipamount
       \thm@topsepadd\theoreminframepostskipamount
      \else
       \thm@topsep\theorempreskipamount
```

```
      \thm@topsepadd\theorempostskipamount
    \fi
  \else% oldframeskips
    \thm@topsep\theorempreskipamount
    \thm@topsepadd \theorempostskipamount
    \ifvmode\advance\thm@topsepadd\partopsep\fi
 \fi
 \@topsep\thm@topsep
 \@topsepadd\thm@topsepadd
 \advance\linewidth -\theorem@indent
 \advance\linewidth -\theorem@rightindent
 \advance\@totalleftmargin \theorem@indent
 \parshape \@ne \@totalleftmargin \linewidth
 \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
}
}{% not @ifpackageloaded{cleveref}
\gdef\@thm#1#2#3{%
 \if@thmmarks
   \stepcounter{end\InTheoType ctr}%
 \fi
 \renewcommand{\InTheoType}{#1}%
 \if@thmmarks
   \stepcounter{curr#1ctr}%
   \setcounter{end#1ctr}{0}%
 \fi
 \refstepcounter{#2}%
 \theorem@prework
  \LWR@forcenewpage% new
  \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
 \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
 \ifuse@newframeskips  % cf. latex.ltx for topsepadd: \@trivlist
   \ifthm@inframe
     \thm@topsep\theoreminframepreskipamount
     \thm@topsepadd\theoreminframepostskipamount
    \else
     \thm@topsep\theorempreskipamount
     \thm@topsepadd\theorempostskipamount
    \fi
  \else% oldframeskips
    \thm@topsep\theorempreskipamount
    \thm@topsepadd \theorempostskipamount
    \ifvmode\advance\thm@topsepadd\partopsep\fi
 \fi
 \@topsep\thm@topsep
 \@topsepadd\thm@topsepadd
 \advance\linewidth -\theorem@indent
 \advance\linewidth -\theorem@rightindent
 \advance\@totalleftmargin \theorem@indent
 \parshape \@ne \@totalleftmargin \linewidth
 \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
```

```
}
}
}% AtBeginDocument
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
   \@ifundefined{th@#1}{\@warning
          {Unknown theoremstyle `#1'. Using `plain'}%
          \theorem@style{plain}
            \renewcommand{\LWR@newtheoremstyle}{plain}% new
            }%
      {
        \theorem@style{#1}
        \renewcommand{\LWR@newtheoremstyle}{#1}% new
        }
}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

\gdef\@xnthm#1#2[#3]{%
  \ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \expandafter\@ifundefined{c@#1}%
        {\@definecounter{#1}}{}%
     \@newctr{#1}[#3]%
     \expandafter\xdef\csname the#1\endcsname{%
       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
          {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi
}

\gdef\@ynthm#1#2{%
  \ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \expandafter\@ifundefined{c@#1}%
        {\@definecounter{#1}}{}%
     \expandafter\xdef\csname the#1\endcsname
        {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
```

```
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi
}

\gdef\@othm#1[#2]#3{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}%
   {\ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
     \global\@namedef{the#1}{\@nameuse{the#2}}%
     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
              \noexpand\@num@addtheoremline{#1}{#3}}%
     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
              \noexpand\@nonum@addtheoremline{#1}{#3}}%
    \theoremkeyword{#3}%
    \expandafter\protected@xdef\csname #1Keyword\endcsname
             {\the\theoremkeyword}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
                \@thm{#1}{#2}{#3}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  \fi}
}
```

§ 218.4 HTML cross-referencing

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```
\newcommand{\LWR@inctheorem}{%
\addtocounter{LWR@thisautoid}{1}%
\LWR@stoppars%
\LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"}\LWR@htmltag{/a}%
\LWR@startpars%
}
```

§ 218.5 `\newtheoremstyle`

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```
\gdef\newtheoremstyle#1#2#3{%
  \expandafter\@ifundefined{th@#1}%
   {\expandafter\gdef\csname th@#1\endcsname{%
    \def\@begintheorem####1####2{%
    \LWR@inctheorem% new
    #2}%
    \def\@opargbegintheorem####1####2####3{%
    \LWR@inctheorem% new
    #3}%
}%
}%
{\PackageError{\basename}{Theorem style #1 already defined}\@eha}
}
```

§ 218.6 Standard styles

```
\renewtheoremstyle{plain}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}

\renewtheoremstyle{break}%
  {\item[
    \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
    ]}%
  {\item[
    \InlineClass{theoremheaderbreak}%
        {##1\ ##2\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{change}%
  {\item[
    \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{changebreak}%
  {\item[
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\theorem@separator}\newline
    ]}%
  {\item[
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\ (##3)\theorem@separator}\newline
    ]}

```

```
\renewtheoremstyle{margin}%
  {\item[
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\theorem@separator}
    ]}%
  {\item[
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\ (##3)\theorem@separator}
    ]}

\renewtheoremstyle{marginbreak}%
  {\item[
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\theorem@separator}\newline
    ]}%
  {\item[
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{nonumberplain}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{nonumberbreak}%
  {\item[
    \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
    ]}%
  {\item[
    \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{empty}%
  {\item[]}%
  {\item[
    \InlineClass{theoremheaderplain}{##3}]}

\renewtheoremstyle{emptybreak}%
  {\item[]}%
  {\item[
    \InlineClass{theoremheaderplain}{##3}] \ \newline}
```

§ 218.7 Additional objects

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```
\ifbool{LWR@ntheoremamsthm}{}{%
```

Upright text via CSS:

```
    \newtheoremstyle{plainupright}%
    {\item[
        \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
    {\item[
        \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
```

Upright text and small caps header via CSS:

```
    \newtheoremstyle{nonumberplainuprightsc}%
    {\item[
        \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
    {\item[
        \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}
}% not amsthm
```

§ 218.8 Renewed standard configuration

The following standard configuration is renewed using the new CSS:

```
\ifbool{LWR@ntheoremamsthm}{}{%
    \theoremnumbering{arabic}
    \theoremstyle{plain}
    \RequirePackage{latexsym}
    \theoremsymbol{\Box}
    \theorembodyfont{\itshape}
    \theoremheaderfont{\normalfont\bfseries}
    \theoremseparator{}
    \renewtheorem{Theorem}{Theorem}
    \renewtheorem{theorem}{Theorem}
    \renewtheorem{Satz}{Satz}
    \renewtheorem{satz}{Satz}
    \renewtheorem{Proposition}{Proposition}
    \renewtheorem{proposition}{Proposition}
    \renewtheorem{Lemma}{Lemma}
    \renewtheorem{lemma}{Lemma}
    \renewtheorem{Korollar}{Korollar}
    \renewtheorem{korollar}{Korollar}
    \renewtheorem{Corollary}{Corollary}
    \renewtheorem{corollary}{Corollary}

    \theoremstyle{plainupright}
    \theorembodyfont{\upshape}
    \theoremsymbol{\HTMLunicode{25A1}}% UTF-8 white box
    \renewtheorem{Example}{Example}
    \renewtheorem{example}{Example}
    \renewtheorem{Beispiel}{Beispiel}
    \renewtheorem{beispiel}{Beispiel}
```

```
    \renewtheorem{Bemerkung}{Bemerkung}
    \renewtheorem{bemerkung}{Bemerkung}
    \renewtheorem{Anmerkung}{Anmerkung}
    \renewtheorem{anmerkung}{Anmerkung}
    \renewtheorem{Remark}{Remark}
    \renewtheorem{remark}{Remark}
    \renewtheorem{Definition}{Definition}
    \renewtheorem{definition}{Definition}

    \theoremstyle{nonumberplainuprightsc}
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \renewtheorem{Proof}{Proof}
    \renewtheorem{proof}{Proof}
    \renewtheorem{Beweis}{Beweis}
    \renewtheorem{beweis}{Beweis}
    \qedsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof

    \theoremsymbol{}
}% not amsthm
```

§ 218.9 `amsthm` option

Only if the `amsthm` option was given:

```
\ifbool{LWR@ntheoremamsthm}{

\gdef\th@plain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderplain}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberplain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderplain}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
```

```
  \InlineClass{theoremheaderplain}{##1\ (##3).}
        ]}}

\gdef\th@definition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderdefinition}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberdefinition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderdefinition}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderdefinition}{##1\ (##3).}
        ]}}

\gdef\th@remark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderremark}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberremark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderremark}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
```

```
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderremark}{##1\ (##3).}
        ]}}

\gdef\th@proof{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% new
      \item[
  \InlineClass{theoremheaderproof}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% new
     \item[
  \InlineClass{theoremheaderproof}{##1\ (##3).}
        ]}}



\newcounter{proof}%
\if@thmmarks
    \newcounter{currproofctr}%
    \newcounter{endproofctr}%
\fi

\gdef\proofSymbol{\openbox}

\newcommand{\proofname}{Proof}

\newenvironment{proof}[1][\proofname]{
    \th@proof
    \def\theorem@headerfont{\itshape}%
    \normalfont
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \@thm{proof}{proof}{#1}
}%
{\@endtheorem}

}{}% amsthm option
```

§ 218.10 Ending a theorem

Patched for css:

```
\let\LWR@origendtheorem\@endtheorem
\renewcommand{\@endtheorem}{%
\ifbool{LWR@ntheoremmarks}{%
    \ifsetendmark%
    \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
    \setendmarkfalse%
    \fi%
}{}%
\LWR@origendtheorem% also does \@endtrivlist
\ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{}%
\endBlockClass%
}
```

§ 218.11 \NoEndMark

```
\gdef\NoEndMark{\global\setendmarkfalse}
```

§ 218.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

`\thm@thmline` {⟨*1: printed type*⟩} {⟨*2: #*⟩} {⟨*3: optional*⟩} {⟨*4: page*⟩}

```
\renewcommand{\thm@@thmline@noname}[4]{%
\hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
}

\renewcommand{\thm@@thmline@name}[4]{%
\hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
}
```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```
\def\thm@@thmline{\thm@@thmline@name}
```

Patch for css:

```
\def\listtheorems#1{
\LWR@htmlelementclass{nav}{lothm}%
\begingroup
\c@tocdepth=-2%
\def\thm@list{#1}\thm@processlist
\endgroup
\LWR@htmlelementclassend{nav}{lothm}%
}
```

§ 218.13 Symbols

Proof QED symbol:

```
\newcommand{\qed}{\qquad\the\qedsymbol}

\AtBeginDocument{
\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
}
```

§ 218.14 Cross-referencing

`\thref {`⟨*label*⟩`}`

```
\newcommand*{\thref}[1]{\cref{#1}}
```

File 134 `lwarp-overpic.sty`

§ 219 Package overpic

(*Emulates or patches code by* Rolf Niepraschk.)

Pkg `overpic` overpic is patched for use by lwarp.

⚠ scaling The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

See section 74.2 for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

for HTML output:

```
\LWR@ProvidesPackagePass{overpic}

\newcommand*{\overpicfontsize}{12}
\newcommand*{\overpicfontskip}{14}

\BeforeBeginEnvironment{overpic}{%
    \begin{lateximage}%
    \fontsize{\overpicfontsize}{\overpicfontskip}%
    \selectfont%
}

\AfterEndEnvironment{overpic}{\end{lateximage}}

\BeforeBeginEnvironment{Overpic}{%
```

```
    \begin{lateximage}%
    \fontsize{\overpicfontsize}{\overpicfontskip}%
    \selectfont%
}

\AfterEndEnvironment{Overpic}{\end{lateximage}}
```

File 135 `lwarp-pagenote.sty`

§ 220 Package pagenote

Pkg `pagenote` pagenote works as-is, but the page option is disabled.

for HTML output:

```
\DeclareOption{page}{}
\LWR@ProvidesPackagePass{pagenote}
```

File 136 `lwarp-pagesel.sty`

§ 221 Package pagesel

Pkg `pagesel` pagesel is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{pagesel}
```

File 137 `lwarp-paralist.sty`

§ 222 Package paralist

(*Emulates or patches code by* Bernd Schandl.)

Pkg `paralist` paralist is supported with minor changes.

for HTML output:

```
\LWR@ProvidesPackagePass{paralist}
```

The compact environments are identical to the regular ones:

```
\AtBeginEnvironment{compactitem}{\LWR@itemizestart}
\AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
\AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
\AtEndEnvironment{compactitem}{\LWR@listend}
\AtEndEnvironment{compactenum}{\LWR@listend}
\AtEndEnvironment{compactdesc}{\LWR@listend}
```

For the inline environments, revert `\item` to its original print-mode version:

```
\AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\LWR@origitem}
```

Manual formatting of the description labels:

```
\def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

File 138 `lwarp-parskip.sty`

§ 223 Package parskip

Pkg `parskip` parskip is ignored.

for HTML output: Discard all options for lwarp-parskip.

```
\LWR@ProvidesPackageDrop{parskip}
```

File 139 `lwarp-pdflscape.sty`

§ 224 Package pdflscape

Pkg `pdflscape` pdflscape is ignored.

for HTML output: Discard all options for lwarp-pdflscape:

```
\LWR@ProvidesPackageDrop{pdflscape}
```

File 140 `lwarp-pdfsync.sty`

§ 225 Package pdfsync

(*Emulates or patches code by* J. Laurens.)

Pkg `pdfsync` Emulated.

for HTML output: Discard all options for lwarp-pdfsync:

```
\LWR@ProvidesPackageDrop{pdfsync}
```

```
\newcommand*{\pdfsync}{}
\newcommand*{\pdfsyncstart}{}
\newcommand*{\pdfsyncstop}{}
```

File 141 `lwarp-pfnote.sty`

§ 226 Package pfnote

Pkg `pfnote` pfnote is emulated.

⚠ pfnote numbers While emulating pfnote, lwarp is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. lwarp therefore uses continuous footnote numbering even for pfnote.

for HTML output:

```
\LWR@ProvidesPackageDrop{pfnote}
```

File 142 `lwarp-placeins.sty`

§ 227 Package placeins

(*Emulates or patches code by* Donald Arseneau.)

Pkg `placeins` placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output:

```
\LWR@ProvidesPackageDrop{placeins}
```

```
\newcommand*{\FloatBarrier}{}
```

File 143 `lwarp-prelim2e.sty`

§ 228 Package prelim2e

(*Emulates or patches code by* Martin Schröder.)

Pkg `prelim2e` Emulated.

for HTML output: Discard all options for lwarp-prelim2e:

```
\LWR@ProvidesPackageDrop{prelim2e}
```

```
\newcommand{\PrelimText}{}
\newcommand{\PrelimTextStyle}{}
\newcommand{\PrelimWords}{}
```

File 144 `lwarp-prettyref.sty`

§ 229 Package

prettyref

(*Emulates or patches code by* Kevin S. Ruland.)

Pkg `prettyref` prettyref is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{prettyref}
```

```
\newrefformat{fig}{Figure \ref{#1}}
\newrefformat{tab}{Table \ref{#1}}
```

File 145 `lwarp-preview.sty`

§ 230 Package

preview

Pkg `preview` preview is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{preview}
```

```
\newenvironment{preview}{}{}
\newenvironment{nopreview}{}{}
\NewDocumentCommand{\PreviewMacro}{s o o +m}{}
\NewDocumentCommand{\PreviewEnvironment}{s o o +m}{}
\newcommand{\PreviewSnarfEnvironment}[2][]{}
\NewDocumentCommand{\PreviewOpen}{s o}{}
\NewDocumentCommand{\PreviewClose}{s o}{}
\let\ifPreview\iffalse% \fi for syntax highlighting
```

File 146 `lwarp-quotchap.sty`

§ 231 Package

quotchap

(*Emulates or patches code by* Karsten Tinnefeld, Jan Klever.)

Pkg `quotchap` quotchap is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{quotchap}

\newcommand{\@quotchap}{}
\newlength{\LWR@quotchapwidth}

\let\@printcites\relax

\newcommand*{\@iprintcites}{%
```

Place the quotes inside a of class quotchap, of the maximum selected width:

```
\uselengthunit{PT}%
\begin{BlockClass}[max-width: \rndprintlength{\LWR@quotchapwidth}]{quotchap}
%\begin{minipage}{\LWR@quotchapwidth}
\@quotchap
%\end{minipage}
\end{BlockClass}
```

Deactivate the quote printing:

```
\global\let\@printcites\relax
}

\NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:
https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment

```
\setlength{\LWR@quotchapwidth}{#1*2}%
\global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
\global\let\@quotchap\BODY
\global\let\@printcites\@iprintcites%
}
```

The quotation author is placed inside a of class qauthor:

```
\newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
\newcommand{\qsetcnfont}[1]{}
```

File 147 `lwarp-ragged2e.sty`

§ 232 Package ragged2e

(*Emulates or patches code by* Martin Schröder.)

Pkg `ragged2e` ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

for HTML output:

```
\LWR@ProvidesPackageDrop{ragged2e}

\newcommand*{\Centering}{\centering}
\newcommand*{\RaggedLeft}{\raggedleft}
\newcommand*{\RaggedRight}{\raggedright}
\newcommand*{\justifying}{}
\newlength{\CenteringLeftskip}
\newlength{\RaggedLeftLeftskip}
\newlength{\RaggedRightLeftskip}
\newlength{\CenteringRightskip}
\newlength{\RaggedLeftRightskip}
\newlength{\RaggedRightRightskip}
\newlength{\CenteringParfillskip}
\newlength{\RaggedLeftParfillskip}
\newlength{\RaggedRightParfillskip}
\newlength{\JustifyingParfillskip}
\newlength{\CenteringParindent}
\newlength{\RaggedLeftParindent}
\newlength{\RaggedRightParindent}
\newlength{\JustifyingParindent}
\newenvironment*{Center}{\center}{\endcenter}
\newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
\newenvironment*{FlushRight}{\flushright}{\endflushright}
\newenvironment*{justify}{\justifying}{\endjustifying}
```

File 148 `lwarp-realscripts.sty`

§ 233 Package realscripts

(*Emulates or patches code by* Will Robertson.)

Pkg `realscripts` realscripts is emulated. See `lwarp.css` for the `<span>` of class `supsubscript`.

for HTML output:

```
\LWR@ProvidesPackageDrop{realscripts}

\let\realsuperscript\textsuperscript
\let\realsubscript\textsubscript

\let\fakesuperscript\textsuperscript
\let\fakesubscript\textsubscript

\newlength{\subsupersep}

\newcommand*{\LWR@realscriptsalign}{}

\newcommand*{\LWR@setrealscriptsalign}[1]{%
\renewcommand*{\LWR@realscriptsalign}{}%
\ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
\ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
}

\DeclareDocumentCommand \textsubsuperscript {s O{l} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsuperscript{#4}\textsubscript{#3}%
}%
}

\DeclareDocumentCommand \textsupersubscript {s O{l} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsubscript{#4}\textsuperscript{#3}%
}%
}
```

File 149 `lwarp-relsize.sty`

§ 234 Package relsize

(*Emulates or patches code by* Donald Arseneau, Bernie Cosell, Matt Swift.)

Pkg `relsize` relsize is patched for use by lwarp.

For HTML only the inline macros are supported: `\textlarger`, `\textsmaller`, and `\textscale`. Each becomes an inline span of a modified `font-size`.

`\relsize`, `\larger`, `\smaller`, and `\relscale` are ignored.

While creating SVG math for HTML, the original definitions are temporarilty restored, and so should work as expected.

⚠ not small The HTML browser's setting for minumum font size may limit how small the output will be displayed.

for HTML output:

```
\LWR@ProvidesPackagePass{relsize}

\let\LWR@origrelsize\relsize
\LetLtxMacro\LWR@origlarger\larger
\LetLtxMacro\LWR@origsmaller\smaller
\let\LWR@relscale\relscale
\LetLtxMacro\LWR@origtextlarger\textlarger
\LetLtxMacro\LWR@origtextsmaller\textsmaller
\let\LWR@textscale\textscale

\appto{\LWR@restoreorigformatting}{%
\let\relsize\LWR@origrelsize%
\LetLtxMacro\larger\LWR@origlarger%
\LetLtxMacro\smaller\LWR@origsmaller%
\let\relscale\LWR@relscale%
\LetLtxMacro\textlarger\LWR@origtextlarger%
\LetLtxMacro\textsmaller\LWR@origtextsmaller%
\let\textscale\LWR@textscale%
}

\newcounter{LWR@relsizetemp}

\renewcommand*{\relsize}[1]{}
\renewcommand*{\larger}[1][]{}
\renewcommand*{\smaller}[1][]{}
\renewcommand*{\relscale}[1]{}

\renewcommand*{\textlarger}[2][1]{%
\setcounter{LWR@relsizetemp}{100+(#1*20)}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textlarger}{#2}%
}

\renewcommand*{\textsmaller}[2][1]{%
\setcounter{LWR@relsizetemp}{100-(#1*20)}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textsmaller}{#2}%
}

\renewcommand*{\textscale}[2]{%
\setcounter{LWR@relsizetemp}{100*\real{#1}}%
\InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textscale}{#2}%
}
```

File 150 `lwarp-romanbar.sty`

§ 235 Package romanbar

(*Emulates or patches code by* H.-Martin Münch.)

Pkg `romanbar` romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

for HTML output:

```
\LWR@ProvidesPackagePass{romanbar}

\DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
\InlineClass[%
    text-decoration: overline underline ;
]{romanbar}{#1}%
}
```

File 151 `lwarp-romanbarpagenumber.sty`

§ 236 Package romanbarpagenumber

Pkg `romanbarpagenumber` romanbarpagenumber is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{romanbarpagenumber}
```

File 152 `lwarp-rotating.sty`

§ 237 Package rotating

(*Emulates or patches code by* Robin Fairbairns, Sebastian Rahtz, Leonor Barroca.)

Pkg `rotating` rotating is emulated.

All rotations are ignored in HTML output.

for HTML output:

```
\LWR@ProvidesPackageDrop{rotating}

\LetLtxMacro\sidewaystable\table
\let\endsidewaystable\endtable
```

```
4
5 \LetLtxMacro\sidewaysfigure\figure
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption
```

File 153 `lwarp-rotfloat.sty`

§ 238 Package

rotfloat

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg `rotfloat` rotfloat is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}
```

`\newfloat` {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the `\newfloat` command from the float package. Sideways floats are `\let` to the same as regular floats.

"placement" is ignored.

```
4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}%
6 {%
7     \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {%
10     \DeclareFloatingEnvironment[fileext=#3]{#1}%

11     \DeclareFloatingEnvironment[fileext=#3]{sideways#1}%
12 }%
13 \csletcs{sideways#1}{#1}%
14 \csletcs{endsideways#1}{end#1}%
```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
\cslet{listof#1s}\relax%
\cslet{listof#1es}\relax%
}
```

File 154 `lwarp-savetrees.sty`

§ 239 Package savetrees

Pkg `savetrees` Emulated.

for HTML output: Discard all options for lwarp-savetrees:

```
\LWR@ProvidesPackageDrop{savetrees}
```

File 155 `lwarp-scalefnt.sty`

§ 240 Package scalefnt

(*Emulates or patches code by* D. Carlisle.)

Pkg `scalefnt` scalefnt is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{scalefnt}
\DeclareRobustCommand\scalefont[1]{}
```

File 156 `lwarp-scrextend.sty`

§ 241 Package scrextend

Pkg `scrextend` scrextend is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```
\LWR@ProvidesPackageDrop{scrextend}
\DeclareDocumentCommand{\setkomafont}{m m}{}
\DeclareDocumentCommand{\addkomafont}{m m}{}
\DeclareDocumentCommand{\usekomafont}{m}{}
```

```
\DeclareDocumentCommand{\usefontofkomafont}{m}{}
\DeclareDocumentCommand{\useencodingofkomafont}{m}{}
\DeclareDocumentCommand{\usesizeofkomafont}{m}{}
\DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
\DeclareDocumentCommand{\useseriesofkomafont}{m}{}
\DeclareDocumentCommand{\useshapeofkomafont}{m}{}

\AtBeginDocument{
\let\LWR@maketitle\maketitle
\DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
}

\DeclareDocumentCommand{\extratitle}{m}{}
\DeclareDocumentCommand{\titlehead}{m}{}
\DeclareDocumentCommand{\subject}{m}{}
\DeclareDocumentCommand{\publishers}{m}{\published{#1}}
\DeclareDocumentCommand{\uppertitleback}{m}{}
\DeclareDocumentCommand{\lowertitleback}{m}{}
\DeclareDocumentCommand{\dedication}{m}{}

\DeclareDocumentCommand{\ifthispageodd}{m m}{#1}

\DeclareDocumentCommand{\titlepagestyle}{}{}

\DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleplainpage}{}{}
\DeclareDocumentCommand{\cleardoublestandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleoddemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleoddplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddstandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}

\DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
  \begingroup\let\thefootnotemark\multfootsep\@makefnmark\endgroup
}

\DeclareDocumentCommand{\multfootsep}{}{,}

\DeclareDocumentCommand{\footref}{m}{%
  \begingroup
    \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
  \endgroup
```

```
  \@footnotemark
}

\DeclareDocumentCommand{\deffootnote}{o m m m}{}
\DeclareDocumentCommand{\deffootnotemark}{m}{}
\DeclareDocumentCommand{\setfootnoterule}{o m}{}
\DeclareDocumentCommand{\raggedfootnote}{}{}

\DeclareDocumentCommand{\dictum}{o m}{
\begin{LWR@BlockClassWP}{text-align:right}{}{dictum}
    #2
    \IfValueT{#1}
    {
        \ifbool{FormatWP}
        {\begin{BlockClass}[border-top:1px solid gray]{dictumauthor}}
        {\begin{BlockClass}{dictumauthor}}
        \dictumauthorformat{#1}
        \end{BlockClass}
    }
\end{LWR@BlockClassWP}
}

\DeclareDocumentCommand{\dictumwidth}{}{}
\DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
\DeclareDocumentCommand{\dictumrule}{}{}
\DeclareDocumentCommand{\raggeddictum}{}{}
\DeclareDocumentCommand{\raggeddictumtext}{}{}
\DeclareDocumentCommand{\raggeddictumauthor}{}{}

\DeclareDocumentEnvironment{labeling}{o m}
{%
\def\sc@septext{#1}%
\list{}{}%
\let\makelabel\labelinglabel%
}
{
\endlist
}

\DeclareDocumentCommand{\labelinglabel}{m}{%
#1 \qquad \sc@septext%
}

\let\addmargin\relax
\let\endaddmargin\relax
\cslet{addmargin*}{\relax}
\cslet{endaddmargin*}{\relax}

\NewDocumentEnvironment{addmargin}{s O{} m}
{
```

```
\uselengthunit{PT}%
\setlength{\LWR@templengthtwo}{#3}
\ifblank{#2}
{
    \begin{BlockClass}[
        margin-left:\rndprintlength{\LWR@templengthtwo} ;
        margin-right:\rndprintlength{\LWR@templengthtwo}
    ]{addmargin}
}
{
    \setlength{\LWR@templengthone}{#2}
    \begin{BlockClass}[
        margin-left:\rndprintlength{\LWR@templengthone} ;
        margin-right:\rndprintlength{\LWR@templengthtwo}
    ]{addmargin}
}
}
{\end{BlockClass}}
```

Ref to create a starred environment:
https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment

```

\ExplSyntaxOn
\cs_new:cpn {addmargin*} {\addmargin*}
\cs_new_eq:cN {endaddmargin*} \endaddmargin
\ExplSyntaxOff

\DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}
```

File 157 `lwarp-scrhack.sty`

§ 242 Package scrhack

Pkg `scrhack` scrhack is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{scrhack}
```

File 158 `lwarp-scrlayer.sty`

§ 243 Package scrlayer

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer` scrlayer is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer}

\newcommand*{\DeclareSectionNumberDepth}[2]{}
\newcommand*{\DeclareLayer}[2][]{}
\newcommand*{\DeclareNewLayer}[2][]{}
\newcommand*{\ProvideLayer}[2][]{}
\newcommand*{\RedeclareLayer}[2][]{}
\newcommand*{\ModifyLayer}[2][]{}
\newcommand*{\layerhalign}{}
\newcommand*{\layervalign}{}
\newcommand*{\layerxoffset}{}
\newcommand*{\layeryoffset}{}
\newcommand*{\layerwidth}{}
\newcommand*{\layerheight}{}
\providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
\newcommand*{\putUL}[1]{}
\newcommand*{\putUR}[1]{}
\newcommand*{\putLL}[1]{}
\newcommand*{\putLR}[1]{}
\newcommand*{\putC}[1]{}
\newcommand*{\GetLayerContents}[1]{}
\newcommand{\IfLayerExists}[3]{#3}
\newcommand*{\DestroyLayer}[1]{}
\newcommand*{\layercontentsmeasure}{}
\newcommand*{\currentpagestyle}{}
\newcommand*{\BeforeSelectAnyPageStyle}[1]{}
\newcommand*{\AfterSelectAnyPageStyle}[1]{}
\newcommand*{\DeclarePageStyleAlias}[2]{}
\newcommand*{\DeclareNewPageStyleAlias}[2]{}
\newcommand*{\ProvidePageStyleAlias}[2]{}
\newcommand*{\RedeclarePageStyleAlias}[2]{}
\newcommand*{\DestroyPageStyleAlias}[1]{}
\newcommand*{\GetRealPageStyle}[1]{}
\newcommand*{\DeclarePageStyleByLayers}[3][]{}
\newcommand*{\DeclareNewPageStyleByLayers}[3][]{}
\newcommand*{\ProvidePageStyleByLayers}[3][]{}
\newcommand*{\RedeclarePageStyleByLayers}[3][]{}
\NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}{}
\newcommand*{\AddLayersToPageStyle}[2]{}
\newcommand*{\AddLayersAtBeginOfPageStyle}[2]{}
\newcommand*{\AddLayersAtEndOfPageStyle}[2]{}
\newcommand*{\RemoveLayersFromPageStyle}[2]{}
\newcommand*{\AddLayersToPageStyleBeforeLayer}[3]{}
\newcommand*{\AddLayersToPageStyleAfterLayer}[3]{}
\newcommand*{\UnifyLayersAtPageStyle}[1]{}
\newcommand*{\ModifyLayerPageStyleOptions}[2]{}
```

```
\newcommand*{\AddToLayerPageStyleOptions}[2]{}
\newcommand{\IfLayerPageStyleExists}[3]{#3}
\newcommand{\IfRealLayerPageStyleExists}[3]{#3}
\newcommand{\IfLayerAtPageStyle}[4]{#4}
\newcommand{\IfSomeLayerAtPageStyle}[4]{#4}
\newcommand{\IfLayersAtPageStyle}[4]{#4}
\newcommand*{\DestroyRealLayerPageStyle}[1]{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\@mkleft}[1]{}
\newcommand*{\@mkright}[1]{}
\newcommand*{\@mkdouble}[1]{}
\newcommand*{\@mkboth}[2]{}
\newcommand*{\scrlayerInitInterface}[1][]{}
\newcommand{\scrlayerAddToInterface}[3][]{}
\newcommand{\scrlayerAddCsToInterface}[3][]{}
\newcommand{\scrlayerOnAutoRemoveInterface}[2][]{}
```

File 159 **lwarp-scrlayer-notecolumn.sty**

§ 244 Package scrlayer-notecolumn

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-notecolumn` scrlayer-notecolumn is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer-notecolumn}
```

```
\newcommand*{\DeclareNoteColumn}[2][]{}
\newcommand*{\DeclareNewNoteColumn}[2][]{}
\newcommand*{\ProvideNoteColumn}[2][]{}
\newcommand*{\RedeclareNoteColumn}[2][]{}
\NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
\newcommand*{\syncwithnotecolumn}[1][]{}
\newcommand*{\syncwithnotecolumns}[1][]{}
\newcommand*{\clearnotecolumn}[1][]{}
\newcommand*{\clearnotecolumns}[1][]{}
```

File 160 `lwarp-scrlayer-scrpage.sty`

§ 245 Package

scrlayer-scrpage

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-scrpage` scrlayer-scrpage is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer-scrpage}

\@ifundefined{footheight}{\newlength\footheight}{}
\NewDocumentCommand{\lehead}{s o m}{}
\NewDocumentCommand{\cehead}{s o m}{}
\NewDocumentCommand{\rehead}{s o m}{}
\NewDocumentCommand{\lohead}{s o m}{}
\NewDocumentCommand{\cohead}{s o m}{}
\NewDocumentCommand{\rohead}{s o m}{}
\NewDocumentCommand{\lefoot}{s o m}{}
\NewDocumentCommand{\cefoot}{s o m}{}
\NewDocumentCommand{\refoot}{s o m}{}
\NewDocumentCommand{\lofoot}{s o m}{}
\NewDocumentCommand{\cofoot}{s o m}{}
\NewDocumentCommand{\rofoot}{s o m}{}
\NewDocumentCommand{\ohead}{s o m}{}
\NewDocumentCommand{\chead}{s o m}{}
\NewDocumentCommand{\ihead}{s o m}{}
\NewDocumentCommand{\ofoot}{s o m}{}
\NewDocumentCommand{\cfoot}{s o m}{}
\NewDocumentCommand{\ifoot}{s o m}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\defpairofpagestyles}[3][]{}
\newcommand*{\newpairofpagestyles}[3][]{}
\newcommand*{\renewpairofpagestyles}[3][]{}
\newcommand*{\providepairofpagestyles}[3][]{}
\newcommand*{\clearmainofpairofpagestyles}
\newcommand*{\clearplainofpairofpagestyles}
\newcommand*{\clearpairofpagestyles}
\NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
```

```
\NewDocumentCommand{\providetriplepagestyle}{m o o m m m m m m}{}
\newcommand*{\defpagestyle}[3]{}
\newcommand*{\newpagestyle}[3]{}
\newcommand*{\providepagestyle}[3]{}
\newcommand*{\renewpagestyle}[3]{}
```

File 161 `lwarp-section.sty`

§ 246 Package section

Pkg `section` section is ignored.

(*Emulates or patches code by* Oliver Pretzel.)

for HTML output:

```
\LWR@ProvidesPackageDrop{section}

\ifx\chapter\undefined
 \def\chsize{\Large}\def\hdsize{\huge}\else
 \def\chsize{\huge}\def\hdsize{\Huge}
\fi
\let\ttsize\LARGE
\let\ausize\large
\let\dasize\large
\let\secsize\Large
\let\subsize\large
\let\hdpos\raggedright
\newcounter{hddepth}
\let\fpind\relax
\def\ttfnt{}
\def\hdfnt{}
\def\fefnt{}
\def\thfnt{}
\def\pgfnt{}
\def\hmkfnt{}
\let\mkcse\uppercase
\def\hddot{}
\def\cpdot{:}
\def\nmdot{}
\ifx\secindent\undefined
 \newdimen\secindent
 \newskip\secpreskp
 \newskip\secpstskp
 \newdimen\subindent
 \newskip\subpreskp
 \newskip\subpstskp
 \newskip\parpstskp
 \newcount\c@hddepth
```

```
\fi
```

File 162 `lwarp-sectsty.sty`

§ 247 Package sectsty

(*Emulates or patches code by* Rowland McDonnell.)

Pkg `sectsty` sectsty is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{sectsty}

\newcommand*{\partfont}          [1] {}
\newcommand*{\partnumberfont}    [1] {}
\newcommand*{\parttitlefont}     [1] {}
\newcommand*{\chapterfont}       [1] {}
\newcommand*{\chapternumberfont} [1] {}
\newcommand*{\chaptertitlefont}  [1] {}
\newcommand*{\sectionfont}       [1] {}
\newcommand*{\subsectionfont}    [1] {}
\newcommand*{\subsubsectionfont} [1] {}
\newcommand*{\paragraphfont}     [1] {}
\newcommand*{\subparagraphfont}  [1] {}
\newcommand*{\minisecfont}  [1] {}
\newcommand*{\allsectionsfont}[1] {}
\newcommand{\nohang}{}
```

`\sectionrule` is only to be used in `*font` commands, thus it is ignored.

```
\newcommand*{\sectionrule}[5]{}

\def\ulemheading#1#2{}
```

File 163 `lwarp-setspace.sty`

§ 248 Package setspace

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `setspace` setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

for HTML output:

```
\LWR@ProvidesPackageDrop{setspace}
```

```
 2
 3 \newcommand*{\setstretch}[1]{}
 4 \newcommand*{\SetSinglespace}[1]{}
 5 \newcommand*{\singlespacing}{}
 6 \newcommand*{\onehalfspacing}{}
 7 \newcommand*{\doublespacing}{}
 8
 9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

File 164 `lwarp-shadow.sty`

§ 249 Package shadow

(*Emulates or patches code by* Mauro Orlandini.)

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-shadow:

```
\LWR@ProvidesPackageDrop{shadow}

\newdimen\sboxsep
\newdimen\sboxrule
\newdimen\sdim

\newcommand{\shabox}[1]{%
\InlineClass{shabox}{#1}%
}
```

File 165 `lwarp-showidx.sty`

§ 250 Package showidx

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```
\LWR@ProvidesPackageDrop{showidx}
```

File 166 `lwarp-showkeys.sty`

§ 251 Package showkeys

(*Emulates or patches code by* David Carlisle, Morten Høgholm.)

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for lwarp-showkeys:

```
\LWR@ProvidesPackageDrop{showkeys}

\NewDocumentCommand{\showkeys}{s}{}
```

File 167 `lwarp-sidecap.sty`

§ 252 Package sidecap

(*Emulates or patches code by* Rolf Niepraschk, Hubert Gässlein.)

Pkg sidecap sidecap is emulated.

for HTML output: Discard all options for lwarp-sidecap.

```
\LWR@ProvidesPackageDrop{sidecap}
```

See:
http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment
regarding the creation of starred environments with xparse.

```
\NewDocumentEnvironment{SCtable}{soo}
{\IfValueTF{#3}{\table[#3]}{\table}}
{\endtable}

\ExplSyntaxOn
\cs_new:cpn {SCtable*} {\SCtable*}
\cs_new_eq:cN {endSCtable*} \endSCtable
\ExplSyntaxOff


\NewDocumentEnvironment{SCfigure}{soo}
{\IfValueTF{#3}{\figure[#3]}{\figure}}
{\endfigure}

\ExplSyntaxOn
\cs_new:cpn {SCfigure*} {\SCfigure*}
\cs_new_eq:cN {endSCfigure*} \endSCfigure
\ExplSyntaxOff


\newenvironment*{wide}{}{}
```

File 168 `lwarp-sidenotes.sty`

§ 253 Package sidenotes

(*Emulates or patches code by* Andy Thomas, Oliver Schebaum.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
\LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
\RenewDocumentCommand \sidecaption {s o o m}
{
   \LWR@stoppars
  \captionsetup{style=sidecaption}
  \IfBooleanTF{#1}
  { % starred
    \IfNoValueOrEmptyTF{#2}
    {\marginnote{\caption*{#4}}}
    {\marginnote{\caption*{#4}}[#2]}
  }
  { % unstarred
  \IfNoValueOrEmptyTF{#2}
    {\def\@sidenotes@sidecaption@tof{#4}}
    {\def\@sidenotes@sidecaption@tof{#2}}
  \IfNoValueOrEmptyTF{#3}
    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
  }
    \LWR@startpars
}
```

Borrowed from the lwarp version of keyfloat:

```
\NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
{% start
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}

\RenewDocumentEnvironment{marginfigure}{o}
  {\begin{KFLTsidenotes@marginfloat}{figure}}
  {\end{KFLTsidenotes@marginfloat}}

\RenewDocumentEnvironment{margintable}{o}
  {\begin{KFLTsidenotes@marginfloat}{table}}
  {\end{KFLTsidenotes@marginfloat}}
```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the LaTeX 2ε `article.cls` source:

```
\renewenvironment{figure*}
               {\@dblfloat{figure}}
```

```
                {\end@dblfloat}

\renewenvironment{table*}
                {\@dblfloat{table}}
                {\end@dblfloat}
```

File 169 `lwarp-siunitx.sty`

§ 254 Package siunitx

(*Emulates or patches code by* Joseph Wright.)

Pkg `siunitx` siunitx is patched for use by lwarp.

⚠ per-mode Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ math mode required Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the siunitx extension for MathJax is not currently hosted at any public CDN, thus siunitx is not usable with MathJax unless a local copy of this extension is created first.

for HTML output:

```
\LWR@ProvidesPackagePass{siunitx}

\AtBeginDocument{% in case textcomp was not loaded
\DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
\DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
\DeclareSIUnit\elementarycharge{\textit{e}}
\DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
\DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
}% AtBeginDocument
```

The following is executed by siunitx at the end of `\document`. `\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original.

```
\ExplSyntaxOn
\cs_undefine:N \__siunitx_set_math_fam:n
\cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
   \LWR@traceinfo{sunitx set math fam}
 \int_new:c { c__siunitx_math #1 _int }
 \group_begin:
   \LetLtxMacro\@ensuredmath\LWR@origensuredmath
   \hbox_set:Nn \l__siunitx_tmp_box
     {
       \ensuremath
```

```
          {
            \use:c { math #1 }
              {
                \int_gset:cn { c__siunitx_math #1 _int } { \fam }
              }
          }
      }
  \group_end:
    \LWR@traceinfo{sunitx set math fam: done}
}

\cs_undefine:N \__siunitx_combined_output:n
\cs_new_protected:Npn \__siunitx_combined_output:n #1 {
    \group_begin:
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath
  \bool_if:NTF \l__siunitx_number_parse_bool
    {
      \tl_clear:N \l__siunitx_number_out_tl
      \bool_set_false:N \l__siunitx_number_compound_bool
      \__siunitx_number_output_parse:n {#1}
    }
    {
      \__siunitx_unit_output_pre_print:
      \__siunitx_print:nn { number } { \ensuremath {#1} }
      \__siunitx_unit_output_print:
    }
    \group_end:
}

\ExplSyntaxOff
```

File 170 **lwarp-soul.sty**

§ 255 Package

soul

(*Emulates or patches code by* Melchior FRANZ.)

Pkg `soul` Emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{soul}[2003/11/17]
\RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
\newcommand*{\LWR@soululcolor}{}

\newcommand*{\LWR@soulstcolor}{}
```

```
6
7 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*{\LWR@soulhlcolor}{LWR@soulhlcolordefault}
9 \newcommand*{\LWR@soulhlcolor}{}
```

`\so` {⟨*text*⟩}

Basic markup with CSS:

```
10 \newcommand{\so}[1]{%
11 \LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
12 }
```

`\caps` {⟨*text*⟩}

```
13 \newcommand{\caps}[1]{%
14 \LWR@HTMLtextstyle%
15     {font-variant:small-caps;letter-spacing:.1ex}%
16     {capsspacing}{#1}%
17 }
```

`\LWR@soulcolor` {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsempty{#2}%
20 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
21 {%
22     \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
23     \LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}%
24 }%
25 }

26 \newcommand{\ul}[1]{%
27 \LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
28     {text-decoration:underline;text-decoration-skip;}%
29 }
30
31 \newcommand{\st}[1]{
32 \LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
33     {text-decoration:line-through}%
34 }
35
36 \newcommand{\hl}[1]{
37 \LWR@soulcolor{#1}{LWR@soulhlcolor}{highlight}{background-color}%
38     {background:\#F8E800}
39 }
```

Nullified:

```
\newcommand*{\soulaccent}[1]{}
\newcommand*{\soulregister}[2]{}
\newcommand{\sloppyword}[1]{#1}
\newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
\newcommand*{\resetso}{}
\newcommand*{\capsdef}[5]{}
\newcommand*{\capsreset}{}
\newcommand*{\capssave}[1]{}
\newcommand*{\capsselect}[1]{}
\newcommand*{\setul}[2]{}
\newcommand*{\resetul}{}
\newcommand*{\setuldepth}[1]{}
\newcommand*{\setuloverlap}[1]{}
```

Set colors:

```
\newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
\newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
\newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
\let\textso\so
\let\textul\ul
\let\texthl\hl
\let\textcaps\caps
```

File 171 `lwarp-soulpos.sty`

§ 256 Package soulpos

(*Emulates or patches code by* Javier Bezos.)

Pkg `soulpos` soulpos is emulated.

for HTML output:

```
\RequirePackage{soul}
\RequirePackage{soulutf8}
\LWR@ProvidesPackageDrop{soulpos}

\NewDocumentCommand{\ulposdef}{m o m}{}

\newdimen\ulwidth

\newcommand\ifulstarttype[1]{%
```

```
\expandafter\@secondoftwo%
}

\newcommand\ifulendtype[1]{%
\expandafter\@secondoftwo%
}

\newcommand{\ulstarttype}{0}
\newcommand{\ulendtype}{0}
\newcommand\ulpostolerance{0}%
```

File 172 `lwarp-soulutf8.sty`

§ 257 Package soulutf8

Pkg `soulutf8` soulutf8 is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output:

```
\LWR@ProvidesPackageDrop{soulutf8}
```

File 173 `lwarp-stabular.sty`

§ 258 Package stabular

(*Emulates or patches code by* SIGITAS TOLUŠIS.)

Pkg `stabular` stabular is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{stabular}
```

Env `stabular` `[`⟨*vpos*⟩`]` `{`⟨*colspec*⟩`}`

```
\newenvironment{stabular}[2][c]
{
\renewcommand{\noalign}[1]{}
\begin{tabular}[#1]{#2}
}
{\end{tabular}}
```

Env `stabular` `{`⟨*width*⟩`}` `[`⟨*vpos*⟩`]` `{`⟨*colspec*⟩`}`

```
\NewDocumentEnvironment{stabular*}{m o m}
```

```
{
\renewcommand{\noalign}[1]{}
\begin{tabular}[#2]{#3}
}
{\end{tabular}}
```

File 174 `lwarp-subfig.sty`

§ 259 Package

subfig

(*Emulates or patches code by* Steven Douglas Cochran.)

Pkg `subfig` subfig is supported and patched by lwarp.

⚠ `lof/lotdepth` At present, the package options for `lofdepth` and `lotdepth` are not working. These counters must be set separately after the package has been loaded.

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

for HTML output: Accept all options for lwarp-subfig:

```
\LWR@ProvidesPackagePass{subfig}
```

`\sf@@@subfloat` `{`⟨*1 type*⟩`}` `[`⟨*2 lof entry*⟩`]` `[`⟨*3 caption*⟩`]` `{`⟨*4 contents*⟩`}`

The outer minipage allows side-by-side subfloats with `\hfill` between.

```
\long\def\sf@@@subfloat#1[#2][#3]#4{%
\begin{minipage}{\linewidth}% new
\LWR@stoppars% new
    \@ifundefined{FBsc@max}{}%
        {\FB@readaux{\let\FBsuboheight\relax}}%
    \@tempcnta=\@ne
    \if@minipage
      \@tempcnta=\z@
    \else\ifdim \lastskip=\z@ \else
      \@tempcnta=\tw@
    \fi\fi
    \ifmaincaptiontop
      \sf@top=\sf@nearskip
      \sf@bottom=\sf@farskip
    \else
      \sf@top=\sf@farskip
      \sf@bottom=\sf@nearskip
```

```
    \fi
    \leavevmode
    \setbox\@tempboxa \hbox{#4}%
    \@tempdima=\wd\@tempboxa
    \@ifundefined{FBsc@max}{}%
        {\global\advance\Xhsize-\wd\@tempboxa
         \dimen@=\ht\@tempboxa
         \advance\dimen@\dp\@tempboxa
         \ifdim\dimen@>\FBso@max
           \global\FBso@max\dimen@
         \fi}%
```

Do not use boxes, which interfere with `lateximages`:

```
%     \vtop%
    \bgroup
%       \vbox%
      \bgroup
        \ifcase\@tempcnta
          \@minipagefalse
        \or
%           \vskip\sf@top
        \or
          \ifdim \lastskip=\z@ \else
%             \@tempskipb\sf@top\relax\@xaddvskip
          \fi
        \fi
        \sf@ifpositiontop{%
          \ifx \@empty#3\relax \else
            \sf@subcaption{#1}{#2}{#3}%
%             \vskip\sf@capskip
%             \vskip\sf@captopadj
          \fi\egroup
%           \hrule width0pt height0pt depth0pt
            \LWR@startpars% new
%   \box\@tempboxa
            #4
            \LWR@stoppars% new
        }{%
        \LWR@startpars% new
        \@ifundefined{FBsc@max}%
            {
%   \box\@tempboxa
                #4
            }%
            {\ifx\FBsuboheight\relax
%                \box\@tempboxa
                #4
             \else
%                 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
                #4
```

```
            \fi}%
        \LWR@stoppars% new
          \egroup
          \ifx \@empty#3\relax \else
%             \vskip\sf@capskip
%             \hrule width0pt height0pt depth0pt
            \sf@subcaption{#1}{#2}{#3}%
         \fi
        }%
%       \vskip\sf@bottom
    \egroup
    \@ifundefined{FBsc@max}{}%
        {\addtocounter{FRobj}{-1}%
         \ifnum\c@FRobj=0\else
           \subfloatrowsep
         \fi}%
    \ifmaincaptiontop\else
      \global\advance\@nameuse{c@\@captype}\m@ne
    \fi
\end{minipage}% new
\LWR@startpars% new
  \endgroup\ignorespaces%
}%
```

`\sf@subcaption` {⟨*1 type*⟩} {⟨*2 lof entry*⟩} {⟨*3 caption*⟩}

```
\long\def\sf@subcaption#1#2#3{%
\LWR@stoppars% new
  \ifx \relax#2\relax \else
    \bgroup
      \let\label=\@gobble
      \let\protect=\string
      \def\@subcaplabel{%
        \caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
      \sf@updatecaptionlist{#1}{#2}{\the\value{\@captype}}{\the\value{#1}}%
    \egroup
  \fi
  \bgroup
    \ifx \relax#3\relax
      \let\captionlabelsep=\relax
    \fi
%     \setbox0\vbox{%
%        \hb@xt@\the\@tempdima{%
%
% %          \hss
% %         \parbox[t]{\the\@tempdima}{%
%           \caption@make
%               {\@nameuse{sub\@captype name}}%
%               {\@nameuse{thesub\@captype}}%
```

```
%                {#3}
% %  }%
% %             \hss
%   }
%  }%
    \@ifundefined{FBsc@max}%
%         {\box0}%
        {
%  \parbox[t]{\the\@tempdima}{%
\LWR@traceinfo{sfsubcap B1}% new
            \LWR@figcaption% new
            \caption@make
                {\@nameuse{sub\@captype name}}%
                {\@nameuse{thesub\@captype}}%
                {#3}

            \endLWR@figcaption% new
\LWR@traceinfo{sfsubcap B2}% new
%  }%
        }%
        {\dimen@\ht0%
         \advance\dimen@\dp0%
         \ifdim\dimen@>\FBsc@max
           \global\FBsc@max\dimen@
         \fi
         \FB@readaux{\let\FBsubcheight\relax}%
         \ifx\FBsubcheight\relax
           \def\next{
%  \parbox[t]{\the\@tempdima}
           }%
         \else
           \def\next{
%  \parbox[t][\FBsubcheight][t]{\the\@tempdima}
           }%
         \fi
%          \vbox{%
%            \hb@xt@\the\@tempdima{%

%              \hss
%              \next{%
\LWR@traceinfo{sfsubcap C1}% new
                \caption@make
                    {\@nameuse{sub\@captype name}}%
                    {\@nameuse{thesub\@captype}}%
                    {#3}
\LWR@traceinfo{sfsubcap C1}% new
%  }%
%              \hss

%  }
```

```
%           }
        }%
  \egroup
\LWR@startpars% new
}
```

\caption@@@make {⟨*caption label*⟩} {⟨*caption text*⟩}

```
\renewcommand\caption@@@make[2]{%
\LWR@startpars% new
  \sbox\@tempboxa{#1}%
  \ifdim\wd\@tempboxa=\z@
    \let\caption@lsep\relax
  \fi
  \caption@ifempty{#2}{%
    \let\caption@lsep\@empty
    \let\caption@tfmt\@firstofone
  }%
%   \@setpar{\@@par\caption@@par}\caption@@par
    \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}% new
  \caption@applyfont
  \caption@fmt
    {\ifcaption@star\else
       \begingroup
         \captionlabelfont
         #1%
       \endgroup
     \fi}%
    {\ifcaption@star\else
       \begingroup
         \caption@iflf\captionlabelfont
         \relax\caption@lsep
       \endgroup
     \fi}%
    {{\captiontextfont
      \caption@ifstrut
        {\vrule\@height\ht\strutbox\@width\z@}%
        {}%
      \nobreak\hskip\z@skip % enable hyphenation
      \caption@tfmt{#2}
    \LWR@ensuredoingapar% new
      \caption@ifstrut
        {\ifhmode\@finalstrut\strutbox\fi}%
        {}%
      \par}}
\LWR@stoppars% new
}
```

\caption@@make@ {⟨⟩} {⟨⟩}

```
\renewcommand{\caption@@make@}[2]{%
  \caption@stepthecounter
  \caption@beginhook
      \caption@@@make{#1}{#2}%
  \caption@endhook
}
```

`\subfloat@label` Patches for `\sf@sub@label`:

```
\def\subfloat@label{%
\LWR@ensuredoingapar% new
  \@ifnextchar(%   %) match left parenthesis
    {\sf@sub@label}
    {\sf@sub@label(Sub\@captype\space
                   \@ifundefined{thechapter}{}{\@nameuse{thechapter}\space}%
                   \@nameuse{p@sub\@captype}%
                   \@nameuse{thesub\@captype}.)}}
```

Patches for `\subref`.

`\sf@subref` {⟨*label*⟩}

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
\renewcommand{\sf@subref}[1]{%
\LWR@subnewref{#1}{sub@#1}%
}
```

`\sf@@subref` {⟨*label*⟩}

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
\renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

`\@newsubfloat` [⟨*keys/values*⟩] {⟨*float name*⟩}

```
\LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat

\def\@newsubfloat[#1]#2{%
\LWR@orig@newsubfloat[#1]{#2}%
\renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
}
```

Pre-defined for figures and tables:

`\l@subfigure` {⟨*text*⟩} {⟨*pagenum*⟩}

```
230 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
```

`\l@subtable` {⟨*text*⟩} {⟨*pagenum*⟩}

```
231 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

File 175 `lwarp-subfigure.sty`

§ 260 Package subfigure

Pkg `subfigure` subfigure is emulated by subfig.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{subfigure}
2 \RequirePackage{subfig}

3 \LetLtxMacro\subfigure\subfloat
4 \LetLtxMacro\subtable\subfloat
5 \LetLtxMacro\Subref\subref
6 \@ifundefined{figuretopcaptrue}{\newif\iffiguretopcap}{}
7 \newif\ifsubfiguretopcap
8 \newif\ifsubcaphang
9 \newif\ifsubcapcenter
10 \newif\ifsubcapcenterlast
11 \newif\ifsubcapnooneline
12 \newif\ifsubcapraggedright
13 \newskip\subfigtopskip
14 \newskip\subfigcapskip
15 \newdimen\subfigcaptopadj
16 \newskip\subfigbottomskip
17 \newdimen\subfigcapmargin
18 \newskip\subfiglabelskip
19 \newcommand*{\subcapsize}{}
20 \newcommand*{\subcaplabelfont}{}
21 \newcommand*{\subcapfont}{}
```

File 176 `lwarp-supertabular.sty`

§ 261 Package supertabular

(*Emulates or patches code by* Johannes Braams, Theo Jurriens.)

Pkg `supertabular` supertabular is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{supertabular}
```

⚠ misplaced alignment
alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.8.

```
\newcommand{\LWRST@firsthead}{}

\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRST@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}
\newcommand{\tabletail}[1]{}

\newcommand{\LWRST@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRST@lasttail{#1}%
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRST@caption{\caption[#1]{#2}}%
}

\let\topcaption\tablecaption
\let\bottomcaption\tablecaption


\newcommand*{\LWRST@caption}{}

\newcommand*{\shrinkheight}[1]{}

\NewDocumentEnvironment{supertabular}{s o m}
{%
\LWR@traceinfo{supertabular}%
\table%
\LWRST@caption%
\begin{tabular}{#3}%
\TabularMacro\ifdefvoid{\LWRST@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
}%
{%
\ifdefvoid{\LWRST@lasttail}%
{}%
```

```
{%
\TabularMacro\ResumeTabular%
\LWRST@lasttail%
}%
\end{tabular}%
\endtable%
\LWR@traceinfo{supertabular done}%
}

\NewDocumentEnvironment{mpsupertabular}{s o m}
{\minipage{\linewidth}\supertabular{#3}}
{\endsupertabular\endminipage}
```

File 177 `lwarp-syntonly.sty`

§ 262 Package syntonly

(*Emulates or patches code by* Frank Mittelbach, Rainer Schöpf.)

Pkg `syntonly` Emulated.

for HTML output: Discard all options for lwarp-syntonly:

```
\LWR@ProvidesPackageDrop{syntonly}

\newif\ifsyntax@
\syntax@false

\newcommand*{\syntaxonly}{}

\@onlypreamble\syntaxonly
```

File 178 `lwarp-tabls.sty`

§ 263 Package tabls

(*Emulates or patches code by* Donald Arseneau.)

Pkg `tabls` tabls is emulated. `\LWR@hline` is used to handle the optional argument when tabls is loaded.

for HTML output:

```
\LWR@ProvidesPackageDrop{tabls}

\newdimen\tablinesep
```

```
\newdimen\arraylinesep
\newdimen\extrarulesep
```

File 179 `lwarp-tabularx.sty`

§ 264 Package tabularx

(*Emulates or patches code by* David Carlisle.)

Pkg `tabularx` tabularx is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabularx:

```
\LWR@ProvidesPackageDrop{tabularx}

\DeclareDocumentEnvironment{tabularx}{m o m}
{\tabular{#3}}
{\endtabular}

\DeclareDocumentEnvironment{tabularx*}{m o m}
{\tabular{#3}}
{\endtabular}
```

File 180 `lwarp-tabulary.sty`

§ 265 Package tabulary

(*Emulates or patches code by* David Carlisle.)

Pkg `tabulary` tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types `L`, `C`, `R`, and `J` are emulated by lwarp core code.

```
\LWR@ProvidesPackageDrop{tabulary}

\NewDocumentEnvironment{tabulary}{m o m}
{\tabular{#3}}
{\endtabular}

\NewDocumentEnvironment{tabulary*}{m o m}
{\tabular{#3}}
{\endtabular}
```

```
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

File 181 `lwarp-textarea.sty`

§ 266 Package textarea

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg `textarea` textarea is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{textarea}

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restoretextarea\@empty
```

File 182 `lwarp-textcomp.sty`

§ 267 Package textcomp

(*Emulates or patches code by* Frank Mittelbach, Robin Fairbairns, Werner Lemberg.)

Pkg `textcomp` textcomp is patched for use by lwarp.

§ 267.1 Limitations

Some textcomp symbols do not have Unicode equivalents, and thus are not supported.

⚠ Missing symbols Many textcomp symbols are not supported by many fonts. Try using more complete fonts in the css, but expect to see gaps in coverage.

§ 267.2 Package loading

for HTML output:

```
\LWR@ProvidesPackagePass{textcomp}
```

§ 267.3 Remembering original defintions

The following are restored for print when inside a `lateximage`:

```
\let\LWR@origtextdegree\textdegree
\let\LWR@origtextcelsius\textcelsius
\let\LWR@origtextohm\textohm
\let\LWR@origtextmu\textmu
\let\LWR@origtextlquill\textlquill
\let\LWR@origtextrquill\textrquill
\let\LWR@origtextcircledP\textcircledP
\let\LWR@origtexttwelveudash\texttwelveudash
\let\LWR@origtextthreequartersemdash\textthreequartersemdash
\let\LWR@origtextmho\textmho
\let\LWR@origtextnaira\textnaira
\let\LWR@origtextpeso\textpeso
\let\LWR@origtextrecipe\textrecipe
\let\LWR@origtextinterrobangdown\textinterrobangdown
\let\LWR@origtextpertenthousand\textpertenthousand
\let\LWR@origtextbaht\textbaht
\let\LWR@origtextdiscount\textdiscount
\let\LWR@origtextservicemark\textservicemark
\LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
\LetLtxMacro\LWR@origcapitalogonek\capitalogonek
\LetLtxMacro\LWR@origcapitalgrave\capitalgrave
\LetLtxMacro\LWR@origcapitalacute\capitalacute
\LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
\LetLtxMacro\LWR@origcapitaltilde\capitaltilde
\LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
\LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
\LetLtxMacro\LWR@origcapitalring\capitalring
\LetLtxMacro\LWR@origcapitalcaron\capitalcaron
\LetLtxMacro\LWR@origcapitalbreve\capitalbreve
\LetLtxMacro\LWR@origcapitalmacron\capitalmacron
\LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
\LetLtxMacro\LWR@origtextcircled\textcircled
```

§ 267.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

\AtBeginDocument improves support for LuaLaTeX and XeLaTeX.

§ 267.4.1 pdfLaTeX symbols

```
\AtBeginDocument{
\ifPDFTeX
\renewcommand*{\textdegree}{\HTMLentity{deg}}
\renewcommand*{\textcelsius}{\HTMLunicode{2103}}
\renewcommand*{\textohm}{\HTMLunicode{2126}}
\renewcommand*{\textmu}{\HTMLunicode{00B5}}
\renewcommand*{\textlquill}{\HTMLunicode{2045}}
\renewcommand*{\textrquill}{\HTMLunicode{2046}}
\renewcommand*{\textcircledP}{\HTMLunicode{2117}}
\renewcommand*{\texttwelveudash}{\HTMLunicode{2014}}% emdash
\renewcommand*{\textthreequartersemdash}{\HTMLunicode{2014}}% emdash
\renewcommand*{\textmho}{\HTMLunicode{2127}}
\renewcommand*{\textnaira}{\HTMLunicode{20A6}}
\renewcommand*{\textpeso}{\HTMLunicode{20B1}}
\renewcommand*{\textrecipe}{\HTMLunicode{211E}}
\renewcommand*{\textinterrobangdown}{\HTMLunicode{2E18}}
\renewcommand*{\textpertenthousand}{\HTMLunicode{2031}}
\renewcommand*{\textbaht}{\HTMLunicode{0E3F}}
\renewcommand*{\textdiscount}{\%}
\renewcommand*{\textservicemark}{\HTMLunicode{2120}}
\else
```

§ 267.4.2 XeLaTeX and LuaLaTeX symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```
\renewcommand*{\textdegree}{°}
\renewcommand*{\textcelsius}{℃}
\renewcommand*{\textohm}{Ω}
\renewcommand*{\textmu}{µ}
\renewcommand*{\textlquill}{⁅}
\renewcommand*{\textrquill}{⁆}
\renewcommand*{\textcircledP}{℗}
\renewcommand*{\texttwelveudash}{--}% emdash
\renewcommand*{\textthreequartersemdash}{--}% emdash
\renewcommand*{\textmho}{℧}
\renewcommand*{\textnaira}{₦}
\renewcommand*{\textpeso}{₱}
```

```
\renewcommand*{\textrecipe}{℞}
\renewcommand*{\textinterrobangdown}{⸘}
\renewcommand*{\textpertenthousand}{‱}
\renewcommand*{\textbaht}{฿}
\renewcommand*{\textdiscount}{\%}
\renewcommand*{\textservicemark}{}
\fi
```

§ 267.5 HTML dicritics

For HTML, Unicode diacritical marks are used:

```
\renewcommand*{\capitalcedilla}[1]{#1\HTMLunicode{0327}}
\renewcommand*{\capitalogonek}[1]{#1\HTMLunicode{0328}}
\renewcommand*{\capitalgrave}[1]{#1\HTMLunicode{0300}}
\renewcommand*{\capitalacute}[1]{#1\HTMLunicode{0301}}
\renewcommand*{\capitalcircumflex}[1]{#1\HTMLunicode{0302}}
\renewcommand*{\capitaltilde}[1]{#1\HTMLunicode{0303}}
\renewcommand*{\capitaldieresis}[1]{#1\HTMLunicode{0308}}
\renewcommand*{\capitalhungarumlaut}[1]{#1\HTMLunicode{30B}}
\renewcommand*{\capitalring}[1]{#1\HTMLunicode{30A}}
\renewcommand*{\capitalcaron}[1]{#1\HTMLunicode{30C}}
\renewcommand*{\capitalbreve}[1]{#1\HTMLunicode{306}}
\renewcommand*{\capitalmacron}[1]{#1\HTMLunicode{304}}
\renewcommand*{\capitaldotaccent}[1]{#1\HTMLunicode{307}}
```

`\textcircled` becomes a span with a rounded border:

```
\renewcommand*{\textcircled}[1]{%
\InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
}
}% AtBeginDocument
```

§ 267.6 Inside a `lateximage`

When a `lateximage` is begun:

```
\appto{\LWR@restoreorigformatting}{%
\let\textdegree\LWR@origtextdegree%
\let\textcelsius\LWR@origtextcelsius%
\let\textohm\LWR@origtextohm%
\let\textmu\LWR@origtextmu%
\let\textlquill\LWR@origtextlquill%
\let\textrquill\LWR@origtextrquill%
\let\textcircledP\LWR@origtextcircledP%
\let\texttwelveudash\LWR@origtexttwelveudash%
\let\textthreequartersemdash\LWR@origtextthreequartersemdash%
\let\textmho\LWR@origtextmho%
\let\textnaira\LWR@origtextnaira%
```

```
\let\textpeso\LWR@origtextpeso%
\let\textrecipe\LWR@origtextrecipe%
\let\textinterrobangdown\LWR@origtextinterrobangdown%
\let\textpertenthousand\LWR@origtextpertenthousand%
\let\textbaht\LWR@origtextbaht%
\let\textdiscount\LWR@origtextdiscount%
\let\textservicemark\LWR@origtextservicemark%
\LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
\LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
\LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
\LetLtxMacro\capitalacute\LWR@origcapitalacute%
\LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
\LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
\LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
\LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
\LetLtxMacro\capitalring\LWR@origcapitalring%
\LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
\LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
\LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
\LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
\LetLtxMacro\textcircled\LWR@origtextcircled%
}
```

File 183 `lwarp-textfit.sty`

§ 268 Package textfit

Pkg `textfit` textfit is emulated.

Text is placed into a of class `textfit`. Sizes are approximated, and also limited by browser min/max font-size settings.

for HTML output:

```
\LWR@ProvidesPackageDrop{textfit}

\newsavebox{\LWR@textfitbox}

\newcommand*{\LWR@textfitscale}[2]{%
\uselengthunit{PT}%
\setlength{\LWR@templengthone}{#1}%
\setlength{\LWR@templengthone}{%
    1em*\ratio{\LWR@templengthone}{\LWR@templengthtwo}%
}%
\InlineClass[font-size:\rndprintlength{\LWR@templengthone}]{textfit}{#2}%
}

\newcommand*{\scaletowidth}[2]{%
\sbox{\LWR@textfitbox}{#2}%
```

```
\settowidth{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
\LWR@textfitscale{#1}{#2}%
}

\newcommand*{\scaletoheight}[2]{%
\sbox{\LWR@textfitbox}{#2}%
\settoheight{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
\LWR@textfitscale{#1}{#2}%
}
```

File 184 `lwarp-textpos.sty`

§ 269 Package textpos

(*Emulates or patches code by* Norman Gray.)

Pkg `textpos` textpos is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{textpos}

\NewDocumentEnvironment{textblock}{m r()}{}{}
\NewDocumentEnvironment{textblock*}{m o r()}{}{}
\newcommand*{\TPGrid}[3][]{}
\NewDocumentCommand{\TPMargin}{s o}{}
\newcommand*{\textblockcolour}[1]{}
\newcommand*{\textblockrulecolour}[1]{}
\newcommand*{\textblockcolor}[1]{}
\newcommand*{\textblockrulecolor}[1]{}
\newcommand*{\tekstblokkulur}[1]{}
\newcommand*{\tekstblokrulekulur}[1]{}
\newlength{\TPHorizModule}
\newlength{\TPVertModule}
\newlength{\TPboxrulesize}
\newcommand{\textblocklabel}[1]{}
\newcommand*{\showtextsize}{}
\newcommand{\textblockorigin}[2]{}
```

File 185 `lwarp-theorem.sty`

§ 270 Package theorem

(*Emulates or patches code by* Frank Mittelbach.)

Pkg `theorem` theorem is patched for use by lwarp.

Table 14: Theorem package — CSS styling of theorems and proofs

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `<span>` of class `theoremheader`

where `<theoremstyle>` is `plain`, `break`, etc.

for HTML output:

```
\LWR@ProvidesPackagePass{theorem}
```

§ 270.1 Remembering the theorem style

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
   \@ifundefined{th@#1}{\@warning
          {Unknown theoremstyle `#1'. Using `plain'}%
          \theorem@style{plain}%
           \renewcommand{\LWR@newtheoremstyle}{plain}% new
          }%
      {%
          \theorem@style{#1}%
          \renewcommand{\LWR@newtheoremstyle}{#1}% new
      }%
      \begingroup
        \csname th@\the\theorem@style \endcsname
      \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
\gdef\@xnthm#1#2[#3]{%
   \expandafter\@ifdefinable\csname #1\endcsname
   {%
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
    \@definecounter{#1}\@newctr{#1}[#3]%
    \expandafter\xdef\csname the#1\endcsname
      {\expandafter \noexpand \csname the#3\endcsname
       \@thmcountersep \@thmcounter{#1}}%
    \def\@tempa{\global\@namedef{#1}}%
    \expandafter \@tempa \expandafter{%
      \csname th@\the \theorem@style
            \expandafter \endcsname \the \theorem@bodyfont
```

```
    \@thm{#1}{#2}}%
   \global \expandafter \let \csname end#1\endcsname \@endtheorem
   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  }}

\gdef\@ynthm#1#2{%
   \expandafter\@ifdefinable\csname #1\endcsname
  {
   \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
   \@definecounter{#1}%
   \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
    \expandafter{\csname th@\the \theorem@style \expandafter
    \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
   \global \expandafter \let \csname end#1\endcsname \@endtheorem
   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
  }}

\gdef\@othm#1[#2]#3{%
 \expandafter\ifx\csname c@#2\endcsname\relax
  \@nocounterr{#2}%
 \else
  \expandafter\@ifdefinable\csname #1\endcsname
  {
   \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
   \expandafter \xdef \csname the#1\endcsname
    {\expandafter \noexpand \csname the#2\endcsname}%
   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
    \expandafter{\csname th@\the \theorem@style \expandafter
    \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
   \global \expandafter \let \csname end#1\endcsname \@endtheorem
   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
   }%
 \fi}
```

§ 270.2 CSS patches

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```
\gdef\th@plain{%
 \def\@begintheorem##1##2{%
       \item[
```

```
            \InlineClass{theoremheader}{##1\ ##2}
        ]
    }%
\def\@opargbegintheorem##1##2##3{%
   \item[
        \InlineClass{theoremheader}{##1\ ##2\ (##3)}
    ]
    }
}

\gdef\th@break{%
  \def\@begintheorem##1##2{%
    \item[
        \InlineClass{theoremheader}{##1\ ##2}\newline%
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
  \item[
        \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
    ]
    }
}

\gdef\th@marginbreak{%
  \def\@begintheorem##1##2{
    \item[
        \InlineClass{theoremheader}{##2 \qquad ##1}\newline
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
    \item[
        \InlineClass{theoremheader}{##2 \qquad ##1\ %
        (##3)}\newline
    ]
    }
}

\gdef\th@changebreak{%
  \def\@begintheorem##1##2{
    \item[
        \InlineClass{theoremheader}{##2\ ##1}\newline
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
    \item[
        \InlineClass{theoremheader}{ ##2\ ##1\ %
        (##3)}\newline
    ]
    }
}
```

```
\gdef\th@change{%
  \def\@begintheorem##1##2{
    \item[
        \InlineClass{theoremheader}{##2\ ##1}
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
    \item[
        \InlineClass{theoremheader}{##2\ ##1\ (##3)}
    ]
    }
}

\gdef\th@margin{%
  \def\@begintheorem##1##2{
    \item[
        \InlineClass{theoremheader}{##2 \qquad ##1}
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
    \item[
            \InlineClass{theoremheader}{##2 \qquad ##1\ (##3)}
    ]
    }
}
```

Patched for css:

```
\gdef\@thm#1#2{\refstepcounter{#1}%
\LWR@forcenewpage% new
    \BlockClass{theorembody\LWR@thisthmstyle}% new
   \trivlist
   \@topsep \theorempreskipamount               % used by first \item
   \@topsepadd \theorempostskipamount           % used by \@endparenv
   \@ifnextchar [%
   {\@ythm{#1}{#2}}%
   {\@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}}

\gdef\@endtheorem{%
\endtrivlist
\endBlockClass
}
```

File 186 `lwarp-threeparttable.sty`

§ 271 Package

threeparttable

(*Emulates or patches code by* Donald Arseneau.)

Pkg `threeparttable` threeparttable is emulated.

Table note are contained inside a css `<div>` of class `tnotes`. If enumitem is used, the note item labels are also individually highlighted with an additional css `<span>` of class `tnoteitemheader`, otherwise they are plain text.

for HTML output:

```
\LWR@ProvidesPackageDrop{threeparttable}
```

`\LWR@printtablenote` {⟨*text*⟩}

Prints the table note item header inside a css class of `tnoteitemheader`.

```
\newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env `threeparttable` [⟨*alignment*⟩] To emulate threeparttable:

```
\newenvironment*{threeparttable}[1][b]{}{}
```

Env `tablenotes` [⟨*options*⟩]

```
\newenvironment*{tablenotes}[1][]
{%
\LWR@forcenewpage
\BlockClass{tnotes}%
\ltx@ifpackageloaded{enumitem}{%
\setlist[description]{format=\LWR@printtablenote}%
}{}%
\description%
}
{%
\enddescription%
\endBlockClass%
}
```

`\tnote` {⟨*text*⟩}

```
\newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

File 187 `lwarp-tikz.sty`

§ 272 Package

tikz

(*Emulates or patches code by* Till Tantau.)

Pkg `tikz` tikz is supported.

Accept all options for lwarp-tikz:

```
\LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp changes the catcode of $ for its own use. The Ti*k*z `babel` library temporarily changes catcodes back to normal for Ti*k*z's use. tikz v3.0.0 introduced the `babel` library which handles catcode changes. For older versions, lwarp must change $'s catcode itself.

Also see:
https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded

```
\newboolean{LWR@tikzbabel}

\@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
{\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
{\boolfalse{LWR@tikzbabel}}
```

Env `tikzpicture` `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```
\BeforeBeginEnvironment{tikzpicture}{%
\begin{lateximage}%
\ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
{}%
{\catcode`\$=3} % dollar sign is math shift
}

\AfterEndEnvironment{tikzpicture}{%
\end{lateximage}%
\ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
{}%
{\catcode`\$=\active}%
}
```

File 188 `lwarp-titleps.sty`

§ 273 Package

titleps

(*Emulates or patches code by* Javier Bezos.)

Pkg `titleps` titleps is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titleps:

for HTML output:

```
\LWR@ProvidesPackageDrop{titleps}
```

`\pagestyle` and `\thispagestyle` are already disabled in the lwarp code.

`\newpagestyle` {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\newpagestyle}{m o m}{}
```

`\renewpagestyle` {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\renewpagestyle}{m o m}{}
```

`\sethead` [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\sethead}{o o o m m m}{}
```

`\setfoot` [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\setfoot}{o o o m m m}{}
```

`\settitlemarks` * {⟨*names*⟩}

```
\NewDocumentCommand{\settitlemarks}{s m}{}
```

`\headrule`

```
\newcommand*{\headrule}{}
```

`\footrule`

```
\newcommand*{\footrule}{}
```

\setheadrule {⟨*length*⟩}

```
\newcommand*{\setheadrule}[1]{}
```

\setfootrule {⟨*length*⟩}

```
\newcommand*{\setfootrule}[1]{}
```

\makeheadrule

```
\newcommand*{\makeheadrule}{}
```

\makefootrule

```
\newcommand*{\makefootrule}{}
```

\setmarkboth {⟨*code*⟩}

```
\newcommand{\setmarkboth}[1]{}
```

\widenhead

```
\NewDocumentCommand{\widenhead}{s o o m m}{}
```

\bottitlemarks

```
\newcommand*{\bottitlemarks}{}
```

\toptitlemarks

```
\newcommand*{\toptitlemarks}{}
```

\firsttitlemarks

```
\newcommand*{\firsttitlemarks}{}
```

\nexttitlemarks

```
\newcommand*{\nexttoptitlemarks}{}
```

\outertitlemarks

```
\newcommand*{\outertitlemarks}{}
```

\innertitlemarks

```
\newcommand*{\innertitlemarks}{}
```

\newtitlemark * {⟨*name*⟩}

```
\NewDocumentCommand{\newtitlemark}{s m}{}
```

\pretitlemark * {⟨*section*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\pretitlemark}{s m m}{}
```

\ifsamemark {⟨*group*⟩} {⟨*command*⟩} {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\ifsamemark}[4]{}
```

\setfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}
```

\setfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}
```

\nextfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}
```

\nextfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}
```

\newmarkset {⟨*markset*⟩}

```
\newcommand{\newmarkset}[1]{}
```

\newextramark * {⟨*markset*⟩} {⟨*macro-name*⟩}

```
\NewDocumentCommand{\newextramarkset}{s m m}{}
```

\botextramarks {⟨*markset*⟩}

```
\newcommand{\botextramarks}[1]{}
```

\topextramarks {⟨*markset*⟩}

```
\newcommand{\topextramarks}[1]{}
```

\firstextramarks {⟨*markset*⟩}

```
\newcommand{\firstextramarks}[1]{}
```

`\nextextramarks` {⟨*markset*⟩}

```
\newcommand{\nexttopextramarks}[1]{}
```

`\outerextramarks` {⟨*markset*⟩}

```
\newcommand{\outerextramarks}[1]{}
```

`\innerextramarks` {⟨*markset*⟩}

```
\newcommand{\innerextramarks}[1]{}
```

File 189 `lwarp-titleref.sty`

§ 274 Package titleref

Pkg `titleref` titleref is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{titleref}

\LetLtxMacro\titleref\nameref

\providecounter{LWR@currenttitle}

\newcommand*{\currenttitle}{%
    \addtocounter{LWR@currenttitle}{1}%
    \label{currenttitle\arabic{LWR@currenttitle}}%
    \nameref{currenttitle\arabic{LWR@currenttitle}}%
}

\newcommand*{\theTitleReference}[2]{}
```

File 190 `lwarp-titlesec.sty`

§ 275 Package titlesec

(*Emulates or patches code by* Javier Bezos.)

Pkg `titlesec` titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

for HTML output:

```
\LWR@ProvidesPackageDrop{titlesec}
```

\titlelabel {⟨*label-format*⟩}

```
\newcommand*{\titlelabel}[1]{}
```

\titleformat* {⟨*command*⟩} {⟨*format*⟩}

\titleformat {⟨*command*⟩} [⟨*shape*⟩] {⟨*format*⟩} {⟨*label*⟩} {⟨*sep*⟩} {⟨*begfore*⟩} [⟨*after*⟩]

```
\newcommand\titleformat{%
  \@ifstar{\ttl@format@s}%
          {\ttl@format@i}}
\newcommand{\ttl@format@s}[1]{}
\NewDocumentCommand{\ttl@format@i}{m o m m m m o}{}
```

\chaptertitlename

```
\@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
\newcommand\chaptertitlename{\@chapapp}
```

\titlespacing * {⟨*command*⟩} {⟨*left*⟩} {⟨*before*⟩} {⟨*after*⟩} [⟨*right*⟩]

```
\NewDocumentCommand{\titlespacing}{s m m m m o}{}
```

\filright

```
\newcommand*{\filright}{}
```

\filcenter

```
\newcommand*{\filcenter}{}
```

\filleft

```
\newcommand*{\filleft}{}
```

\fillast

```
\newcommand*{\fillast}{}
```

\filinner

```
\newcommand*{\filinner}{}
```

\filouter

```
\newcommand*{\filouter}{}
```

\wordsep

```
\newcommand\wordsep{\fontdimen\tw@\font \@plus
  \fontdimen\thr@@\font \@minus \fontdimen4\font}
```

\titleline * [⟨*align*⟩] {⟨*material*⟩}

```
\NewDocumentCommand{\titleline}{s o m}{}
```

\titlerule [⟨*height*⟩]

```
\providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
\newcommand*{\ttl@rule}[1][]{}
\newcommand*{\ttl@row}[2][]{}
```

\iftitlemeasuring {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\iftitlemeasuring}[2]{#2}
```

\assignpagestyle {⟨*command*⟩} {⟨*pagestyle*⟩}

```
\newcommand{\assignpagestyle}[2]{#2}
```

\titleclass {⟨*name*⟩} [⟨*startlevel*⟩] {⟨*class*⟩} [⟨*cmd*⟩]

```
\NewDocumentCommand{\titleclass}{m o m o}
```

File 191 **lwarp-titletoc.sty**

§ 276 Package

titletoc

(*Emulates or patches code by* Javier Bezos.)

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output:

```
\LWR@ProvidesPackageDrop{titletoc}
```

\dottedcontents {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*label*⟩} {⟨*leader*⟩}

```
\NewDocumentCommand{\dottedcontents}{m o m m m}{}
```

\titlecontents * {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*numbered*⟩} {⟨*numberless*⟩} {⟨*filler*⟩} [⟨*below*

or begin⟩] [⟨*separator*⟩] [⟨*end*⟩]

```
\newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
\NewDocumentCommand{\ttl@tcstar}{m o m m m m o o o}{}
\NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}
```

`\contentsmargin` [⟨*correction*⟩] {⟨*right*⟩}

```
\newcommand{\contentsmargin}[2][]{}
```

`\thecontentslabel`

```
\newcommand*{\thecontentslabel}{thecontentslabel}
```

`\thecontentspage`

```
\newcommand*{\thecontentspage}{thecontentspage}
```

`\contentslabel` [⟨*format*⟩] {⟨*space*⟩}

```
\newcommand{\contentslabel}[2][]{\thecontentslabel}
```

`\contentspage` [⟨*format*⟩]

```
\newcommand{\contentspage}[1][]{\thecontentspage}
```

`\contentspush` {⟨*text*⟩}

```
\newcommand{\contentspush}[1]{}
```

`\contentsuse` {⟨*name*⟩} {⟨*text*⟩}

```
\newcommand{\contentsuse}[2]{}
```

`\startcontents` [⟨*name*⟩]

```
\newcommand*{\startcontents}[1][]{}
```

`\stopcontents` [⟨*name*⟩]

```
\newcommand*{\stopcontents}[1][]{}
```

`\resumecontents` [⟨*name*⟩]

```
\newcommand*{\resumecontents}[1][]{}
```

`\printcontents` [⟨*name*⟩] {⟨*prefix*⟩} {⟨*start*⟩} {⟨*code*⟩}

```
16 \newcommand{\printcontents}[4][]{}
```

\startlist `[⟨name⟩] {⟨list⟩}`

```
17 \newcommand{\startlist}[2][]{}
```

\stoplist `[⟨name⟩] {⟨list⟩}`

```
18 \newcommand{\stoplist}[2][]{}
```

\resumelist `[⟨name⟩] {⟨list⟩}`

```
19 \newcommand{\resumelist}[2][]{}
```

\printlist `[⟨name⟩] {⟨list⟩} {⟨prefix⟩} {⟨code⟩}`

```
20 \newcommand{\printlist}[4][]{}
```

File 192 `lwarp-titling.sty`

§ 277 Package

titling

(*Emulates or patches code by* Peter Wilson.)

Pkg titling

package support
⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

\published and \subtitle

If using the titling package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 56.7.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output:

```
1 \LWR@ProvidesPackagePass{titling}
```

\@bsmtitlempty

Patch \@bsmtitlempty:

```
2 \let\LWR@orig@bsmtitlempty\@bsmtitlempty
3 \renewcommand*{\@bsmtitlempty}{%
4 \LWR@orig@bsmtitlempty%
5 }
```

\keepthetitle

Patch \keepthetitle:

```
\let\LWR@origkeepthetitle\keepthetitle
\renewcommand*{\keepthetitle}{%
\LWR@orig@keepthetitle%
}
```

`\killtitle` Patch `\killtitle`:

```
\let\LWR@origkilltitle\killtitle
\renewcommand*{\killtitle}{%
\LWR@orig@killtitle%
}
```

Env `titlingpage`

```
\renewenvironment*{titlingpage}
{%
```

Start an HTML titlepage div:

```
\LWR@printpendingfootnotes
\begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
\LWR@maketitlesetup
\let\maketitle\LWR@titlingmaketitle
}
{
```

At the end of the environment, end the HTML titlepage div:

```
\end{titlepage}
}
```

Patch the pre/post title/author/date to add HTML tags, then initilize:

```

\pretitle{}
\posttitle{}

\preauthor{}
\postauthor{}

\predate{}
\postdate{}
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
\renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\def\@makefnmark{\textsuperscript{\@thefnmark}}
```

`\thefootnote` ⇒ `\nameuse`{arabic}{footnote}, or
`\thefootnote` ⇒ `\nameuse`{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\makethanksmark~%
```

`\makethanksmark` ⇒ `\thanksfootmark` ⇒ `\tamark` ⇒
`\@thefnmark` ⇒ `\itshape` a (or similar)

Print the text:

```
##1%
}%
}
```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
\begin{titlepage}
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Close the HTML titlepage div:

```
\end{titlepage}
```

Reset the footnote counter:

```
\@bscontmark
}
```

\@maketitle Typesets the title, etc. Patched for HTML.

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \maketitlehooka
    {
        \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
        \@bspretitle \@title \@bsposttitle
        \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    }
    \maketitlehookb
    {
        \begin{BlockClass}{author}
        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
        \@bspreauthor \@author  \@bspostauthor
        \end{BlockClass}
        \end{BlockClass}
    }
    \maketitlehookc
    {
        \begin{BlockClass}{titledate}
        \@bspredate \@date \@bspostdate
        \end{BlockClass}
    }
    \maketitlehookd
}
```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML `titlingpage` environment.

```
\renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
\@thanks
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any `\thanks` footnotes:

```
\@thanks
```

Reset the footnote counter:

```
\@bscontmark
}
```

`\thanksmarkseries` `{`⟨*series*⟩`}`

Sets the type of footnote marks used by `\thanks`, where type is 'arabic', 'roman', 'fnsymbol', etc.

```
\renewcommand{\thanksmarkseries}[1]{%
\def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
}
```

Set default titlepage thanks footnote marks. See section 56.6.

```
\@ifclassloaded{memoir}{
   \thanksmarkseries{arabic}
}{% not memoir
\if@titlepage
   \thanksmarkseries{arabic}
\else
```

```
93    \thanksmarkseries{fnsymbol}
94 \fi
95 }% not memoir
```

File 193 `lwarp-tocbasic.sty`

§ 278 Package tocbasic

(*Emulates or patches code by* Markus Kohm.)

Pkg `tocbasic` tocbasic is patched for use by lwarp.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackagePass{tocbasic}

2 \DeclareDocumentCommand{\usetocbasicnumberline}{o}{}
3 \DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{}
```

File 194 `lwarp-tocbibind.sty`

§ 279 Package tocbibind

(*Emulates or patches code by* Peter Wilson.)

Pkg `tocbibind` tocbibind is patched for use by lwarp.

Opt `IndexLanguage` The lwarp package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.

⚠ tocloft & other packages If using tocloft with tocbibind, anonchap, fncychap, or other packages which change

chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

placement and TOC options

An index may be placed inline with other HTML text, or on its own HTML page:

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

On its own HTML page, with a manual TOC entry:

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

Inline, with an automatic TOC entry:

Pkg `tocbibind`

The tocbibind package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

On its own HTML page, with an automatic TOC entry:

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt `tocbibind` `numindex`

numbered index section

Use the tocbibind `numindex` option to generate a numbered index. Without this option, the index heading has no number.

for HTML output:

```
\let\simplechapterdelim\relax

\LWR@ProvidesPackagePass{tocbibind}
```

```
\renewenvironment{theindex}%
{%
     \if@bibchapter
        \if@donumindex
            \chapter{\indexname}
        \else
          \if@dotocind
            \chapter*{\indexname}
            \addcontentsline{toc}{chapter}{\indexname}
          \else
            \chapter*{\indexname}
          \fi
        \fi
     \else
        \if@donumindex
            \section{\indexname}
        \else
          \if@dotocind
            \section*{\indexname}
            \addcontentsline{toc}{\@tocextra}{\indexname}
          \else
            \section*{\indexname}
          \fi
        \fi
     \fi
\let\item\LWR@indexitem%
\let\subitem\LWR@indexsubitem%
\let\subsubitem\LWR@indexsubsubitem%
}{}
```

The following code is shared by anonchap.

```
\DeclareDocumentCommand{\simplechapter}{O{\@empty}}{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\DeclareDocumentCommand{\restorechapter}{}{%
\let\@chapcntformat\@seccntformat%
}
```

File 195 `lwarp-tocloft.sty`

§ 280 Package tocloft

(*Emulates or patches code by* Peter Wilson.)

Pkg tocloft

tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

⚠ tocloft & other packages

If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard LaTeX commands to create the titles, allowing other packages to work with it.

Discard all options for lwarp-tocloft:

for HTML output:

```
\LWR@ProvidesPackageDrop{tocloft}
```

`\tocloftpagestyle` {⟨*style*⟩}

```
\newcommand{\tocloftpagestyle}[1]{}
```

`\cftmarktoc`

```
\newcommand*{\cftmarktoc}{}
```

`\cfttoctitlefont`

```
\newcommand*{\cfttoctitlefont}{}
```

`\cftaftertoctitle`

```
\newcommand*{\cftaftertoctitle}{}

\newlength{\cftbeforetoctitleskip}
\newlength{\cftaftertoctitleskip}
```

`\cftmarklof`

```
\newcommand*{\cftmarklof}{}
```

`\cftloftitlefont`

```
\newcommand*{\cftloftitlefont}{}
```

`\cftafterloftitle`

```
\newcommand*{\cftafterloftitle}{}

\newlength{\cftbeforeloftitleskip}
\newlength{\cftafterloftitleskip}
```

`\cftmarklot`

```
\newcommand*{\cftmarklot}{}
```

\cftlottitlefont

```
\newcommand*{\cftlottitlefont}{}
```

\cftafterlottitle

```
\newcommand*{\cftafterlottitle}{}

\newlength{\cftbeforelottitleskip}
\newlength{\cftafterlottitleskip}
```

\cftdot

```
\providecommand*{\cftdot}{.}
```

\cftdotsep

```
\providecommand*{\cftdotsep}{1}
```

\cftnodots

```
\providecommand*{\cftnodots}{5000}
```

\cftdotfill {⟨*sep*⟩}

```
\providecommand{\cftdotfill}[1]{}
```

\cftsetpnumwidth {⟨*length*⟩}

```
\DeclareDocumentCommand{\cftsetpnumwidth}{m}{}
```

\cftsetrmarg {⟨*length*⟩}

```
\DeclareDocumentCommand{\cftsetrmarg}{m}{}
```

\cftpnumalign {⟨*alignment*⟩}

```
\DeclareDocumentCommand{\cftpnumalign}{m}{}

\LWR@providelength{\cftparskip}
```

The part-related items are also provided by memoir:

```
\LWR@providelength{\cftbeforepartskip}
```

```
\LWR@providelength{\cftpartindent}
\LWR@providelength{\cftpartnumwidth}
\providecommand*{\cftpartfont}{}
\providecommand*{\cftpartpresnum}{}
\providecommand*{\cftpartaftersnum}{}
\providecommand*{\cftpartaftersnumb}{}
\providecommand*{\cftpartleader}{}
\providecommand*{\cftpartdotsep}{1}
\providecommand*{\cftpartpagefont}{}
\providecommand*{\cftpartafterpnum}{}
```

memoir uses the full name “chapter” instead of “chap”:

```
\LWR@providelength{\cftbeforechapskip}
\LWR@providelength{\cftchapindent}
\LWR@providelength{\cftchapnumwidth}
\newcommand*{\cftchapfont}{}
\newcommand*{\cftchappresnum}{}
\newcommand*{\cftchapaftersnum}{}
\newcommand*{\cftchapaftersnumb}{}
\newcommand*{\cftchapleader}{}
\newcommand*{\cftchapdotsep}{1}
\newcommand*{\cftchappagefont}{}
\newcommand*{\cftchapafterpnum}{}
```

The following do not appear in memoir:

```
\LWR@providelength{\cftbeforesecskip}
\LWR@providelength{\cftsecindent}
\LWR@providelength{\cftsecnumwidth}
\newcommand*{\cftsecfont}{}
\newcommand*{\cftsecpresnum}{}
\newcommand*{\cftsecaftersnum}{}
\newcommand*{\cftsecaftersnumb}{}
\newcommand*{\cftsecleader}{}
\newcommand*{\cftsecdotsep}{1}
\newcommand*{\cftsecpagefont}{}
\newcommand*{\cftsecafterpnum}{}

\LWR@providelength{\cftbeforesubsecskip}
\LWR@providelength{\cftsubsecindent}
\LWR@providelength{\cftsubsecnumwidth}
\newcommand*{\cftsubsecfont}{}
\newcommand*{\cftsubsecpresnum}{}
\newcommand*{\cftsubsecaftersnum}{}
\newcommand*{\cftsubsecaftersnumb}{}
\newcommand*{\cftsubsecleader}{}
\newcommand*{\cftsubsecdotsep}{1}
\newcommand*{\cftsubsecpagefont}{}
```

```
\newcommand*{\cftsubsecafterpnum}{}

\LWR@providelength{\cftbeforesubsubsecskip}
\LWR@providelength{\cftsubsubsecindent}
\LWR@providelength{\cftsubsubsecnumwidth}
\newcommand*{\cftsubsubsecfont}{}
\newcommand*{\cftsubsubsecpresnum}{}
\newcommand*{\cftsubsubsecaftersnum}{}
\newcommand*{\cftsubsubsecaftersnumb}{}
\newcommand*{\cftsubsubsecleader}{}
\newcommand*{\cftsubsubsecdotsep}{1}
\newcommand*{\cftsubsubsecpagefont}{}
\newcommand*{\cftsubsubsecafterpnum}{}

\LWR@providelength{\cftbeforeparaskip}
\LWR@providelength{\cftparaindent}
\LWR@providelength{\cftparanumwidth}
\newcommand*{\cftparafont}{}
\newcommand*{\cftparapresnum}{}
\newcommand*{\cftparaaftersnum}{}
\newcommand*{\cftparaaftersnumb}{}
\newcommand*{\cftparaleader}{}
\newcommand*{\cftparadotsep}{1}
\newcommand*{\cftparapagefont}{}
\newcommand*{\cftparaafterpnum}{}

\LWR@providelength{\cftbeforesubparaskip}
\LWR@providelength{\cftsubparaindent}
\LWR@providelength{\cftsubparanumwidth}
\newcommand*{\cftsubparafont}{}
\newcommand*{\cftsubparapresnum}{}
\newcommand*{\cftsubparaaftersnum}{}
\newcommand*{\cftsubparaaftersnumb}{}
\newcommand*{\cftsubparaleader}{}
\newcommand*{\cftsubparadotsep}{1}
\newcommand*{\cftsubparapagefont}{}
\newcommand*{\cftsubparaafterpnum}{}

\LWR@providelength{\cftbeforefigskip}
\LWR@providelength{\cftfigindent}
\LWR@providelength{\cftfignumwidth}
\newcommand*{\cftfigfont}{}
\newcommand*{\cftfigpresnum}{}
\newcommand*{\cftfigaftersnum}{}
\newcommand*{\cftfigaftersnumb}{}
\newcommand*{\cftfigleader}{}
\newcommand*{\cftfigdotsep}{1}
\newcommand*{\cftfigpagefont}{}
\newcommand*{\cftfigafterpnum}{}
```

```
\LWR@providelength{\cftbeforesubfigskip}
\LWR@providelength{\cftsubfigindent}
\LWR@providelength{\cftsubfignumwidth}
\newcommand*{\cftsubfigfont}{}
\newcommand*{\cftsubfigpresnum}{}
\newcommand*{\cftsubfigaftersnum}{}
\newcommand*{\cftsubfigaftersnumb}{}
\newcommand*{\cftsubfigleader}{}
\newcommand*{\cftsubfigdotsep}{1}
\newcommand*{\cftsubfigpagefont}{}
\newcommand*{\cftsubfigafterpnum}{}

\LWR@providelength{\cftbeforetabskip}
\LWR@providelength{\cfttabindent}
\LWR@providelength{\cfttabnumwidth}
\newcommand*{\cfttabfont}{}
\newcommand*{\cfttabpresnum}{}
\newcommand*{\cfttabaftersnum}{}
\newcommand*{\cfttabaftersnumb}{}
\newcommand*{\cfttableader}{}
\newcommand*{\cfttabdotsep}{1}
\newcommand*{\cfttabpagefont}{}
\newcommand*{\cfttabafterpnum}{}

\LWR@providelength{\cftbeforesubtabskip}
\LWR@providelength{\cftsubtabindent}
\LWR@providelength{\cftsubtabnumwidth}
\newcommand*{\cftsubtabfont}{}
\newcommand*{\cftsubtabpresnum}{}
\newcommand*{\cftsubtabaftersnum}{}
\newcommand*{\cftsubtabaftersnumb}{}
\newcommand*{\cftsubtableader}{}
\newcommand*{\cftsubtabdotsep}{1}
\newcommand*{\cftsubtabpagefont}{}
\newcommand*{\cftsubtabafterpnum}{}

\DeclareDocumentCommand{\cftsetindents}{m m m}{}

\newcommand{\pagenumbersoff}[1]{}
\newcommand{\pagenumberson}[1]{}
```

`\newlistentry` `[⟨within⟩]` `{⟨counter⟩}` `{⟨ext⟩}` `{⟨level-1⟩}`

```
\DeclareDocumentCommand{\newlistentry}{o m m m}
{%
\LWR@traceinfo{newlistentry #2 #3 #4}%
\IfValueTF{#1}%
{%
```

```
    \@ifundefined{c@#2}{%
        \newcounter{#2}[#1]%
        \expandafter\edef\csname the#2\endcsname{%
            \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}%
        }%
    }{}%
}%
{%
    \@ifundefined{c@#2}{%
        \newcounter{#2}%
    }{}%
}%
\@namedef{l@#2}##1##2{%
    \hypertocfloat{1}{#2}{#3}{##1}{##2}%
    \def\cftwhatismyname{#2}% from memoir
}%
\expandafter\newlength\csname cftbefore#2skip\endcsname%
\expandafter\newlength\csname cft#2indent\endcsname%
\expandafter\newlength\csname cft#2numwidth\endcsname%
\@namedef{cft#2font}{}%
\@namedef{cft#2presnum}{}%
\@namedef{cft#2aftersnum}{}%
\@namedef{cft#2aftersnumb}{}%
\@namedef{cft#2leader}{}%
\@namedef{cft#2dotsep}{1}%
\@namedef{cft#2pagefont}{}%
\@namedef{cft#2afterpnum}{}%
\@namedef{toclevel@#2}{#4}%
\@namedef{cft#2fillnum}##1{}%
\LWR@traceinfo{newlistentry done}%
}
```

`\newlistof` [⟨*within*⟩] {⟨*type*⟩} {⟨*ext*⟩} {⟨*listofname*⟩}

Emulated through the `\newfloat` mechanism.

```
\DeclareDocumentCommand{\newlistof}{o m m m}
{%
\IfValueTF{#1}
{\newlistentry[#1]{#2}{#3}{0}}
{\newlistentry{#2}{#3}{0}}
\@namedef{ext@#2}{#3}
\@ifundefined{c@#3depth}{\newcounter{#3depth}}{}
\setcounter{#3depth}{1}
\@namedef{cftmark#3}{}
\@namedef{listof#2}{\listof{#2}{#4}}
\@namedef{@cftmake#3title}{}
\expandafter\newlength\csname cftbefore#3titleskip\endcsname
\expandafter\newlength\csname cftafter#3titleskip\endcsname
\@namedef{cft#3titlefont}{}
```

```
\@namedef{cftafter#3title}{}
\@namedef{cft#3prehook}{}
\@namedef{cft#3posthook}{}
}
```

`\cftchapterprecis` {⟨*text*⟩}

```
\newcommand{\cftchapterprecis}[1]{%
  \cftchapterprecishere{#1}
  \cftchapterprecistoc{#1}}
\newcommand{\cftchapterprecishere}[1]{%
  \begin{quote}\textit{#1}\end{quote}}
\newcommand{\cftchapterprecistoc}[1]{
  \addtocontents{toc}{%
  {
     \protect\begin{quote}#1\protect\end{quote}}
  }
}
```

File 196 `lwarp-tocstyle.sty`

§ 281 Package tocstyle

Pkg `tocstyle` tocstyle is ignored.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{tocstyle}

\newcommand*{\usetocstyle}[2][]{}
\newcommand*{\deactivatetocstyle}[1][]{}
\newcommand*{\reactivatetocstyle}[1][]{}
\NewDocumentCommand{\settocfeature}{o o m m}{}
\NewDocumentCommand{\settocstylefeature}{o m m}{}
\NewDocumentCommand{\newtocstyle}{o o m m}{}
\newcommand*{\aliastoc}[2]{}
\newcommand*{\showtoc}[2][]{}
\newcommand{\iftochasdepth}[4]{}
```

File 197 `lwarp-todo.sty`

§ 282 Package todo

(*Emulates or patches code by* Federico Garcia.)

Pkg `todo` todo is patched for use by lwarp.

for HTML output:

```
\LWR@ProvidesPackagePass{todo}

\renewcommand\todoitem[2]{%
    \refstepcounter{todo}%
    \item[%
        \HTMLunicode{2610} \quad
        \ref{todopage:\thetodo}
     ] : {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
    \label{todolbl:\thetodo}%
}%

\renewcommand\doneitem[2]{%
    \stepcounter{todo}%
    \item[%
        \HTMLunicode{2611} \quad
        \ref{todopage:\thetodo}
    ] \@nameuse{@done\the\c@todo}:
        {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
}

\xpatchcmd{\@displaytodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displaytodo.}}

\xpatchcmd{\@displayfulltodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displayfulltodo.}}

\patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch todoenv.}}

\patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch astodos.}}

\AtBeginDocument{
\crefname{todo}{todo}{todos}
\Crefname{todo}{Todo}{Todos}
}
```

File 198 `lwarp-todonotes.sty`

§ 283 Package todonotes

(*Emulates or patches code by* Henrik Skov Midtiby.)

Pkg todonotes

todonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

for HTML output:

```
\LWR@ProvidesPackagePass{todonotes}

\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}


\renewcommand{\@todonotes@drawMarginNoteWithLine}{
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
        {\@todonotes@author:\,}%
        \fi%
        \@todonotes@text%
    }%
}

\renewcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

```

```
\renewcommand{\@todonotes@drawLineToRightMargin}{}

\renewcommand{\@todonotes@drawLineToLeftMargin}{}

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

\LetLtxMacro\LWRTODONOTES@orig@todo\@todo

\RenewDocumentCommand{\@todo}{o m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\IfValueTF{#1}{%
    \LWRTODONOTES@orig@todo[#1]{#2}%
}{%
    \LWRTODONOTES@orig@todo{#2}%
}
\endgroup%
}

\fi% \if@todonotes@disabled
```

File 199 **lwarp-transparent.sty**

§ 284 Package

transparent

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg transparent

Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

⚠ Not XeLaTeX!

Note that transparent does not work with XeLaTeX.

for HTML output:

Discard all options for lwarp-transparent:

```
\LWR@ProvidesPackageDrop{transparent}

\newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}
```

```
3
4 \newcommand*{\texttransparent}[2]{%
5 \begingroup%
6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }
```

File 200 `lwarp-trivfloat.sty`

§ 285 Package

trivfloat

(*Emulates or patches code by* Joseph Wright.)

Pkg `trivfloat` trivfloat is forced to use the built-in lwarp emulation for floats.

Discard all options for lwarp-trivfloat. This tells trivfloat not to use floatrow or memoir.

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by trivfloat to correct float chapter numbers, but is not needed for lwarp.

for HTML output:

```
3 \begin{warpHTML}
4 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
5 \end{warpHTML}
```

§ 285.1 Combining \newfloat, \trivfloat, and algorithmicx

for HTML & PRINT:

```
6 \begin{warpall}
```

For both print and HTML output:

⚠ When using float, trivfloat, or algorithmicx at the same time, be aware of conflicting file usage. algorithmicx uses `.loa`. trivfloat by default starts with `.loa` and goes up for additional floats, skipping `.lof` and `.lot`.

⚠ When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid `.loa` used by algorithmicx, and any files used by trivfloat. Also avoid using `.lof` and `.lot`.

⚠ When using `\trivfloat`, you may force it to avoid conflicting with algorithmicx by starting trivfloat's file extensions with `.lob`:

```
\makeatletter
\setcounter{tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter
```

```
7 \end{warpall}
```

File 201 `lwarp-turnthepage.sty`

§ 286 Package turnthepage

Pkg `turnthepage` turnthepage is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{turnthepage}
```

```
2 \newcommand{\turnthepage}{}
```

File 202 `lwarp-typearea.sty`

§ 287 Package typearea

(*Emulates or patches code by* Markus Kohm.)

Pkg `typearea` typearea is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. \DeclareDocumentCommand is used to overwrite the koma-script definitions.

for HTML output:

```
\LWR@ProvidesPackageDrop{typearea}

\DeclareDocumentCommand{\typearea}{o m}{}
\DeclareDocumentCommand{\recalctypearea}{}{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\areaset}{o m m}{}
\DeclareDocumentCommand{\activateareas}{}{}
\DeclareDocumentCommand{\storeareas}{m}{}
\DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}
\DeclareDocumentCommand{\AfterSettingArea}{s m}{}
```

File 203 `lwarp-ulem.sty`

§ 288 Package ulem

(*Emulates or patches code by* Donald Arseneau.)

Pkg `ulem` Emulated.

for HTML output: Original lwarp definitions:

```
\LetLtxMacro\LWR@ulemorigemph\emph
\LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using css:

```
\NewDocumentCommand{\uline}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:underline;text-decoration-skip}%
    {uline}{#1}%
}

\NewDocumentCommand{\uuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;text-decoration-skip;%
        text-decoration-style:double%
    }%
    {uuline}{#1}%
}

```

```
\NewDocumentCommand{\uwave}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;text-decoration-skip;%
        text-decoration-style:wavy%
    }%
    {uwave}{#1}%
}

\NewDocumentCommand{\sout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {sout}{#1}%
}

\NewDocumentCommand{\xout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {xout}{#1}%
}

\NewDocumentCommand{\dashuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip;%
        text-decoration-style:dashed%
    }%
    {dashuline}{#1}%
}

\NewDocumentCommand{\dotuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip;%
        text-decoration-style:dotted%
    }%
    {dotuline}{#1}%
}
```

Nullified parameters:

```
\NewDocumentCommand{\ULthickness}{}{}
\newlength{\ULdepth}
```

Nullified/emulated macros:

```
\NewDocumentCommand{\markoverwith}{m}{}
```

```
\NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```
\NewDocumentCommand{\useunder}{m m m}{%
\relax%
\ifx\relax#3\relax\else % argumentative command
    \def#3{#1}\MakeRobust{#3}\fi
}
```

Triggered by package options, also available for the users:

```
\newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
\newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
\ULforem% default
```

Package options:

```
\DeclareOption{normalem}{\normalem}
\DeclareOption{ULforem}{\ULforem}
\DeclareOption{normalbf}{}
\DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}
```

Emulate the original package:

```
\LWR@ProvidesPackageDrop{ulem}
```

File 204 `lwarp-upref.sty`

§ 289 Package upref

Pkg `upref` Ignored.

for HTML output: Discard all options for lwarp-upref:

```
\LWR@ProvidesPackageDrop{upref}
```

File 205 `lwarp-verse.sty`

§ 290 Package verse

(*Emulates or patches code by* Peter Wilson.)

Pkg `verse` verse is supported and patched by lwarp.

for HTML output: Pass all options for lwarp-verse:

```
\LWR@ProvidesPackagePass{verse}
```

`\attrib` The documentation for the verse and memoir packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. lwarp provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len `\leftskip`
Len `\leftmargini`
Len `\TMLvleftskip`
Len `\TMLleftmargini`

These lengths are used by verse and memoir to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word "Chorus", in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain "Chorus". The default is wide enough for a stanza number.

⚠ spacing Horizontal spacing relies on `pdftotext`'s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

Env `verse` The `verse` environment will be placed inside a HTML `<pre>`.

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```
\AtBeginEnvironment{verse}
{%
```

Use the original `list` environment inside a `<pre>` to attempt to preserve formatting.

```
\LWR@restoreoriglists%
```

Pkg `verse`
Cls `memoir`

The verse or memoir packages can place stanza numbers to the left with their

Len \leftskip \flagverse

\flagverse command. Do not allow them to go into the left margin, which would cause pdfcrop to crop the entire page further to the left:

```
\ifdef{\vleftskip}{%
\setlength{\vleftskip}{\HTMLvleftskip}
\setlength{\leftmargini}{\HTMLleftmargini}
}{}
\LWR@forcenewpage
\LWR@atbeginverbatim{verse}
\unskip\LWR@origvspace{-\baselineskip}
}
```

After the end of the verse environment, which places the <pre> tag at the regular left margin:

```
\AtEndEnvironment{verse}{
\LWR@afterendverbatim
}
```

Patch to place poemtitle inside an HTML of class poemtitle:

```
\ifdef{\poemtitle}{
\DeclareDocumentCommand{\@vstypeptitle}{m}{%
   \vspace{\beforepoemtitleskip}%
   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
   \vspace{\afterpoemtitleskip}%
 }
}{}

\LWR@traceinfo{Finished patching verse.}
}% AfterEndPreamble
```

File 206 **lwarp-wallpaper.sty**

§ 291 Package

wallpaper

(*Emulates or patches code by* MICHAEL H.F. WILKINSON.)

Pkg wallpaper

wallpaper is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{wallpaper}

\newcommand*{\CenterWallPaper}[2]{}
\newcommand*{\ThisCenterWallPaper}[2]{}
\newcommand*{\TileWallPaper}[3]{}
\newcommand*{\ThisTileWallPaper}[3]{}
```

```
\newcommand*{\TileSquareWallPaper}[2]{}
\newcommand*{\ThisTileSquareWallPaper}[2]{}
\newcommand*{\ULCornerWallPaper}[2]{}
\newcommand*{\ThisULCornerWallPaper}[2]{}
\newcommand*{\LLCornerWallPaper}[2]{}
\newcommand*{\ThisLLCornerWallPaper}[2]{}
\newcommand*{\URCornerWallPaper}[2]{}
\newcommand*{\ThisURCornerWallPaper}[2]{}
\newcommand*{\LRCornerWallPaper}[2]{}
\newcommand*{\ThisLRCornerWallPaper}[2]{}
\newcommand*{\ClearWallPaper}{}
\newlength{\wpXoffset}
\newlength{\wpYoffset}
```

File 207 `lwarp-watermark.sty`

§ 292 Package watermark

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg `watermark` watermark is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{watermark}
```

```
\newcommand{\watermark}[1]{}
\newcommand{\leftwatermark}[1]{}
\newcommand{\rightwatermark}[1]{}
\newcommand{\thiswatermark}[1]{}
\newcommand{\thispageheading}[1]{}
```

File 208 `lwarp-wrapfig.sty`

§ 293 Package wrapfig

(*Emulates or patches code by* Donald Arseneau.)

Pkg `wrapfig` wrapfig is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{wrapfig}
```

```
\newcommand*{\LWR@wrapposition}{}

\newcommand*{\LWR@subwrapfigure}[2]{%
\renewcommand*{\LWR@wrapposition}{}%
```

```
\ifthenelse{%
    \equal{#1}{r}\OR\equal{#1}{R}\OR%
    \equal{#1}{o}\OR\equal{#1}{O}%
}%
{\renewcommand*{\LWR@wrapposition}{float:right}}%
{\renewcommand*{\LWR@wrapposition}{float:left}}%
\setlength{\LWR@templengthone}{#2}%
\uselengthunit{PT}%
\LWR@BlockClassWP{%
    width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition; %
    margin:10pt%
}%
{%
    width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition%
}%
{marginblock}%
}


\NewDocumentEnvironment{wrapfigure}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
\captionsetup{type=figure}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wraptable}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
\captionsetup{type=table}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wrapfloat}{m o m o m}
{%
\LWR@subwrapfigure{#3}{#5}%
\captionsetup{type=#1}%
}
{%
\endLWR@BlockClassWP%
}

\newlength{\wrapoverhang}
```

File 209 `lwarp-xcolor.sty`

§ 294 Package

xcolor

(*Emulates or patches code by* Dr. Uwe Kern.)

Pkg `xcolor` xcolor is supported by lwarp.

§ 294.1 Limitations

`\colorboxBlock` and `\fcolorboxBlock` and `\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by lwarp's core if xcolor is loaded.

background: `none` `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

color support Color definitions, models, and mixing are fully supported without any changes required.

tables Colored tables are ignored so far. Use CSS to style tables.

colored text and boxes `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

`\color` and `\pagecolor` `\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

§ 294.2 Xcolor definitions: location and timing

The lwarp core and its lwarp-xcolor package are tightly integrated to allow comparable results for print, HTML and print inside an HTML `lateximage`. This requires a number of definitions and redefintions depending on whether each of xcolor and `lateximage` is being used, and whether print or HTML is being generated. Some of these actions are one-time when xcolor is loaded, and others are temporary as `lateximage` is used.

When xcolor is loaded in print mode: No special actions are taken at the time that xcolor is loaded in print mode, but see `\AtBeginDocument` below.

When lwarp-xcolor is loaded in HTML mode: xcolor's original definitions are saved for later restoration. `\LWR@restoreorigformatting` is appended to restore

these definitions for use inside a lateximage. New HTML-mode definitions are created for \textcolor, \pagecolor, \nopagecolor, \colorbox, \colorboxBlock, \fcolorbox, \fcolorboxBlock, and fcolorminipage.

\AtBeginDocument in print or HTML mode: See Section 75. If xcolor has been loaded, the print-mode \fcolorbox is modified to accept a background color of none, and additional definitions are created for lwarp's new macros print-mode macros \colorboxBlock, \fcolorboxBlock, and fcolorminipage. The HTML versions of these macros will already have been created by lwarp-xcolor if it has been loaded.

For use inside an HTML lateximage, \LWR@restoreorigformatting is appended to temporarily set these functions to their print-mode versions.

In a lateximage in HTML mode: \LWR@restoreorigformatting temporarily restores the print-mode definitions of xcolor's functions. See \LWR@restoreorigformatting on page 382.

\color:

Print: Used as-is.

HTML: Ignored by pdftotext, and will not appear.

HTML lateximage: Colors will appear in a lateximage.

\textcolor:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 679.

HTML lateximage: Remembers and reuses the print version.

\pagecolor:

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a lateximage.

\nopagecolor:

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a lateximage.

\colorbox:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 679.

HTML lateximage: Remembers and reuses the print version.

`\colorboxBlock`:

Print: Becomes `\colorbox`.

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 680.

HTML `lateximage`: Remembers and reuses the print version `\colorbox`.

`\fcolorbox`:

Print: Modified to allow a background of `none`.
`\LWRprint@fcolorbox` at section 75

HTML: Redefined by lwarp-xcolor, page 680.

HTML `lateximage`: Remembers and reuses the print version.

`\fcolorboxBlock`:

Print: Becomes `\fcolorbox`. Section 75

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 681.

HTML `lateximage`: Remembers and reuses the print version `\fcolorbox`.

`fcolorminipage`:

Print: Newly defined in the lwarp core.
`LWRprint@fcolorminipage` at section 75

HTML: Newly defined by lwarp-xcolor, page 681.

HTML `lateximage`: Uses the print version.

`\boxframe`:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 682.

HTML `lateximage`: Remembers and reuses the print version.

§ 294.3 Package loading

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}
2 \begin{warpHTML}
```

§ 294.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
\LetLtxMacro\LWRprint@textcolor\textcolor
\LetLtxMacro\LWRprint@pagecolor\pagecolor
\LetLtxMacro\LWRprint@nopagecolor\nopagecolor
\LetLtxMacro\LWRprint@colorbox\colorbox
\LetLtxMacro\LWRprint@colorboxBlock\colorbox
\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
\LetLtxMacro\LWRorigprint@boxframe\boxframe
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
\appto{\LWR@restoreorigformatting}{%
\LetLtxMacro\textcolor\LWRprint@textcolor%
\LetLtxMacro\pagecolor\LWRprint@pagecolor%
\LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
\LetLtxMacro\colorbox\LWRprint@colorbox%
\LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
\LetLtxMacro\boxframe\LWRorigprint@boxframe%
}
```

§ 294.5 HTML color style

`\LWR@tempcolor` The color converted to HTML colorspace.

```
\newcommand*{\LWR@tempcolor}{}
\newcommand*{\LWR@tempcolortwo}{}
```

`\LWR@findcurrenttextcolor` Sets `\LWR@tempcolor` to the current color.

```
\newcommand*{\LWR@findcurrenttextcolor}{%
\protect\colorlet{LWR@current@color}{.}%
\protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
}
```

`\LWR@currenttextcolorstyle` Prints a color style for the current color.

```
\newcommand*{\LWR@currenttextcolorstyle}{%
\LWR@findcurrenttextcolor%
\ifdefstring{\LWR@tempcolor}{000000}%
{}%
{color: \#\LWR@tempcolor ; }%
}
```

`\LWR@textcurrentcolor` {⟨*text*⟩} Like `\textcolor` but uses the current `\color` instead.

```
\newcommand*{\LWR@textcurrentcolor}[1]{%
\begingroup%
\LWR@FBcancel%
\LWR@findcurrenttextcolor%
\InlineClass[color:\#\LWR@tempcolor]{textcolor}{%
    \renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
    #1%
}%
\endgroup%
}
```

`\LWR@colorstyle` `{`⟨*1: styletext*⟩`}` `{`⟨*2: model*⟩`}` `{`⟨*3: color*⟩`}`

For a color style, prints the color converted to HTML colors.

```
\NewDocumentCommand{\LWR@colorstyle}{m m}{%
\begingroup%
\LWR@FBcancel%
```

Use the xcolor package to convert to an HTML color space:

```
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
\#\LWR@tempcolor%
\endgroup%
}
```

`\LWR@backgroundcolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `{`⟨*text*⟩`}`

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```
\NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
#3%
}%
\endgroup%
}
```

§ 294.6 HTML border

\LWR@borderpadding {⟨*colorstyle*⟩} {⟨*color*⟩} Prints the HTML attributes for a black border and padding. \LWR@forceminwidth must be used first in order to set the border width.

```
\newcommand*{\LWR@borderpadding}[2]{%
\uselengthunit{PT}%
border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
}
```

§ 294.7 High-level macros

\color \color appears in the LaTeX PDF output, but is ignored by pdftotext and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is \let to the final name once the HTML conversion starts.

\textcolor [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex color span.

```
\RenewDocumentCommand{\textcolor}{O{named} m m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
#3%
}%
\endgroup%
}
```

\pagecolor [⟨*model*⟩] {⟨*color*⟩}

Ignored. Use \CSSFilename instead.

```
\renewcommand*{\pagecolor}[2][named]{}
```

\nopagecolor Ignored.

```
\renewcommand*{\nopagecolor}{}
```

\colorbox [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex background color .

```
\RenewDocumentCommand{\colorbox}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\InlineClass[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
]{colorbox}{#3}%
\endgroup%
}
```

`\colorboxBlock` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `{`⟨*text*⟩`}`

Converted into an HTML hex background color `<div>`.

```
\NewDocumentCommand{\colorboxBlock}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\begin{BlockClass}[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\rndprintlength{\fboxsep}%
]{colorboxBlock}
#3
\end{BlockClass}%
\endgroup%
}
```

`\fcolorbox` `[`⟨*framemodel*⟩`]` `{`⟨*framecolor*⟩`}` `[`⟨*boxmodel*⟩`]` `{`⟨*boxcolor*⟩`}` `{`⟨*text*⟩`}`

Converted into a framed HTML hex background color span.

A background color of `none` creates a colored frame without a background color.

```
\RenewDocumentCommand{\fcolorbox}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorbox #2 #4}%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \InlineClass[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorbox}{#5}%
}%
{% yes background color
```

```
    \InlineClass[%
    \LWR@borderpadding{#1}{#2} ; %
    background:\LWR@colorstyle{#3}{#4}%
    ]{fcolorbox}{#5}%
}%
\endgroup%
}
```

`\fcolorboxBlock` `[`⟨*frame model*⟩`]` `{`⟨*frame color*⟩`}` `[`⟨*box model*⟩`]` `{`⟨*box color*⟩`}` `{`⟨*text*⟩`}`

Converted into a framed HTML hex background color span.

A background color of `none` creates a colored frame without a background color.

```
\NewDocumentCommand{\fcolorboxBlock}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
\begingroup%
\LWR@FBcancel%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \begin{BlockClass}[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
    \end{BlockClass}%
}%
{% yes background color
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \begin{BlockClass}[%
    background:\#\LWR@tempcolortwo\ ; %
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
    \end{BlockClass}%
}%
\endgroup%
\LWR@traceinfo{HTML fcolorboxBlock done}%
}
```

Creates a framed HTML `<div>` around its contents.

A print-output version is defined in the lwarp core: section 75

`\LWR@subfcolorminipage` `{`⟨*frame model*⟩`}` `{`⟨*frame color*⟩`}` `{`⟨*background tag*⟩`}` `{`⟨*height*⟩`}`

```
\NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
\begin{BlockClass}[%
#3%
\LWR@borderpadding{#1}{#2} ; %
\IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
width:\rndprintlength{\LWR@tempwidth}%
]{fcolorminipage}%
}
```

Env `fcolorminipage` [⟨*1:framemodel*⟩] {⟨*2:framecolor*⟩} [⟨*3:boxmodel*⟩] {⟨*4:boxcolor*⟩} [⟨*5:align*⟩] [⟨*6:height*⟩] [⟨*7:inner-align*⟩] {⟨*8:width*⟩}

```
\NewDocumentEnvironment{fcolorminipage}{O{named} m O{named} m O{c} o o m}
{%
\LWR@FBcancel%
\setlength{\LWR@tempwidth}{#8}%
\IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
\ifthenelse{\equal{#4}{none}}%
{\LWR@subfcolorminipage{#1}{#2}{}{#6}}%
{%
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
}%
}
{\end{BlockClass}}
```

`\boxframe` {⟨*width*⟩} {⟨*height*⟩} {⟨*depth*⟩}

The depth is added to the height, but the box is not decended below by the depth. `\textcolor` is honored.

```
\renewcommand*{\boxframe}[3]{%
{%
\setlength{\LWR@tempwidth}{#1}%
\setlength{\LWR@tempheight}{#2}%
\addtolength{\LWR@tempheight}{#3}%
\uselengthunit{PT}%
\LWR@forceminwidth{\fboxrule}%
\InlineClass[%
display:inline-block ; %
border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
width:\rndprintlength{\LWR@tempwidth} ; %
height:\rndprintlength{\LWR@tempheight}%
]{boxframe}{}%
```

```
}%
}
```

```
\end{warpHTML}
```

File 210 **lwarp-xfrac.sty**

§ 295 Package

xfrac

(*Emulates or patches code by* The LaTeX3 Project.)

Pkg xfrac Supported by adding xfrac instances.

for HTML output:

```
\LWR@ProvidesPackagePass{xfrac}
```

⚠ font size In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by `pdftotext` to a plain text file. While the text may be in any font which `pdftotext` can read, the math is directly converted into SVG images using this same user-selected font. `xfrac` below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` `[`⟨*instance*⟩`]` `{`⟨*num*⟩`}` `[`⟨*sep*⟩`]` `{`⟨*denom*⟩`}`

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for HTML & PRINT:

```
\begin{warpall}
```

`\xfracHTMLfontsize` User-redefinable macro which controls the font size of the fraction.

```
\newcommand*{\xfracHTMLfontsize}{.6em}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

font size A span for a small font, used in the numerator and denominator:

```
\newcommand*{\LWR@htmlsmallfontstart}{%
```

```
\LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}%
\LWR@nestspan%
%
}

\newcommand*{\LWR@htmlsmallfontend}{%
\LWR@htmltagc{/span}%
\endLWR@nestspan%
}
```

\scalebox A nullified \scalebox command, to avoid introducing HTML scaling tags:

```
\NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}
```

instances Instances of xfrac for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that pdftotext correctly reads the result.

```
\DeclareInstance{xfrac}{default}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmr}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmss}{text}{
```

```
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmtt}{text}{
numerator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
denominator-format = {%
\LetLtxMacro{\scalebox}{\LWR@noscalebox}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
scaling = false
}


\end{warpHTML}
```

File 211 **lwarp-xltxtra.sty**

§ 296 Package

xltxtra

(*Emulates or patches code by* Will Robertson, Jonathan Kew.)

Pkg xltxtra xltxtra is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{xltxtra}


\RequirePackage{realscripts}
\RequirePackage{metalogo}
\newcommand*\TeX@logo@spacing[6]{}

\newcommand*{\vfrac}[2]{%
\textsuperscript{#1}/\textsubscript{#2}%
}

\newcommand\namedglyph[1]{%
```

```
\@tempcnta=\XeTeXglyphindex "#1"\relax
\ifnum\@tempcnta>0
  \XeTeXglyph\@tempcnta
\else
  \xxt@namedglyph@fallback{#1}%
\fi}

\newcommand\xxt@namedglyph@fallback[1]{[#1]}

\DeclareDocumentCommand{\showhyphens}{m}{}
```

File 212 `lwarp-xmpincl.sty`

§ 297 Package xmpincl

(*Emulates or patches code by* Maarten Sneep.)

Pkg `xmpincl` Emulated.

for HTML output: Discard all options for lwarp-xmpincl:

```
\LWR@ProvidesPackageDrop{xmpincl}

\newcommand*{\includexmp}[1]{}
```

File 213 `lwarp-xtab.sty`

§ 298 Package xtab

(*Emulates or patches code by* Peter Wilson.)

Pkg `xtab` xtab is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{xtab}
```

⚠ misplaced alignment
alignment tab character &

For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 8.8.

```
\newcommand{\LWRXT@firsthead}{}

\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRXT@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}

\newcommand{\tablelasthead}[1]{}

\newcommand{\notablelasthead}{}

\newcommand{\tabletail}[1]{}

\newcommand{\LWRXT@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRXT@lasttail{#1}%
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRXT@caption{\caption[#1]{#2}}%
}

\let\topcaption\tablecaption
\let\bottomcaption\tablecaption


\newcommand*{\LWRXT@caption}{}

\newcommand*{\shrinkheight}[1]{}

\newcommand*{\xentrystretch}[1]{}

\NewDocumentEnvironment{xtabular}{s o m}
{%
\LWR@traceinfo{xtabular}%
\table%
\LWRXT@caption%
\begin{tabular}{#3}%
\TabularMacro\ifdefvoid{\LWRXT@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
}%
{%
\ifdefvoid{\LWRXT@lasttail}%
{}%
{%
\TabularMacro\ResumeTabular%
```

```
\LWRXT@lasttail%
}%
\end{tabular}%
\endtable%
\LWR@traceinfo{xtabular done}%
}

\NewDocumentEnvironment{mpxtabular}{s o m}
{\minipage{\linewidth}\xtabular{#3}}
{\endxtabular\endminipage}
```

File 214 `lwarp-zwpagelayout.sty`

§ 299 Package zwpagelayout

(*Emulates or patches code by* Zdeněk Wagner.)

Pkg `zwpagelayout` zwpagelayout is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{zwpagelayout}

\def\noBboxes{}
\@onlypreamble\noBboxes

\expandafter\ifx\csname definecolor\endcsname\relax \else
  \definecolor{cmykblack}{cmyk}{0,0,0,1}
  \definecolor{grblack}{gray}{0}
%   \ifzwpl@redefineblack
%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
%   \fi
  \definecolor{cmykred}{cmyk}{0,1,1,0}
  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
  \definecolor{cmykblue}{cmyk}{1,1,0,0}
  \definecolor{rgbred}{rgb}{1,0,0}
  \definecolor{rgbgreen}{rgb}{0,1,0}
  \definecolor{rgbblue}{rgb}{0,0,1}
%   \ifzwpl@redefinetocmyk
%     \definecolor{red}{cmyk}{0,1,1,0}
%     \definecolor{green}{cmyk}{1,0,1,0}
%     \definecolor{blue}{cmyk}{1,1,0,0}
%   \fi
\fi

\let\OverprintXeTeXExtGState\relax

\DeclareRobustCommand\SetOverprint{\ignorespaces}
\DeclareRobustCommand\SetKnockout{\ignorespaces}
```

```
\DeclareRobustCommand\textoverprint[1]{{\SetOverprint#1}}
\DeclareRobustCommand\textknockout[1]{{\SetKnockout#1}}

\def\SetPDFminorversion#1{}
\@onlypreamble\SetPDFminorversion

\newcommand*\Vcorr{}

\DeclareRobustCommand\vb[1][]{}
\NewDocumentCommand{\NewOddPage}{* o}{}
\NewDocumentCommand{\NewEvenPage}{* o}{}
\def\SetOddPageMessage#{\gdef\ZW@oddwarning}
\def\SetEvenPageMessage#{\gdef\Z@@evenwarning}
\def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning

\def\clap#1{#1}

\def\CropFlap{2in}
\def\CropSpine{1in}
\def\CropXSpine{1in}
\def\CropXtrim{.25in}
\def\CropYtrim{.25in}
\def\UserWidth{5in}
\def\UserLeftMargin{1in}
\def\UserRightMargin{1in}
\def\UserTopMargin{1in}
\def\UserBotMargin{1in}
\def\thePageNumber{\#\,\arabic{page}}
\ifXeTeX
\def\ifcaseZWdriver{\ifcase2}
\else
\def\ifcaseZWdriver{\ifcase1}
\fi
\DeclareRobustCommand\ZWifdriver[2]{}
```

File 215 `lwarp-patch-komascript.sty`

§ 300 Package

patch-komascript

Pkg `lwarp-patch-komascript`

Patches for komascript classes.

lwarp loads this package when scrbook, scrartcl, or scrreprt classes are detected.

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

\titlehead, \subject, \captionformat, \figureformat, and \tableformat are not yet emulated.

⚠ Not fully tested! Please send bug reports!

Some features have not yet been tested. Please contact the author with any bug reports.

for HTML output:

```
\ProvidesPackage{lwarp-patch-komascript}
```

typearea is emulated.

```
\RequirePackage{lwarp-typearea}
```

tocbasic is emulated.

```
\RequirePackage{lwarp-tocbasic}
```

scrextend patches most of the new macros.

```
\RequirePackage{lwarp-scrextend}
```

The \minisec is placed inside a of class minisec.

```
\renewcommand*{\minisec}[1]{
\begin{BlockClass}{minisec}
#1
\end{BlockClass}
}
```

The part and chapter preambles are placed as plain text just after each heading.

```
\@ifundefined{setpartpreamble}{}{
\RenewDocumentCommand{\setpartpreamble}{o o +m}{%
\renewcommand{\part@preamble}{#3}%
}
}

\@ifundefined{setchapterpreamble}{}{
\RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
\renewcommand{\chapter@preamble}{#3}%
}
}
```

Simple captions are used in all cases.

```
\LetLtxMacro\captionbelow\caption
\LetLtxMacro\captionabove\caption
```

```
23
24 \LetLtxMacro\captionofbelow\captionof
25 \LetLtxMacro\captionofabove\captionof
26
27 \RenewDocumentEnvironment{captionbeside}{o m o o o s}
28 {}
29 {%
30 \IfValueTF{#1}%
31 {\caption[#1]{#2}}%
32 {\caption{#2}}%
33 }
34
35 \RenewDocumentEnvironment{captionofbeside}{m o m o o o s}
36 {}
37 {%
38 \IfValueTF{#2}%
39 {\captionof{#1}[#2]{#3}}%
40 {\captionof{#1}{#3}}%
41 }
42
43 \RenewDocumentCommand{\setcapindent}{s m}{}
44 \renewcommand*{\setcaphanging}{}
45 \renewcommand*{\setcapwidth}[2][]{}
46 \renewcommand*{\setcapdynwidth}[2][]{}
47 \RenewDocumentCommand{\setcapmargin}{s o m}{}
```

File 216 `lwarp-patch-memoir.sty`

§ 301 Package

patch-memoir

(*Emulates or patches code by* Peter Wilson.)

Pkg `lwarp-patch-memoir` Patches for memoir class.

⚠ Not fully tested! Please send bug reports!

lwarp loads this package when the memoir class is detected.

⚠ options clash While emulating memoir, lwarp pre-loads a number of packages (section 301.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading lwarp:

```
\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}
```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

lwarp loads pagenote to perform memoir's pagenote functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the memoir enhancements. It is currently recommended to load and use fancyvrb instead.

The memoir glossary system is not yet supported by `lwarpmk`. The glossaries package may be used instead, but does require the glossary entries be changed from the memoir syntax to the glossaries syntax.

for HTML output:

```
\ProvidesPackage{lwarp-patch-memoir}
```

§ 301.1 Packages

These are pre-loaded to provide emulation for many of memoir's functions. memoir pretends that abstract, etc. are already loaded, via its "emulated" package mechanism, but lwarp is directly loading the "`lwarp-`" version of each, which happens to avoid memoir's emulation system.

```
\RequirePackage{lwarp-abstract}% req'd
\RequirePackage{lwarp-array}% req'd
\RequirePackage{lwarp-booktabs}% req'd
% \RequirePackage{lwarp-ccaption}% to do
\RequirePackage{lwarp-changepage}% req'd
\RequirePackage{lwarp-crop}
\RequirePackage{lwarp-dcolumn}% req'd
\RequirePackage{lwarp-enumerate}% req'd
\RequirePackage{lwarp-epigraph}% req'd
\RequirePackage{lwarp-fancyvrb}% req'd
\RequirePackage{lwarp-footmisc}% req'd
\RequirePackage{lwarp-framed}% req'd
\RequirePackage{lwarp-hanging}% req'd
\DisemulatePackage{moreverb}
\RequirePackage{lwarp-moreverb}
\RequirePackage{lwarp-mparhack}
```

```
\RequirePackage{lwarp-needspace}% req'd
\RequirePackage{lwarp-nextpage}% req'd
\RequirePackage{lwarp-pagenote}% req'd
\RequirePackage{lwarp-parskip}
\RequirePackage{lwarp-setspace}% req'd
\RequirePackage{lwarp-showidx}
\RequirePackage{lwarp-subfigure}% red'q
```

subfigure is emulated via subfig, which pre-defines `subfigure` and `subtable`, but memoir does not, so they must be tested for here:

```
\LetLtxMacro\LWR@memorignewsubfloat\newsubfloat
\RenewDocumentCommand{\newsubfloat}{O{} m}{%
    \@ifundefined{c@sub#2}{%
        \LWR@memorignewsubfloat[#1]{#2}%
    }{}%
}

\RequirePackage{lwarp-tabularx}% req'd
\RequirePackage{lwarp-titling}% req'd
% \RequirePackage{lwarp-tocbibind}% not emulated by memoir
\RequirePackage{lwarp-tocloft}% req'd
\RequirePackage{lwarp-verse}% req'd
```

§ 301.2 Preliminary setup

Bypass the memoir package mechanism:

```
\LetLtxMacro\LWR@origlabel\@mem@old@label
```

Redefined to write the `LWR@autoindex` counter instead of `page`

```
\AtBeginDocument{
    \def\@@wrindexhyp#1||\\{%
        \addtocounter{LWR@autoindex}{1}%
        \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}%
      \ifshowindexmark\@showidx{#1}\fi
      \protected@write\@auxout{}%
        {\string\@@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}}%
      \endgroup
      \@esphack}%
}
```

memoir already set the page size to a default, so it must be forced large for lwarp's use, to avoid tag overflows off the page.

```
\setstocksize{190in}{20in}
\setlrmarginsandblock{2in}{2in}{*}
\setulmarginsandblock{1in}{1in}{*}
```

§ 301.3 Laying out the page

```
\renewcommand*{\stockavi}{}
\renewcommand*{\stockav}{}
\renewcommand*{\stockaiv}{}
\renewcommand*{\stockaiii}{}
\renewcommand*{\stockbvi}{}
\renewcommand*{\stockbv}{}
\renewcommand*{\stockbiv}{}
\renewcommand*{\stockbiii}{}
% \renewcommand*{\stockmetriccrownvo}{}% in docs but not in the package
\renewcommand*{\stockmlargecrownvo}{}
\renewcommand*{\stockmdemyvo}{}
\renewcommand*{\stockmsmallroyalvo}{}
\renewcommand*{\pageavi}{}
\renewcommand*{\pageav}{}
\renewcommand*{\pageaiv}{}
\renewcommand*{\pageaiii}{}
\renewcommand*{\pagebvi}{}
\renewcommand*{\pagebv}{}
\renewcommand*{\pagebiv}{}
\renewcommand*{\pagebiii}{}
% \renewcommand*{\pagemetriccrownvo}{}% in docs but not in the package
\renewcommand*{\pagemlargecrownvo}{}
\renewcommand*{\pagemdemyvo}{}
\renewcommand*{\pagemsmallroyalvo}{}

\renewcommand*{\stockdbill}{}
\renewcommand*{\stockstatement}{}
\renewcommand*{\stockexecutive}{}
\renewcommand*{\stockletter}{}
\renewcommand*{\stockold}{}
\renewcommand*{\stocklegal}{}
\renewcommand*{\stockledger}{}
\renewcommand*{\stockbroadsheet}{}
\renewcommand*{\pagedbill}{}
\renewcommand*{\pagestatement}{}
\renewcommand*{\pageexecutive}{}
\renewcommand*{\pageletter}{}
\renewcommand*{\pageold}{}
\renewcommand*{\pagelegal}{}
\renewcommand*{\pageledger}{}
\renewcommand*{\pagebroadsheet}{}

\renewcommand*{\stockpottvo}{}
\renewcommand*{\stockfoolscapvo}{}
\renewcommand*{\stockcrownvo}{}
\renewcommand*{\stockpostvo}{}
\renewcommand*{\stocklargecrownvo}{}
\renewcommand*{\stocklargepostvo}{}
```

```
\renewcommand*{\stocksmalldemyvo}{}
\renewcommand*{\stockdemyvo}{}
\renewcommand*{\stockmediumvo}{}
\renewcommand*{\stocksmallroyalvo}{}
\renewcommand*{\stockroyalvo}{}
\renewcommand*{\stocksuperroyalvo}{}
\renewcommand*{\stockimperialvo}{}
\renewcommand*{\pagepottvo}{}
\renewcommand*{\pagefoolscapvo}{}
\renewcommand*{\pagecrownvo}{}
\renewcommand*{\pagepostvo}{}
\renewcommand*{\pagelargecrownvo}{}
\renewcommand*{\pagelargepostvo}{}
\renewcommand*{\pagesmalldemyvo}{}
\renewcommand*{\pagedemyvo}{}
\renewcommand*{\pagemediumvo}{}
\renewcommand*{\pagesmallroyalvo}{}
\renewcommand*{\pageroyalvo}{}
\renewcommand*{\pagesuperroyalvo}{}
\renewcommand*{\pageimperialvo}{}

\renewcommand*{\memfontfamily}{}
\renewcommand*{\memfontenc}{}
\renewcommand*{\memfontpack}{}

\renewcommand*{\anyptfilebase}{}
\renewcommand*{\anyptsize}{10}

\renewcommand*{\setstocksize}[2]{}
\renewcommand*{\settrimmedsize}[3]{}
\renewcommand*{\settrims}[2]{}

% \newlength{\lxvchars}
% \setlength{\lxvchars}{305pt}
% \newlength{\xlvchars}
% \setlength{\xlvchars}{190pt}
\renewcommand*{\setxlvchars}[1]{}
\renewcommand*{\setlxvchars}[1]{}

\renewcommand*{\settypeblocksize}[3]{}
\renewcommand*{\setlrmargins}[3]{}
\renewcommand*{\setlrmarginsandblock}[3]{}
\renewcommand*{\setbinding}[1]{}
\renewcommand*{\setulmargins}[3]{}
\renewcommand*{\setulmarginsandblock}[3]{}
\renewcommand*{\setcolsepandrule}[2]{}

\renewcommand*{\setheadfoot}[2]{}
\renewcommand*{\setheaderspaces}[3]{}
\renewcommand*{\setmarginnotes}[3]{}
```

```
\renewcommand*{\setfootins}[2]{}
\renewcommand*{\checkandfixthelayout}[1][]{}
\renewcommand*{\checkthelayout}[1]{}
\renewcommand*{\fixthelayout}{}
%
% \newlength{\stockheight}
% \newlength{\trimtop}
% \newlength{\trimedge}
% \newlength{\stockwidth}
% \newlength{\spinemargin}
% \newlength{\foremargin}
% \newlength{\uppermargin}
% \newlength{\headmargin}
%
\renewcommand*{\typeoutlayout}{}
\renewcommand*{\typeoutstandardlayout}{}
\renewcommand*{\settypeoutlayoutunit}[1]{}
\renewcommand*{\fixpdflayout}{}
\renewcommand*{\fixdvipslayout}{}

\renewcommand*{\medievalpage}[1][]{}
\renewcommand*{\isopage}[1][]{}
\renewcommand*{\semiisopage}[1][]{}

\renewcommand{\setpagebl}[3]{}
\renewcommand{\setpageml}[3]{}
\renewcommand{\setpagetl}[3]{}
\renewcommand{\setpagetm}[3]{}
\renewcommand{\setpagetr}[3]{}
\renewcommand{\setpagemr}[3]{}
\renewcommand{\setpagebr}[3]{}
\renewcommand{\setpagebm}[3]{}
\renewcommand{\setpagecc}[3]{}
```

§ 301.4 Text and fonts

```
\let\miniscule\tiny
\let\HUGE\Huge

\renewcommand*{\abnormalparskip}[1]{}
\renewcommand*{\nonzeroparskip}{}
\renewcommand*{\traditionalparskip}{}

\let\onelineskip\baselineskip

\let\OnehalfSpacing\onehalfspacing
\let\DoubleSpacing\doublespacing
\renewcommand*{\setPagenoteSpacing}[1]{}
\renewcommand*{\setFloatSpacing}[1]{}
```

```
\let\SingleSpacing\singlespacing
\let\setSingleSpace\SetSinglespace
\let\SingleSpace\singlespace
\let\endSingleSpace\endsinglespace
\let\Spacing\spacing
\let\endSpacing\endspacing
\let\OnehalfSpace\onehalfspace
\let\endOnehalfSpace\endonehalfspace
\csletcs{OnehalfSpace*}{onehalfspace}
\csletcs{endOnehalfSpace*}{endonehalfspace}
\let\DoubleSpace\doublespace
\let\endDoubleSpace\enddoublespace
\csletcs{DoubleSpace*}{doublespace}
\csletcs{endDoubleSpace*}{enddoublespace}
\renewcommand*{\setDisplayskipStretch}[1]{}
\renewcommand*{\memdskipstretch}{}
\renewcommand*{\noDisplayskipStretch}{}
\renewcommand*{\memdskips}{}

\renewcommand*{\midsloppy}{}
\renewenvironment*{midsloppypar}{}{}

\renewcommand*{\sloppybottom}{}
```

§ 301.5 Titles

```
\csletcs{titlingpage*}{titlingpage}
\csletcs{endtitlingpage*}{endtitlingpage}
\let\andnext\and
\renewcommand*{\thanksmarkstyle}[1]{}
\renewcommand{\thanksfootmark}{%
\thanksscript{\tamark}%
}

% \newlength{\thanksmarksep}
```

§ 301.6 Abstracts

```
\renewcommand*{\abstractcol}{}
\renewcommand*{\abstractintoc}{}
\renewcommand*{\abstractnum}{}
\renewcommand*{\abstractrunin}{}
```

§ 301.7 Docment divisions

```

\def\@apppage{%
    \part*{\appendixpagename}
}
```

```
\renewcommand\mempreaddapppagetotochook{}
\renewcommand\mempostaddapppagetotochook{}

\def\@sapppage{%
    \part*{\appendixpagename}
}

\csletcs{frontmatter*}{frontmatter}
\csletcs{mainmatter*}{mainmatter}
\renewcommand*{\raggedbottomsection}{}
\renewcommand*{\normalbottomsection}{}
\renewcommand*{\bottomsectionskip}{}
\renewcommand*{\bottomsectionpenalty}{}
\csletcs{appendixpage*}{appendixpage}
\renewcommand*{\namedsubappendices}{}
\renewcommand*{\unnamedsubappendices}{}
\renewcommand*{\setsecnumdepth}[1]{}% todo tocvsec2
\renewcommand*{\maxsecnumdepth}[1]{}% todo tocvsec2
\renewcommand*{\beforebookskip}{}
\renewcommand*{\afterbookskip}{}
\renewcommand*{\beforepartskip}{}
\renewcommand*{\afterpartskip}{}
\renewcommand*{\midbookskip}{}
\renewcommand*{\midpartskip}{}
\renewcommand*{\printbookname}{}
\renewcommand*{\booknamefont}{}
\renewcommand*{\booknamenum}{}
\renewcommand*{\printbooknum}{}
\renewcommand*{\booknumfont}{}
\renewcommand*{\printpartname}{}
\renewcommand*{\partnamefont}{}
\renewcommand*{\partnamenum}{}
\renewcommand*{\printpartnum}{}
\renewcommand*{\partnumfont}{}
\renewcommand*{\printbooktitle}[1]{}
\renewcommand*{\booktitlefont}{}
\renewcommand{\printparttitle}[1]{}
\renewcommand*{\parttitlefont}{}
\renewcommand*{\bookpageend}{}
\renewcommand*{\bookblankpage}{}
\renewcommand*{\nobookblankpage}{}
\renewcommand*{\partpageend}{}
\renewcommand*{\partblankpage}{}
\renewcommand*{\nopartblankpage}{}
\RenewDocumentCommand{\newleadpage}{s o m m}{}% todo
\RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo
\renewcommand*{\leadpagetoclevel}{chapter}

\renewcommand*{\openright}{}
\renewcommand*{\openleft}{}
```

```
\renewcommand*{\openany}{}
\renewcommand*{\clearforchapter}{}
\renewcommand*{\memendofchapterhook}{}
\renewcommand*{\chapterheadstart}{}
% \newlength{\beforechapskip}
\renewcommand*{\afterchapternum}{}
% \newlength{\midchapskip}
\renewcommand*{\afterchaptertitle}{}
% \newlength{\afterchapskip}
\renewcommand*{\printchaptername}{}
\renewcommand*{\chapnamefont}{}
\renewcommand*{\chapternamenum}{}
\renewcommand*{\printchapternum}{}
\renewcommand*{\chapnumfont}{}
\renewcommand{\printchaptertitle}[1]{}
\renewcommand*{\chaptitlefont}{}
\renewcommand*{\printchapternonum}{}
\renewcommand*{\indentafterchapter}{}
\renewcommand*{\noindentafterchapter}{}
\renewcommand*{\insertchapterspace}{}

\renewcommand*{\chapterstyle}[1]{}
\renewcommand{\makechapterstyle}[2]{}
\renewcommand*{\chapindent}{}
\let\chapterprecis\cftchapterprecis
\let\chapterprecishere\cftchapterprecishere
\let\chapterprecistoc\cftchapterprecistoc
\renewcommand*{\precisfont}{}
\renewcommand*{\prechapterprecis}{}
\renewcommand*{\postchapterprecis}{}
\renewcommand{\precistoctext}[1]{}
\renewcommand*{\precistocfont}{}
\renewcommand*{\precistocformat}{}
% \newlength{\prechapterprecisshift}

\renewcommand*{\setbeforesecskip}[1]{}
\renewcommand*{\setaftersecskip}[1]{}
\renewcommand*{\setsecindent}[1]{}
\renewcommand*{\setsecheadstyle}[1]{}
\renewcommand*{\setbeforesubsecskip}[1]{}
\renewcommand*{\setaftersubsecskip}[1]{}
\renewcommand*{\setsubsecindent}[1]{}
\renewcommand*{\setsubsecheadstyle}[1]{}
\renewcommand*{\setbeforesubsubsecskip}[1]{}
\renewcommand*{\setaftersubsubsecskip}[1]{}
\renewcommand*{\setsubsubsecindent}[1]{}
\renewcommand*{\setsubsubsecheadstyle}[1]{}
\renewcommand*{\setbeforeparaskip}[1]{}
\renewcommand*{\setafterparaskip}[1]{}
\renewcommand*{\setparaindent}[1]{}
```

```
\renewcommand*{\setparaheadstyle}[1]{}
\renewcommand*{\setbeforesubparaskip}[1]{}
\renewcommand*{\setaftersubparaskip}[1]{}
\renewcommand*{\setsubparaindent}[1]{}
\renewcommand*{\setsubparaheadstyle}[1]{}
\renewcommand{\@hangfrom}[1]{#1}
\renewcommand{\sethangfrom}[1]{}
\renewcommand{\setsecnumformat}[1]{}

\renewcommand*{\hangsecnum}{}
\renewcommand*{\defaultsecnum}{}

\renewcommand*{\sechook}{}
\renewcommand{\setsechook}[1]{}
\renewcommand*{\subsechook}{}
\renewcommand{\setsubsechook}[1]{}
\renewcommand*{\subsubsechook}{}
\renewcommand{\setsubsubsechook}[1]{}
\renewcommand*{\parahook}{}
\renewcommand{\setparahook}[1]{}
\renewcommand*{\subparahook}{}
\renewcommand{\setsubparahook}[1]{}

\RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}

\RenewDocumentCommand{\fancybreak}{s +m}{%
\begin{center}#2\end{center}%
}

\RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
\begin{center}#4\end{center}%
}

\RenewDocumentCommand{\pfbreak}{s}{%
\begin{center}
\pfbreakdisplay
\end{center}
}

% \newlength{\pfbreakskip}
\renewcommand{\pfbreakdisplay}{*\quad*\quad*}

\renewcommand{\makeheadstyles}[2]{}
\renewcommand*{\headstyles}[1]{}
```

§ 301.8 Pagination and headers

```
\renewcommand*{\savepagenumber}{}
\renewcommand*{\restorepagenumber}{}
```

```
\renewcommand*{\uppercaseheads}{}
\renewcommand*{\nouppercaseheads}{}

\renewcommand*{\bookpagemark}[1]{}
\renewcommand*{\partmark}[1]{}
\renewcommand*{\bibmark}{}
\renewcommand*{\indexmark}{}
\renewcommand*{\glossarymark}{}

\LWR@origpagestyle{empty}
\renewcommand*{\ps@empty}{}
\renewcommand*{\makepagestyle}[1]{}
\renewcommand*{\emptypshook}{}%
% \renewcommand*{\empty@oddhead}{}
% \renewcommand*{\empty@oddfoot}{}
% \renewcommand*{\empty@evenhead}{}
% \renewcommand*{\empty@evenfoot}{}
\renewcommand*{\@oddhead}{}
\renewcommand*{\@oddfoot}{}
\renewcommand*{\@evenhead}{}
\renewcommand*{\@evenfoot}{}
\renewcommand*{\aliaspagestyle}[2]{}
\renewcommand*{\copypagestyle}[2]{}

\renewcommand*{\makeevenhead}[4]{}
\renewcommand*{\makeoddhead}[4]{}
\renewcommand*{\makeevenfoot}[4]{}
\renewcommand*{\makeoddfoot}[4]{}
\renewcommand*{\makerunningwidth}[3]{}
% \newlength{\headwidth}
\renewcommand*{\makeheadrule}[3]{}
\renewcommand*{\makefootrule}[3]{}
\renewcommand*{\makeheadfootruleprefix}[3]{}
% \newlength{\normalrulethickness}
% \setlength{\normalrulethickness}{.4pt}
% \newlength{\footruleheight}
% \newlength{\footruleskip}
\renewcommand*{\makeheadposition}[5]{}
\renewcommand{\makepsmarks}[2]{}
\renewcommand*{\makeheadfootstrut}[3]{}

\renewcommand{\createplainmark}[3]{}
\renewcommand{\memUChead}[1]{}
\renewcommand{\createmark}[5]{}
\renewcommand*{\clearplainmark}[1]{}
\renewcommand*{\clearmark}[1]{}
\renewcommand{\addtopsmarks}[3]{}
\renewcommand{\ifonlyfloats}[2]{#2}
\renewcommand*{\mergepagefloatstyle}[3]{}

```

```
\renewcommand*{\framepichead}{}
\renewcommand*{\framepictextfoot}{}
\renewcommand*{\framepichook}{}
\renewcommand*{\showheadfootlocoff}{}
\renewcommand*{\showtextblocklocoff}{}
```

§ 301.9 Paragraphs and lists

```
\renewcommand{\hangfrom}[1]{#1}
\let\centerfloat\centering
\renewcommand*{\raggedyright}[1][]{}
% \newlength{\ragrparindent}
\renewcommand{\sourceatright}[2][]{\attribution{#2}}
\let\memorigdbs\LWR@endofline
\let\memorigpar\par
\let\atcentercr\LWR@endofline
\renewcommand*{\flushleftright}{}
\renewcommand*{\linenottooshort}[1][]{}
\renewcommand*{\russianpar}{}
\renewcommand*{\lastlinerulefill}{}
\renewcommand*{\lastlineparrule}{}
\renewcommand*{\justlastraggedleft}{}
\renewcommand*{\raggedrightthenleft}{}
\renewcommand*{\leftcenterright}{}

\renewcommand{\leftspringright}[4]{%
\begin{minipage}{#1\linewidth}#3\end{minipage}\qquad%
\begin{minipage}{#2\linewidth}\begin{flushright}#4\end{flushright}\end{minipage}%
}

\renewenvironment*{blockdescription}
{\LWR@descriptionstart\LWR@origdescription}
{\enddescription}
\renewcommand*{\blockdescriptionlabel}[1]{\textbf{#1}}
\renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
\renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
\renewcommand*{\tightlists}{}
\renewcommand*{\defaultlists}{}
\RenewDocumentCommand{\firmlists}{s}{}
\renewcommand*{\firmlist}{}
\renewcommand*{\tightlist}{}
\renewcommand*{\zerotrivseps}{}
\renewcommand*{\savetrivseps}{}
\renewcommand*{\restoretrivseps}{}
```

§ 301.10 Contents lists

```
\csletcs{tableofcontents*}{tableofcontents}
\csletcs{listoffigures*}{listoffigures}
```

```
\csletcs{listoftables*}{listoftables}
\renewenvironment{KeepFromToc}{}{}
\renewcommand*{\onecoltocetc}{}
\renewcommand*{\twocoltocetc}{}
\renewcommand*{\ensureonecol}{}
\renewcommand*{\restorefromonecol}{}
\renewcommand*{\doccoltocetc}{}
\renewcommand*{\maxtocdepth}[1]{}% tocvsec2
\renewcommand*{\settocdepth}[1]{}% tocvsec2

\renewcommand{\tocheadstart}{}
\renewcommand{\printtoctitle}[1]{}
\renewcommand{\tocmark}{}
\renewcommand{\aftertoctitle}{}
\renewcommand{\lofheadstart}{}
\renewcommand{\printloftitle}[1]{}
\renewcommand{\lofmark}{}
\renewcommand{\afterloftitle}{}
\renewcommand{\lotheadstart}{}
\renewcommand{\printlottitle}[1]{}
\renewcommand{\lotmark}{}
\renewcommand{\afterlottitle}{}

\renewcommand*{\setpnumwidth}[1]{}
\renewcommand*{\setrmarg}[1]{}
\renewcommand*{\cftbookbreak}{}
\renewcommand*{\cftpartbreak}{}
\renewcommand*{\cftchapterbreak}{}

% \newlength{\cftbeforebookskip}
% \newlength{\cftbookindent}
% \newlength{\cftbooknumwidth}
\renewcommand*{\cftbookfont}{}
\renewcommand*{\cftbookname}{}
\renewcommand*{\cftbookpresnum}{}
\renewcommand*{\cftbookaftersnum}{}
\renewcommand*{\cftbookaftersnumb}{}
\renewcommand*{\cftbookleader}{}
\renewcommand*{\cftbookdotsep}{1}
\renewcommand*{\cftbookpagefont}{}
\renewcommand*{\cftbookafterpnum}{}
\renewcommand*{\cftbookformatpnum}[1]{}
\renewcommand*{\cftbookformatpnumhook}[1]{}
```

Part is already defined by tocloft.

```
% \newlength{\cftbeforechapterskip}
% \newlength{\cftchapterindent}
% \newlength{\cftchapternumwidth}
\renewcommand*{\cftchapterfont}{}
```

```
\renewcommand*{\cftchaptername}{}
\renewcommand*{\cftchapterpresnum}{}
\renewcommand*{\cftchapteraftersnum}{}
\renewcommand*{\cftchapteraftersnumb}{}
\renewcommand*{\cftchapterleader}{}
\renewcommand*{\cftchapterdotsep}{1}
\renewcommand*{\cftchapterpagefont}{}
\renewcommand*{\cftchapterafterpnum}{}
\renewcommand*{\cftchapterformatpnum}[1]{}
\renewcommand*{\cftchapterformatpnumhook}[1]{}

% \newlength{\cftbeforesectionskip}
% \newlength{\cftsectionindent}
% \newlength{\cftsectionnumwidth}
\renewcommand*{\cftsectionfont}{}
\renewcommand*{\cftsectionname}{}
\renewcommand*{\cftsectionpresnum}{}
\renewcommand*{\cftsectionaftersnum}{}
\renewcommand*{\cftsectionaftersnumb}{}
\renewcommand*{\cftsectionleader}{}
\renewcommand*{\cftsectiondotsep}{1}
\renewcommand*{\cftsectionpagefont}{}
\renewcommand*{\cftsectionafterpnum}{}
\renewcommand*{\cftsectionformatpnum}[1]{}
\renewcommand*{\cftsectionformatpnumhook}[1]{}

% \newlength{\cftbeforesubsectionskip}
% \newlength{\cftsubsectionindent}
% \newlength{\cftsubsectionnumwidth}
\renewcommand*{\cftsubsectionfont}{}
\renewcommand*{\cftsubsectionname}{}
\renewcommand*{\cftsubsectionpresnum}{}
\renewcommand*{\cftsubsectionaftersnum}{}
\renewcommand*{\cftsubsectionaftersnumb}{}
\renewcommand*{\cftsubsectionleader}{}
\renewcommand*{\cftsubsectiondotsep}{1}
\renewcommand*{\cftsubsectionpagefont}{}
\renewcommand*{\cftsubsectionafterpnum}{}
\renewcommand*{\cftsubsectionformatpnum}[1]{}
\renewcommand*{\cftsubsectionformatpnumhook}[1]{}

% \newlength{\cftbeforesubsubsectionskip}
% \newlength{\cftsubsubsectionindent}
% \newlength{\cftsubsubsectionnumwidth}
\renewcommand*{\cftsubsubsectionfont}{}
\renewcommand*{\cftsubsubsectionname}{}
\renewcommand*{\cftsubsubsectionpresnum}{}
\renewcommand*{\cftsubsubsectionaftersnum}{}
\renewcommand*{\cftsubsubsectionaftersnumb}{}
\renewcommand*{\cftsubsubsectionleader}{}
```

```
\renewcommand*{\cftsubsubsectiondotsep}{1}
\renewcommand*{\cftsubsubsectionpagefont}{}
\renewcommand*{\cftsubsubsectionafterpnum}{}
\renewcommand*{\cftsubsubsectionformatpnum}[1]{}
\renewcommand*{\cftsubsubsectionformatpnumhook}[1]{}

% \newlength{\cftbeforeparagraphskip}
% \newlength{\cftparagraphindent}
% \newlength{\cftparagraphnumwidth}
\renewcommand*{\cftparagraphfont}{}
\renewcommand*{\cftparagraphname}{}
\renewcommand*{\cftparagraphpresnum}{}
\renewcommand*{\cftparagraphaftersnum}{}
\renewcommand*{\cftparagraphaftersnumb}{}
\renewcommand*{\cftparagraphleader}{}
\renewcommand*{\cftparagraphdotsep}{1}
\renewcommand*{\cftparagraphpagefont}{}
\renewcommand*{\cftparagraphafterpnum}{}
\renewcommand*{\cftparagraphformatpnum}[1]{}
\renewcommand*{\cftparagraphformatpnumhook}[1]{}

% \newlength{\cftbeforesubparagraphskip}
% \newlength{\cftsubparagraphindent}
% \newlength{\cftsubparagraphnumwidth}
\renewcommand*{\cftsubparagraphfont}{}
\renewcommand*{\cftsubparagraphname}{}
\renewcommand*{\cftsubparagraphpresnum}{}
\renewcommand*{\cftsubparagraphaftersnum}{}
\renewcommand*{\cftsubparagraphaftersnumb}{}
\renewcommand*{\cftsubparagraphleader}{}
\renewcommand*{\cftsubparagraphdotsep}{1}
\renewcommand*{\cftsubparagraphpagefont}{}
\renewcommand*{\cftsubparagraphafterpnum}{}
\renewcommand*{\cftsubparagraphformatpnum}[1]{}
\renewcommand*{\cftsubparagraphformatpnumhook}[1]{}

% \newlength{\cftbeforefigureskip}
% \newlength{\cftfigureindent}
% \newlength{\cftfigurenumwidth}
\renewcommand*{\cftfigurefont}{}
\renewcommand*{\cftfigurename}{}
\renewcommand*{\cftfigurepresnum}{}
\renewcommand*{\cftfigureaftersnum}{}
\renewcommand*{\cftfigureaftersnumb}{}
\renewcommand*{\cftfigureleader}{}
\renewcommand*{\cftfiguredotsep}{1}
\renewcommand*{\cftfigurepagefont}{}
\renewcommand*{\cftfigureafterpnum}{}
\renewcommand*{\cftfigureformatpnum}[1]{}
\renewcommand*{\cftfigureformatpnumhook}[1]{}
```

```
% \newlength{\cftbeforesubfigureskip}
% \newlength{\cftsubfigureindent}
% \newlength{\cftsubfigurenumwidth}
\newcommand*{\cftsubfigurefont}{}
\newcommand*{\cftsubfigurename}{}
\newcommand*{\cftsubfigurepresnum}{}
\newcommand*{\cftsubfigureaftersnum}{}
\newcommand*{\cftsubfigureaftersnumb}{}
\newcommand*{\cftsubfigureleader}{}
\newcommand*{\cftsubfiguredotsep}{1}
\newcommand*{\cftsubfigurepagefont}{}
\newcommand*{\cftsubfigureafterpnum}{}
\newcommand*{\cftsubfigureformatpnum}[1]{}
\newcommand*{\cftsubfigureformatpnumhook}[1]{}

% \newlength{\cftbeforetableskip}
% \newlength{\cfttableindent}
% \newlength{\cfttablenumwidth}
\renewcommand*{\cfttablefont}{}
\renewcommand*{\cfttablename}{}
\renewcommand*{\cfttablepresnum}{}
\renewcommand*{\cfttableaftersnum}{}
\renewcommand*{\cfttableaftersnumb}{}
\renewcommand*{\cfttableleader}{}
\renewcommand*{\cfttabledotsep}{1}
\renewcommand*{\cfttablepagefont}{}
\renewcommand*{\cfttableafterpnum}{}
\renewcommand*{\cfttableformatpnum}[1]{}
\renewcommand*{\cfttableformatpnumhook}[1]{}

% \newlength{\cftbeforesubtableskip}
% \newlength{\cftsubtableindent}
% \newlength{\cftsubtablenumwidth}
\newcommand*{\cftsubtablefont}{}
\newcommand*{\cftsubtablename}{}
\newcommand*{\cftsubtablepresnum}{}
\newcommand*{\cftsubtableaftersnum}{}
\newcommand*{\cftsubtableaftersnumb}{}
\newcommand*{\cftsubtableleader}{}
\newcommand*{\cftsubtabledotsep}{1}
\newcommand*{\cftsubtablepagefont}{}
\newcommand*{\cftsubtableafterpnum}{}
\newcommand*{\cftsubtableformatpnum}[1]{}
\newcommand*{\cftsubtableformatpnumhook}[1]{}

\renewcommand*{\booknumberline}[1]{}
\renewcommand*{\partnumberline}[1]{}
\renewcommand*{\chapternumberline}[1]{}
\renewcommand*{\numberlinehook}[1]{}
% \renewcommand*{\cftwhatismyname}{}%
```

```
\renewcommand*{\booknumberlinehook}[1]{}
\renewcommand*{\partnumberlinehook}[1]{}
\renewcommand*{\chapternumberlinehook}[1]{}
\renewcommand{\numberlinebox}[2]{}
\renewcommand{\booknumberlinebox}[2]{}
\renewcommand{\partnumberlinebox}[2]{}
\renewcommand{\chapternumberlinebox}[2]{}
%
% \newlength{\cftparfillskip}
\renewcommand*{\cftpagenumbersoff}[1]{}
\renewcommand*{\cftpagenumberson}[1]{}
\renewcommand*{\cftlocalchange}[3]{}
\renewcommand*{\cftaddtitleline}[4]{}
\renewcommand*{\cftaddnumtitleline}[4]{}
\renewcommand{\cftinsertcode}[2]{}
\renewcommand{\cftinserthook}[2]{}
\renewcommand{\settocpreprocessor}[2]{}
\DeclareRobustCommand{\cftpagenumbersoff}[1]{}
\DeclareRobustCommand{\cftpagenumberson}[1]{}
```

§ 301.11 Floats and captions

`\newfloat` `[`⟨*1: within*⟩`]` `{`⟨*2: type*⟩`}` `{`⟨*3: ext*⟩`}` `{`⟨*4: capname*⟩`}`

```
\RenewDocumentCommand{\newfloat}{o m m m}{%
\IfValueTF{#1}%
{\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}%
{\DeclareFloatingEnvironment[fileext=#3,name={#4}]{#2}}%
```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```
\cslet{listof#2s}\relax%
\cslet{listof#2es}\relax%
}
```

`\newlistof` `[`⟨*within*⟩`]` `{`⟨*type*⟩`}` `{`⟨*ext*⟩`}` `{`⟨*listofname*⟩`}`

Emulated through the `\newfloat` mechanism. Note that memoir uses a different syntax than tocloft for the name.

```
\RenewDocumentCommand{\newlistof}{o m m m}
{%
\IfValueTF{#1}
{\newlistentry[#1]{#2}{#3}{0}}
{\newlistentry{#2}{#3}{0}}
\@namedef{ext@#2}{#3}%
\@ifundefined{c@#3depth}{\newcounter{#3depth}}{}%
\setcounter{#3depth}{1}%
\@namedef{#3mark}{}%
\@namedef{#2}{\listof{#2}{#4}}
```

```
\@namedef{@cftmake#3title}{}
\@ifundefined{cftbefore#3titleskip}{
    \expandafter\newlength\csname cftbefore#3titleskip\endcsname
    \expandafter\newlength\csname cftafter#3titleskip\endcsname
}{}
\@namedef{cft#3titlefont}{}
\@namedef{cftafter#3title}{}
\@namedef{cft#3prehook}{}
\@namedef{cft#3posthook}{}
}

\renewcommand{\setfloatadjustment}[2]{}
```

Borrowed from the lwarp version of keyfloat:

```
\NewDocumentEnvironment{KFLTmemoir@marginfloat}{O{-1.2ex} m}
{% start
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}

\DeclareDocumentEnvironment{marginfigure}{o}
  {\begin{KFLTmemoir@marginfloat}{figure}}
  {\end{KFLTmemoir@marginfloat}}

\DeclareDocumentEnvironment{margintable}{o}
  {\begin{KFLTmemoir@marginfloat}{table}}
  {\end{KFLTmemoir@marginfloat}}

\renewcommand{\setmarginfloatcaptionadjustment}[2]{}
\renewcommand{\setmpjustification}[2]{}
\renewcommand*{\mpjustification}{}
\renewcommand*{\setfloatlocations}[2]{}
\DeclareDocumentCommand{\suppressfloats}{o}{}
\renewcommand*{\FloatBlock}{}
\renewcommand*{\FloatBlockAllowAbove}{}
\renewcommand*{\FloatBlockAllowBelow}{}
\renewcommand*{\setFloatBlockFor}{}
\renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{#1}}
\renewcommand*{\captionnamefont}[1]{}
\renewcommand*{\captiontitlefont}[1]{}
\renewcommand*{\captionstyle}[2][]{}
\renewcommand*{\centerlastline}{}
\renewcommand*{\hangcaption}{}
\renewcommand*{\indentcaption}[1]{}
\renewcommand*{\normalcaption}{}
```

```
\renewcommand*{\changecaptionwidth}{}
\DeclareDocumentCommand{\captionwidth}{m}{}
\renewcommand*{\normalcaptionwidth}{}
\renewcommand{\precaption}[1]{}
\renewcommand{\captiontitlefinal}[1]{}
\renewcommand{\postcaption}[1]{}

\renewcommand{\contcaption}[1]{%
% \ContinuedFloat%
% \caption{#1}%
\begin{LWR@figcaption}% later becomes \caption*
\csuse{\@captype name} \thechapter.\the\value{\@captype}\CaptionSeparator #1
\end{LWR@figcaption}
}
```

The extra \\ here forces a <br> in HTML when \legend is used in a \marginpar.

```
\renewcommand{\legend}[1]{\begin{center}#1\\\end{center}}

\renewcommand{\namedlegend}[2][]{
\begin{center}
\csuse{fleg\@captype}\CaptionSeparator#2\\
\end{center}
\csuse{flegtoc\@captype}{#1}
}

\renewcommand{\flegtable}{\tablename}
\renewcommand{\flegfigure}{\figurename}
\renewcommand{\flegtoctable}{}
\renewcommand{\flegtocfigure}{}

\renewcommand{\newfixedcaption}[3][\caption]{%
  \renewcommand{#2}{\def\@captype{#3}#1}}
\renewcommand{\renewfixedcaption}[3][\caption]{%
  \renewcommand{#2}{\def\@captype{#3}#1}}
\renewcommand{\providefixedcaption}[3][\caption]{%
  \providecommand{#2}{\def\@captype{#3}#1}}

\renewcommand{\bitwonumcaption}[6][]{%
\ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
\addtocounter{\@captype}{-1}%
\begingroup%
\csdef{\@captype name}{#4}%
\ifblank{#5}{\caption{#6}}{\caption[#5]{#6}}%
\endgroup%
\ifblank{#1}{}{\label{#1}}%
}

\LetLtxMacro\bionenumcaption\bitwonumcaption% todo
```

```
787
788 \renewcommand{\bicaption}[5][]{%
789 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
790 \begin{LWR@figcaption}% later becomes \caption*
791 #4 \thechapter.\the\value{\@captype}\CaptionSeparator #5
792 \end{LWR@figcaption}
793 \ifblank{#1}{}{\label{#1}}%
794 }
795
796 \renewcommand{\bicontcaption}[3]{%
797 \contcaption{#1}%
798 \begingroup%
799 \csdef{\@captype name}{#2}%
800 \contcaption{#3}%
801 \endgroup%
802 }
803
804 \renewcommand{\midbicaption}[1]{}
805
806 \renewcommand{\subcaption}[2][]{%
807 \ifblank{#1}{\subfloat[#2]{}}{\subfloat[#1][#2]{}}%
808 }
809
810 \RenewDocumentCommand{\subtop}{O{} O{} m}{%
811 \subfloat[#1][#2]{#3}%
812 }
813
814 \RenewDocumentCommand{\subbottom}{O{} O{} m}{%
815 \subfloat[#1][#2]{#3}%
816 }
817
818 \renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}
819
820 \renewcommand{\contsubtop}{%
821 \ContinuedFloat\addtocounter{\@captype}{1}%
822 \subtop}
823
824 \renewcommand{\contsubbottom}{%
825 \ContinuedFloat\addtocounter{\@captype}{1}%
826 \subbottom}
827
828 \renewcommand{\subconcluded}{}
829
830 \LetLtxMacro\subcaptionref\subref
831
832 \renewcommand*{\tightsubcaptions}{}
833 \renewcommand*{\loosesubcaptions}{}
834
835 \renewcommand*{\subcaptionsize}[1]{}
836 \renewcommand*{\subcaptionlabelfont}[1]{}
```

```
\renewcommand*{\subcaptionfont}[1]{}
\renewcommand*{\subcaptionstyle}[1]{}

\renewcommand*{\hangsubcaption}{}
\renewcommand*{\shortsubcaption}{}
\renewcommand*{\normalsubcaption}{}

\RenewDocumentEnvironment{sidecaption}{o m o}
{}
{
\IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
\IfValueT{#3}{\label{#3}}%
}

% \newlength{\sidecapwidth}
% \newlength{\sidecapsep}
\renewcommand*{\setsidecaps}[2]{}
\renewcommand*{\sidecapmargin}[1]{}
% \newif\ifscapmargleft
\scapmargleftfalse
\renewcommand*{\setsidecappos}[1]{}

\RenewDocumentEnvironment{sidecontcaption}{m o}
{}
{%
\ContinuedFloat%
\caption{#1}%
```

Without `\@captype`, the section is referred to instead.

```
\IfValueT{#2}{\label[\@captype]{#2}}%
}
```

`\sidenamedlegend` does not appear to use the TOC argument.

```
\renewenvironment{sidenamedlegend}[2][]{
\begin{center}
\csuse{\@captype name}\CaptionSeparator#2
\end{center}
}
{}

\renewenvironment{sidelegend}[1]
{\begin{center}
#1

}
{\end{center}}

```

```
\renewcommand*{\sidecapstyle}{}
\renewcommand*{\overridescapmargin}[1]{}
% \newlength{\sidecapraise}
\renewcommand*{\sidecapfloatwidth}{\linewidth}

\LetLtxMacro\ctabular\tabular
\LetLtxMacro\endctabular\endtabular

\renewcommand{\autorows}[5][]{%
#5
}

\renewcommand{\autocols}[5][]{%
#5
}
```

§ 301.12 Page notes

```
\renewcommand*{\feetabovefloat}{}
\renewcommand*{\feetbelowfloat}{}
\renewcommand*{\feetatbottom}{}

\renewcommand*{\verbfootnote}[2][]{
\PackageError{lwarp,memoir}
{Verbatim footnotes are not yet supported by lwarp.}
{This may be improved some day.}
}

\renewcommand*{\plainfootnotes}{}
\renewcommand*{\twocolumnfootnotes}{}
\renewcommand*{\threecolumnfootnotes}{}
\renewcommand*{\paragraphfootnotes}{}
\renewcommand*{\footfudgefiddle}{}

\renewcommand*{\newfootnoteseries}[1]{
\PackageError{lwarp,memoir}
{Memoir footnote series are not yet supported by lwarp.}
{This may be improved some day.}
}

\renewcommand*{\plainfootstyle}[1]{}
\renewcommand*{\twocolumnfootstyle}[1]{}
\renewcommand*{\threecolumnfootstyle}[1]{}
\renewcommand*{\paragraphfootstyle}[1]{}

\renewcommand*{\footfootmark}{}
\renewcommand*{\footmarkstyle}[1]{}

% \newlength{\footmarkwidth}
```

```
% \newlength{\footmarksep}
% \newlength{\footparindent}

\renewcommand*{\foottextfont}{}

\renewcommand*{\marginparmargin}[1]{}
\renewcommand*{\sideparmargin}[1]{}

\LetLtxMacro\sidepar\marginpar
\renewcommand*{\sideparfont}{}
\renewcommand*{\sideparform}{}
\LWR@providelength{\sideparvshift}

\renewcommand*{\parnopar}{}

\renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
\renewcommand*{\sidebarmargin}[1]{}
\renewcommand*{\sidebarfont}{}
\renewcommand*{\sidebarform}{}
% \newlength{\sidebarhsep}
% \newlength{\sidebarvsep}
% \newlength{\sidebarwidth}
% \newlength{\sidebartopsep}
\renewcommand{\setsidebarheight}[1]{}
\renewcommand*{\setsidebars}[6]{}
\renewcommand*{\footnotesatfoot}{}
\renewcommand*{\footnotesinmargin}{}

\LetLtxMacro\sidefootnote\footnote
\LetLtxMacro\sidefootnotemark\footnotemark
\LetLtxMacro\sidefootnotetext\footnotetext

\renewcommand*{\sidefootmargin}[1]{}
% \newlength{\sidefoothsep}
% \newlength{\sidefootvsep}
% \newlength{\sidefootwidth}
% \newlength{\sidefootadjust}
% \newlength{\sidefootheight}
\renewcommand*{\setsidefootheight}[1]{}
% \renewcommand*{\sidefootfont}{}% in docs but not in the package
\renewcommand*{\setsidefeet}[6]{}
\renewcommand*{\sidefootmarkstyle}[1]{}
\renewcommand*{\sidefoottextfont}{}
\renewcommand*{\sidefootform}{}

\renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
\renewcommand*{\notepageref}{}
\renewcommand*{\prenotetext}{}
\renewcommand*{\postnotetext}{}
\renewcommand*{\idtextinnotes}[1]{}
```

```
\renewcommand*{\printpageinnotes}[1]{}
\renewcommand*{\printpageinnoteshyperref}[1]{}
\renewcommand*{\foottopagenote}{}
\renewcommand*{\pagetofootnote}{}
```

§ 301.13 Decorative text

```
\renewcommand*{\epigraphposition}[1]{}
\renewcommand*{\epigraphtextposition}[1]{}
\renewcommand*{\epigraphsourceposition}[1]{}
\renewcommand*{\epigraphfontsize}[1]{}
\renewcommand*{\epigraphforheader}[2][]{}
\renewcommand*{\epigraphpicture}{}
```

§ 301.14 Poetry

```
\renewcommand*{\vinphantom}{}
\renewcommand*{\vleftofline}[1]{#1}
% \let\linenumberfrequency\poemlines
% \renewcommand*{\linenumberfont}[1]{}

\DeclareDocumentCommand{\PoemTitle}{s o o m}{%
\IfValueTF{#2}%
{\poemtitle[#2]{#4}}%
{\poemtitle{#4}}%
}

\renewcommand*{\NumberPoemTitle}{}
\renewcommand*{\PlainPoemTitle}{}
\renewcommand*{\poemtitlepstyle}{}
\renewcommand*{\poemtitlestarmark}[1]{}
\renewcommand*{\poemtitlestarpstyle}{}
\renewcommand*{\PoemTitleheadstart}{}
\renewcommand*{\printPoemTitlenonum}{}
\renewcommand*{\printPoemTitlenum}{}
\renewcommand*{\afterPoemTitlenum}{}
\renewcommand*{\printPoemTitletitle}[1]{}
\renewcommand*{\afterPoemTitle}{}
\newlength{\midpoemtitleskip}
\renewcommand*{\PoemTitlenumfont}{}
\renewcommand*{\PoemTitlefont}{}
```

§ 301.15 Boxes, verbatims and files

```
\renewenvironment{qframe}{\framed}{\endframed}
\renewenvironment{qshade}{\shaded}{\endshaded}
```

Use the comment package:

```
\renewcommand*{\commentsoff}[1]{\includecomment{#1}}
\renewcommand*{\commentson}[1]{\excludecomment{#1}}
\LetLtxMacro\renewcomment\commentson

\renewcommand*{\setverbatimfont}[1]{}
\renewcommand*{\tabson}[1]{}
\renewcommand*{\tabsoff}{}
\renewcommand*{\wrappingon}{}
\renewcommand*{\wrappingoff}{}
\renewcommand*{\verbatimindent}{}
\renewcommand*{\verbatimbreakchar}[1]{}

\DefineVerbatimEnvironment{fboxverbatim}{Verbatim}{frame=single}
```

`boxedverbatim` is already defined by moreverb. `boxedverbatim*` does not appear to work at all, even in a minimal print memoir document.

```
\renewcommand*{\bvbox}{}
\renewcommand*{\bvtopandtail}{}
\renewcommand*{\bvsides}{}
\renewcommand*{\nobvbox}{}
% \newlength\bvboxsep
\renewcommand*{\bvtoprulehook}{}
\renewcommand*{\bvtopmidhook}{}
\renewcommand*{\bvendrulehook}{}
\renewcommand*{\bvleftsidehook}{}
\renewcommand*{\bvrightsidehook}{}
\renewcommand*{\bvperpagetrue}{}
\renewcommand*{\bvperpagefalse}{}
\renewcommand{\bvtopofpage}[1]{}
\renewcommand{\bvendofpage}[1]{}
\renewcommand*{\linenumberfrequency}[1]{}
\renewcommand*{\resetbvlinenumber}{}
\renewcommand*{\setbvlinenums}[2]{}
\renewcommand*{\linenumberfont}[1]{}
\renewcommand*{\bvnumbersinside}{}
\renewcommand*{\bvnumbersoutside}{}
```

§ 301.16 Cross referencing

```
\renewcommand*{\fref}[1]{\cref{#1}}
\renewcommand*{\tref}[1]{\cref{#1}}
\renewcommand*{\pref}[1]{\cpageref{#1}}
\renewcommand*{\Aref}[1]{\cref{#1}}
\renewcommand*{\Bref}[1]{\cref{#1}}
\renewcommand*{\Pref}[1]{\cref{#1}}
\renewcommand*{\Sref}[1]{\cref{#1}}
\renewcommand*{\figurerefname}{Figure}
\renewcommand*{\tablerefname}{Table}
```

```
\renewcommand*{\pagerefname}{page}
\renewcommand*{\bookrefname}{Book~}
\renewcommand*{\partrefname}{Part~}
\renewcommand*{\chapterrefname}{Chapter~}
\renewcommand*{\sectionrefname}{\S}
\renewcommand*{\appendixrefname}{Appendix~}
\LetLtxMacro\titleref\nameref
\renewcommand*{\headnameref}{}
\renewcommand*{\tocnameref}{}

\providecounter{LWR@currenttitle}

\renewcommand*{\currenttitle}{%
    \addtocounter{LWR@currenttitle}{1}%
    \label{currenttitle\arabic{LWR@currenttitle}}%
    \nameref{currenttitle\arabic{LWR@currenttitle}}%
}

\renewcommand*{\theTitleReference}[2]{}
\renewcommand*{\namerefon}{}
\renewcommand*{\namerefoff}{}
```

§ 301.17 Back matter

```
\DeclareDocumentCommand{\newblock}{}{}
%
\renewcommand*{\showindexmarks}{}
\renewcommand*{\hideindexmarks}{}

\renewcommand*{\xindyindex}{}
```

§ 301.18 Miscellaneous

```
\renewcommand*{\changemarks}{}
\renewcommand*{\nochangemarks}{}
\renewcommand*{\added}[1]{}
\renewcommand*{\deleted}[1]{}
\renewcommand*{\changed}[1]{}

\renewcommand*{\showtrimsoff}{}
\renewcommand*{\showtrimson}{}
\renewcommand*{\trimXmarks}{}
\renewcommand*{\trimLmarks}{}
\renewcommand*{\trimFrame}{}
\renewcommand*{\trimNone}{}
\renewcommand*\trimmarkscolor{}
\renewcommand*{\trimmarks}{}
\renewcommand*{\tmarktl}{}
\renewcommand*{\tmarktr}{}
```

```
\renewcommand*{\tmarkbr}{}
\renewcommand*{\tmarkbl}{}
\renewcommand*{\tmarktm}{}
\renewcommand*{\tmarkmr}{}
\renewcommand*{\tmarkbm}{}
\renewcommand*{\tmarkml}{}
\renewcommand*{\trimmark}{}
\renewcommand*{\quarkmarks}{}
\renewcommand*{\registrationColour}[1]{}

\renewcommand*{\leavespergathering}[1]{}

\renewcommand*{\noprelistbreak}{}

\renewcommand*{\cleartorecto}{}
\renewcommand*{\cleartoverso}{}

\renewenvironment{vplace}[1][]{}{}
```

§ 301.19 Final patchwork

```
\newlistof{tableofcontents}{toc}{\contentsname}
\newlistof{listoffigures}{lof}{\listfigurename}
\newlistof{listoftables}{lot}{\listtablename}
```

Change History and Index

§ 301 Change History

v0.10
General: 2016/03/08 Initial version . 1
v0.11
General: 2016/03/11 1
Added section: Operating-System portability. 137
Added section: Selecting the operating system. 86
Test Suite: limages and index in README.txt 1
Test Suite: MS-Windows in README.txt 1
v0.12
\LWR@newhtmlfile: Bugfix: TOC with numbered files. 259
General: 2016/03/14 1
Global: Uses \p@(type) in float captions. 1
Test Suite: Sub-figures 1
v0.13
\CaptionSeparator: Fix for newer babel package. 367
\LWR@LwarpStart: \up and \fup . . 276
General: 2016/03/24 1
Removed package: subfig 1
Test Suite: Ordinals, Subcaption . . 1
tikzpicture: Fix dollar-redefined bug for newer package. 639
v0.14
\LWR@htmlsectionfilename: Fix: Links to home page. 226
General: 2016/03/31 1
floatrow: Added. 501
Docs: Commands for a successful HTML conversion. 91
Docs: Commands into a warpprint environment. 88
Docs: Newclude limitations. . . . 115
Docs: Table: Cross-referencing data structures. 353
Docs: Table: Float data structures. 364
Docs: Trademarks section. 450
Docs: Troubleshooting cross-references. 128
Test Suite: Assigned cleveref name for Test Float. 1
Test Suite: Floatrow 1
v0.15
General: 2016/04/06 1
Added 506
Ampersand (&): Fixed handling when passed as an argument. . 311
Docs: Added warning icons for items needing special attention. 134
Docs: Clarify print/HTML output. 87
Docs: Moved the supported functions table to the introduction. 48
Files: lwarp_formal.css added. . . . 1
Fix: steps counter 506
Fixed & handling. 504
Test Suite: test_suite_formal.css file added. 1
v0.16
General: 2016/04/11 1
\titlingpage: Improved print-output spacing. 284
Added XeLaTeX, LuaLaTeX support. 148
Docs: Font and UTF-8 support. . 76
Docs: Moved location of \usepackage{lwarp}. 78
Docs: Text not converting. 128
Fix: amsmath options clash 152
Fix: newtxmath compatibility. . . 152
Lwarp no longer selects fonts. 76, 148
Removed package: suffix 1
Test Suite: Improved titlingpage. 284
Test Suite: Lwarp no longer selects fonts. 1

Test Suite: Supports XeLaTeX, LuaLaTeX. 1
xfrac: Adjusted for the use of any font: 684
v0.17
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. 226
General: 2016/04/14 1
mdframed: Added. 554
Test Suite: Fix: Print-version front-matter page numbers. . . . 1
Test Suite: Mdframed 1
v0.18
\LWR@hspace: \hspace supported. . 440
\LWR@includegraphicsb: Add: svgz file extension. 523
em, ex, %, px dimensions preserved. 523
Fix: \linewidth, \textwidth, \textheight inside a minipage. 523
Improved HTML output linebreaks. 523
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic. . . . 371
\LWR@newhtmlfile: sideTOC after title, improving responsive design. 258
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic. . . . 277
\LWR@subhyperref: Improved HTML output linebreaks. 361
\LWR@subhyperrefclass: Improved HTML output linebreaks. 362
\LWR@subinlineimage: Surpress extra space. 363
General: 2016/05/19 1
File: lwarp.css: Improved TOC outline display. 1
Files: lwarp.css and lwarp_formal.css: Improved responsive design. 1
Microtype disabled during HTML generation 149
PDF Unicode input characters. . 135
Test Suite: Verse package 1
lateximage: pdfcrop: --hires added. 407
Reorganize \HomeHTMLFilename logic. 407
Surpress extra space. 407
verse: Supports verse, memoir packages. 670
minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage. 422
v0.19
\HTMLFilename: Docs: Escape filename underscores. 225
\HomeHTMLFilename: Docs: Escape filename underscores. 225
\LWR@LwarpStart: Enabled \\ equal to \newline. 274
\LWR@doubledollar: MathJax support. 387
\LWR@filestart: lwarp_mathjax.txt loaded. 272
\LWR@hspace: Fix: \hspace length computations. 440
\LWR@minipagestartpars: Surpresses paragraph tags between minipages. 439
\LWR@singledollar: MathJax support. 388
\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. 403
\minipagefullwidth: Added: No width tag for the next minipage in HTML. 422
\rule: Added 444
\warpHTMLonly: Added. 141
\warpprintonly: Replaces \rowprintedonly. 141
\xfracHTMLfontsize: Added. . . . 683
General: 2016/06/08 1
Added optional args. 571
Avoids MathJax. 386
cleveref: Loaded \AtEndPreamble. 419
CSS for table note item. 638
Docs: Math options. 78
Docs: Table: Cross-referencing data structures, updated. . . . 353
File: lwarp_mathjax.txt added. . 1
File: lwarp.css: tnoteitemheader added 1
Introduction: MathJax support mentioned. 46
MathJax support added. 389, 394, 395
Options: mathsvg and mathjax . 139

Supports colored \rule. 678
titleps: null \pagestyle and \thispagestyle for HTML. . . 640
tikzpicture: Adapts to tikz version. 639
equation: MathJax support. 392
v0.20
\BlockClassSingle: Renamed from "LWR@htmldivclassline". . . . 238
\HTMLDescription: Added \NewHTMLdescription. (Renamed in v0.30.) 248
\HTMLFilename: No longer escape underscores. 225
\HomeHTMLFilename: No longer escape underscores. 225
\InlineClass: Renamed from "inlineclass". 238
\LWR@LwarpStart: Fix: math cross references. 276
\LWR@closeparagraph: \unskip extra spaces. 242
No break tags in the start/end of a tabular. 242
\LWR@endofline: Fix: \\ 438
\LWR@filestart: Adds meta description. 272
\LWR@hspace: Add: Supports HTML thin breakable space. 440
\LWR@htmldivclass: Added optional style. 236
\LWR@htmlelementclass: Added optional style. 236
\LWR@htmlsectionfilename: HTMLFilename: removed additional trailing '-', and may be empty. 226
Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix. . . . 226
\LWR@includegraphicsb: Fix: \linewidth in a floatrow. . . . 523
Fix: Expands filename. 523
\LWR@longtabledatacaptiontag: Fix: Pars in captions. 338
\LWR@section: Combined higher-level sections together into files. 264
\LWR@setOSWindows: Auto-detects operating system. 137
\LWR@subhtmlelementclass: Factored code. 236
\SetHTMLFileNumber: Add: Control file numbers. 225
\cpagerefFor: User-redefinable word for page references. . . . 419
\dotfill: Inserts an ellipsis. 437
\hfill: Inserts a \qquad. 436
\hrulefill: Inserts a short rule. . . 437
\hyperindexref: Print mode provided in case hyperref not used. 380
\pageref: Added. 360
\tracinglwarp: Added. 158
General: 2017/02/09 1
afterpage: Added. 455
alltt: Added. 457
bookmark: Added. 467
caption and subcaption supported. 1
cleveref and referencing patches: Applied \AfterEndPreamble. . 419
draftwatermark: Added. 477
eso-pic: Added. 482
everypage: Added. 483
extramarks: Added. 484
fancyhdr: Added. 488
hyperref: Additional user macros. 533
keyfloat: Added. 539
letterspace: User-interface emulated. 541
listings: Added. 543
ltcaption: Added. 549
lwarp-newproject: Added. 166
microtype: User-interface emulated. 565
needspace: Added. 574
nowidow: Added. 576
placeins: Added. 592
ragged2e: Added. 595
setspace: Improved support. . . . 608
textpos: Added. 633
titleps: Added. 640
titlesec: Added. 643
titletoc: Added. 645
titling: Improved compatibility. . 647
tocloft: Added. 655
wallpaper: Added. 671
wrapfig: Added. 672
Added @, <, > columns. 305

Added single-expansion data arrays. 223
Code factored into independent lwarp_html files. 450
Docs: Examples for generating HTML file names. 84
Docs: Improved index. 1
Enhanced titling support. 282
File: lwarp.css: Minor fixes for validation. 1
File: lwarpmk used to compile print, HTML, indexes, and lateximages. 1
Fix: \linewidth in a floatrow. . . 504
Improved float caption type handling. 499
Moved sidebar and example code to test suite. 1
Page geometry set to 6in wide with large margins. 150
Parallel versions of aux files for print/HTML. 1
Removed reliance on make, grep, gawk. 1
Tabular: \unskip extra spaces. . . 305
Test Suite: HTML meta descriptions. 1
verbatim: Added. 293
BlockClass: Added optional style. . 238
Renamed from "blockclass". . . . 238
LWR@nestspan: Fix: Minipages inside a span. 233
v0.21
\LWR@LwarpStart: Changed lateximages to a .txt file. . . 274
\LWR@filestart: Skip title if not given. 272
\LWR@newhtmlfile: Skip title if not given. 258
\marginpar: Fixed source listing. . . 254
\marginparBlock: Fixed source listing. 254
General: 2017/02/23 1
fontenc: Added. 509
fontspec: Added. 509
inputenc: Added. 539
newclude: Added. 575
newunicodechar: Added. 575
lwarpmk: Fix: lwarpmk again for Windows. 208
lwarpmk: Fix: lwarpmk limages for Windows. 208
lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. 208
Add: Errors for misplaced packages. 142
Docs: Added internet class. . . . 52
Docs: Added TeX2page, GladTeX. 52
Docs: Installing on Windows. . . . 58
File lwarp_tutorial.txt added. 63
v0.22
\LWR@parseDcolumn: Added tabular D column. 318
\LWR@parsebangcolumn: Added tabular ! column. 315
\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns. 319
\LWR@printmccoldata: Added tabular D, !, and X columns. . . 334
General: 2017/03/02 1
abstract: Added. 452
changepage: Added. 472
dcolumn: Added. 476
ftnright: Added. 515
geometry: Nullified commands. . 516
indentfirst: Added. 539
layout: Added. 541
lscape: Added. 549
mcaption: Added. 553
nameref: Added. 573
nextpage: Added. 575
parskip: Added. 591
showkeys: Added. 610
sidecap: Added. 611
tabularx: Added. 627
varioref: Supported. 98
verse: Added. 669
v0.23
\LWR@parsetablecols: Fix for vert bar column type. 319
\LWR@printmccoldata: Fix for vert bar column type. 334
General: 2017/03/02 1
v0.24
\LWR@hspace: Add: \hspace \fill converts to 2em 440
\LWR@htmlfileref: Fix: Index links while \tracinglwarp. 356

\hypertocfloat: List of floats responds to lofdepth, lotdepth. 378
General: 2017/03/15 1
floatrow: Support for subfig. . . . 501
subfig: Added. 618
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary. 639
Docs: Filename underscore. . 81, 92
No longer preloads subcaption; conflicted with subfig. 152
picture: Fix for inline images. . . . 420
tikzpicture: Fix for inline images. 639
v0.25
\LWR@loadnever: Added the ability to prevent conflicting packages. 143
\addcontentsline: Handles theorems. 370
General: 2016/03/22 1
amsthm: Added. 457
ccaption: Prevented. 471
ellipsis: Added. 478
emptypage: Added. 479
framed: Added. 512
lips: Added. 543
mdframed: Help avoid hyphenation. 556
ntheorem: Added. 577
showidx: Added. 610
theorem: Added. 633
Basic LaTeX theorems: improved css. 294
Docs: Adds credits for patched code. 1
Docs: Testing lwarp. 126
Fix: Allows XeLaTeX and LuaLaTeX to preload graphics and graphicx. 144
v0.26
General: 2017/03/31 1
cutwin: Added. 475
endnotes: Added. 480
floatflt: Added. 500
footmisc: Added. 509
footnotehyper: Added. 512
footnote: Added. 511
marginfix: Added. 552
marginnote: Added. 553
mparhack: Added. 569
pagenote: Supported as-is. . . . 590
sidenotes: Added. 611
lwarp.css: Improved responsive marginpar and marginblock. . 168
Docs: Improved MiKTeX install instructions. 57, 59
Dollar span avoided in a lateximage. 386
Footnotes now are LaTeX boxes instead of pagenotes. 249
lateximage: Labels track page numbers of lateximages. 407
Print mode now uses a minipage of \linewidth. 407
picture: Fix for \makebox in picture. 420
v0.27
\LWR@footnotetext: Fix for table footnote par tags. 250
General: 2017/04/04 1
lettrine: Added. 542
microtype: Fix with XeLaTeX, LuaLaTeX. 565
soul: Added. 614
ulem: Added. 667
Docs: Installing utilities for MacOS. 60
Docs: Limitations of saveboxes. . 93
Page geometry modified to reduce line overflow. 150
v0.28
\@wrindex: Improved indexing. . . . 380
\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.) 248
\LWR@LwarpEnd: If FormatEPUB or FormatWordProcessor, no bottom nav. 278
\LWR@LwarpStart: FormatWordProcessor forces single-file output. 274
\LWR@filestart: Adds HTML meta author. 272
\LWR@forcenewpage: Forces new PDF page before major environments. 229
\LWR@htmlcomment: Breaks ligatures in HTML comments. 235
\LWR@includegraphicsb: Adapts to graphics syntax. 523

\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav. 258
\LWR@parsetablecols: Added L, C, R, J column types. 319
\LWR@startref: Removed space. . . 358
\chapter: If EPUB, prints footnotes before each section. 270
\hyperindexref: Improved indexing. 380
\textup: Fixed span class. 432
General: 2017/04/14 1
glossaries: Added. 517
graphics: Added. 518
tabularx: Fix for optional pos. . . 627
tabulary: Added. 627
lwarpmk: Add: printglossary and htmlglossary commands. 208
Added boolean FormatEPUB. . . . 159
Added boolean FormatWP. 159
Added boolean HTMLDebugComments. 159
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42. 160
Docs: Modfying lwarpmk and index processing. 126
File lwarp_mathjax.txt: Updated CDN repository. 206
Forced oneside to maintain large right margin. 150
v0.29
\LWR@includegraphicsb: Fix: Error when no optional arguments. . 523
General: 2017/04/15 1
lwarpmk: Add: language option for config files. 208
Add: lwarpmklang option for lwarp. 139
Docs: Using a glossary 74
File *.lwarpmkconf: Add: language option for config files. 167
File lwarpmk.conf: Add: language option for config files. 167
v0.30
\CSSFilename: Renamed from \NewCSS. 247
\HTMLAuthor: Renamed from \HTMLauthor. 248
\HTMLDescription: Renamed from \NewHTMLdescription. 248
\HTMLFirstPageTop: Renamed from \SetFirstPageTop. 246
\HTMLLanguage: Renamed from \MetaLanguage. 272
\HTMLPageBottom: Renamed from \SetPageBottom. 247
\HTMLPageTop: Renamed from \SetPageTop. 247
General: 2017/04/29 1
lwarp-newproject removed, and combined with lwarp. 166
lwarpmk: Add: xdyfile configuration option. 208
lwarpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex. 208
lwarpmk: Fix: xindy now used for print index generation with latexmk. 208
lwarpmk: language now used for both index and glossary generation. 208
File: lwarp_html.xdy renamed to lwarp.xdy. 205
Fix: *.css files only written in print mode. 168
Fix: lwarp.xdy only written in print mode. 205
Fix: lwarp_mathjax.txt: Only written in print mode. 206
Option OSWindows replaces macro \warpOSwindows. 140
Option latexmk replaces macro \UseLatexmk. 140
Option lwarpmklang changed to IndexLanguage. 139
Option xdyFilename added. . . 139
Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. 140
v0.31
General: 2017/05/15 1
keyfloat: Improved compatibility. 539
v0.32
\RequirePackage: Fix: Ignores blanks in package list. 154
General: 2016/06/09 1

glossaries: Prevent error with \glo@name not defined. 381
lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes. 208
v0.33
\HTMLAuthor: Fix: Provides empty default author if none given. . . 248
\LWR@loadbefore: Fix: No \PackageError if already loaded. 143
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . . 315
\LWR@tabledatasinglecolumntag: Fix: Macros in tabular could cause extra data cell. 325
\LWR@vspace: Add: \vspace nullified. 441
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. 311
General: 2017/07/10 1
amsmath: Removed fleqn option. 152
fancyhdr: Fix: Optional args for \lhead, etc. 488
Add: Tabular at and bang columns now have their own HTML columns. 305
cleveref: Fix: Loaded \AtEndPreamble. 419
Fix: Incorrectly-inline math environments. 395
New handling of & to localize catcode changes. 305
v0.34
\@fnsymbol: Text symbols instead of math. 285
\InlineClass: Moved optional argument in front of mandatory. 238
\LWR@htmldivclass: Moved optional argument in front of mandatory. 236
\LWR@htmlelementclass: Moved optional argument in front of mandatory. 236
\LWR@htmlelementclassline: Moved optional argument in front of mandatory. 237
\LWR@htmlspanclass: Moved optional argument in front of mandatory. 234
\LWR@nullfonts: Improved font control. 433
\LWR@restoreorigformatting: booktabs: Works inside lateximage. 382
Improved font control. 382
\LWR@subhtmlelementclass: Moved optional argument in front of mandatory. 236
\LWR@tabledatacolumntag: booktabs: Works inside lateximage. 345
\fboxBlock: Added. 428
\makebox: Fix: Handles paren arg. . 426
General: 2017/08/08 1
babel-french: Adds fixed-width HTML spaces to punctuation. . 231
balance: Added. 465
booktabs: Works inside lateximage. 348, 468
boxedminipage2e: Added. 469
boxedminipage: Prevented. 468
crop: Added. 474
enumerate: Added. 480
enumitem: Added, no longer required. 481
everyshi: Added. 483
fancybox: Added. 485
fancyvrb: Added, no longer required. 490
figcaps: Added. 496
filecontents: Required. Patched for morewrites. 151
floatpag: Added. 501
flushend: Added. 507
fullpage: Added. 515
hyperxmp: Added. 537
idxlayout: Added. 537
marginfit: Added. 552
mdframed: Improved mdtheorem patch. 562
moreverb: Added. 567
paralist: Added. 590
pdflscape: Added. 591
pdfsync: Added. 591
prelim2e: Added. 592
rotfloat: Added. 599

savetrees: Added. 600
shadow: Added. 610
syntonly: Added. 626
titleps: No longer required. 640
titleref: Prevented. 643
xmpincl: Added. 686
Added. 681
Docs: Horizontal space limitations. 1
Docs: Misplaced alignment character. 128
File: README.txt: updated. 1
File: lwarp_mathjax.txt: Version change. 206
Fix: Added the eqnarray environments. 395
Improved font control. 431
Lists refactored to remove enumitem requirement. 295
Verbatim refactored to remove fancyvrb requirement. 291
lateximage: Fix: lateximage with minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox. . . . 407
BlockClass: Moved optional argument in front of mandatory. 238
fminipage: Added. 428
LWR@nestspan: Fix: Minipages, BlocksClass, and lists inside a span. 233
LWR@tabular: booktabs: Works inside lateximage. 349
v0.35
General: 2017/08/08 1
Fix: \textbf and related. 431
v0.36
\LWR@HTMLsanitize: Fix for babel-french. 404
\LWR@HTMLsanitizeexpand: Fix for babel-french. 404
\LWR@closeparagraph: Extra HTML source space after paragraphs. 242
\LWR@currenttextcolor: Fix for \rule when xcolor not loaded. 442
\LWR@footnotetext: Extra HTML source space after paragraphs. 250
Force HTML superscripts. 250
\LWR@nullfonts: Fix: Filenames while using MathJax. 433
\LWR@restoreorigformatting: siunitx: Improved super/subscripts in a lateximage. 382
\LWR@section: Improved spacing. . 264
\LWR@stoppars: Extra HTML source space after paragraphs. 245
\fbox: Fix: Uses \fboxrule and \fboxsep. 428
\framebox: Fix: Handles width and horiz position. 427
\makebox: Fix: Handles width and horiz position. 426
General: 2017/08/17 1
babel-french: Adjustements for French variants, load order, footnotes, ellipses. 231
footnote: Extra HTML source space after paragraphs. 511
siunitx: Fix for babel-french. 412
siunitx: Improved symbol support. 613
transparent: Added. 664
upref: Added. 669
xcolor: Added \fcolorboxBlock, \colorboxBlock. 674
xcolor: Fix: Background none in print mode. 674
xcolor: Refactored \LWR@colorstyle. 678
xcolor: Uses \fboxrule and \fboxsep. 674
xcolor: \fcolorbox etc. now work inside lateximage. 674
Docs: Reorganized: Special cases and limitations. 91
Source: Improved formatting. . . . 1
lateximage: Footnotes appear in regular text instead of the lateximage minipage. 407
LWR@tabular: Fix for babel-french. . 349
v0.37
\include: Maintains independent .aux files for HTML. 156
General: 2017/08/19 1
LaTeX accents: Added. 164
babel-french: Adjustment for load order. 231
color: Prevented. 474
comment: Maintains independent cutfiles for print, HTML. 141

siunitx: Improved symbol support. 613
textcomp: Improved support. . . . 629
lwarpmk: Removes additional HTML aux files. 208
File handles reorganized. 156
v0.38
\@seccntformat: Added for appendix. 264
\ForceHTMLPage: Added. 261
\ForceHTMLTOC: Added. 261
\LWR@section: \part* starts a new HTML page, for appendix. 264
Modified spacing, uses \numberline. 264
\numberline: Added trailing \quad. 376
\part: Fix with article class. 269
General: 2017/08/27 1
appendix: Added. 462
arabicfront: Added. 463
caption2: Prevented. 471
chappg: Added. 473
color: Forces xcolor as well. 474
fix2col: Added. 496
fncychap: Added. 508
grffile: Added. 530
metalogo: Added. 565
nonumonpart: Added. 576
nopageno: Added. 576
pagenote: Option page disabled. 590
realscripts: Added. 595
relsize: Added. 597
romanbarpagenumber: Added. . . 598
romanbar: Added. 598
scalefnt: Added. 600
siunitx: Removed from lwarp core. 613
textcomp: Removed from lwarp core. 629
tocbibind: Added. 653
xltxtra: Added. 685
lwarpmk: Added print1 and html1 actions. 208
Added \markboth, \sloppy, etc. 229
Docs: Enhanced *Supported Functions* table. 48
Docs: Index, tocbibind. 101
Docs: Starred sections. 99
v0.39
\@maketitle: titling version. 650
Native LaTeX version. 286
Removed minipages. 286, 650
Supports authblk with s of class oneauthor instead of tabular. 286, 650
\AddSubtitlePublished: Added. . 288
\LWR@domulticolumn: Add: Optional vpos and # rows. 335
\LWR@restoreorigformatting: Appended with \appto instead of calling various macros. 382
\LWR@tabledatacolumntag: Don't start a data cell if see \TabularMacro. 345
\ResumeTabular: Added. 344
\TabularMacro: Added. 344
\multicolumnrow: Added. . . 342, 572
\printauthor: Removed minipages. 283
Supports authblk with s of class oneauthor instead of tabular. 283
\thanksmarkseries: Removed minipage footnotes. 651
General: 2017/09/05 1
a4wide: Added. 451
a4: Added. 451
a5comb: Added. 451
addlines: Added. 455
anysize: Added. 462
authblk: Added. 464
bigdelim: Added. 466
bigstrut: Added. 467
chngpage: Prevented. 472
ebook: Added. 478
fullwidth: Added. 515
midpage: Added. 566
supertabular: Added. 624
textarea: Added. 628
titling: Improved compatibility. . 647
titling: Removed extraneous center environments. 648
typearea: Added. 667
xtabular: Added. 686
zwpagelayout: Added. 688
Add: New optional vpos argument. 571
Add: Supports left/right border for bigdelim. 571
Docs: Reorganized tabular discussion. 108
Fix: Long text argument. 571

Titlepage \published and \subtitle removed. \AddSubtitlePublished restores. 287
titlepage: Clear pending footnotes. 281
Removed minipages. 281
titlingpage: Clear pending footnotes. 648
v0.40
\@chapcntformat: Added for tocbibind, anonchap. 264
\LWR@hline: Added. 348
\LWR@includegraphicsb: Add: Full \graphicspath support. . . . 523
\LWR@nullfonts: Fix: Long arguments for expandable command. 433
\LWR@restoreorigformatting: Improved LaTeX logos inside a lateximage. 382
Improved symbols inside a lateximage. 382
Nullified \InlineClass, etc. inside a lateximage. 382
\LWR@tabledatacolumntag: Fix for bigdelim: \ldelim, \rdelim. . . 345
\chapter: Added support for quotchap. 270
\multicolumnrow: Fix: Adapts to older multirow and xparse. . . . 342
\simplechapterdelim: Added for tocbibind, anonchap. 264
\underline: Added. 436
General: 2017/09/25 1
adjmulticol: Added. 454
anonchap: Added. 461
bigdelim: Improved documentation. 466
cuted: Added. 475
dblfnote: Added. 476
fnpos: Added. 509
graphics: Moved out of the lwarp core. 518
graphics: Restores \includegraphics and \DeclareGraphicsExtensions in a lateximage. 518
graphicx: Moved out of the lwarp core. 530
grffile: Directly supported. 530
midfloat: Added. 566
pfnote: Added. 592
quotchap: Added. 593
sectsty: Added. 608
stabular: Added. 617
tabls: Added. 626
textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support. 629
tocbibind: Improved for \simplechapter. 653
xltxtra: Fix for \showhyphens with XeLaTeX. 685
Improved bigdelim borders. . . . 571
No longer preloads xfrac. 153
v0.41
\LWR@addcmidruletrim: Add: \cmidrule trims. 330
\LWR@clearmidrules: Add: \cmidrule trims. 328
\LWR@closetabledatacell: Add: Mute > for \bottomrule. . . . 309
Fix: At/bang column with \multirow. 309
Fix: Cancel < for \multicolumn. 310
\LWR@domulticolumn: Add: \cmidrule trims. 335
Added vertical rules. 337
\LWR@nullifyNoAutoSpacing: Fix: \NoAutoSpacing in a tabular with babel-french. 348
\LWR@parsebarcolumn: Added vertical rules. 317
\LWR@printatbang: Add: \cmidrule trims. 324
Add: Mute at and bang columns for \bottomrule. 324
\LWR@printbartag: Added vertical rules. 324
\LWR@subaddcmidruletrim: Added. 330
\LWR@subcmidrule: Add: \cmidrule trims. 328
\LWR@tabledatasinglecolumntag: Add: \cmidrule trims. 325
Add: Mute < for \bottomrule. . . 325
\LWR@tabularfinishrow: Unfinished tabular rows automatically filled. 312
\mcolrowcell: Added for \multicolumrow cells. 347

General: 2017/10/07 1
Add: \cmidrule trims. 571
Added vertical rules. 571
Fix: < spec. 572
LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french. . 349
Improved rules. 350
v0.42
\@ensuredmath: Improved \ensuremath. 388
\@textsubscript: Added. 436
\@textsuperscript: Added. 436
\LWR@HTMLtextstyle: Added. 431
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment. 332
\LWR@addrulewidth: If FormatWP force explicit border. 330
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments. 406
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments. 407
\LWR@domulticolumn: If FormatWP add cell alignment. 337
\LWR@doubledollar: If FormatWP print LaTeX expression. 387
Improved \ensuremath. 387
Improved line spacing with mathjax. 387
\LWR@floatbegin: If FormatWP add a text frame. 365
\LWR@floatend: If FormatWP add a text frame. 366
\LWR@hline: If FormatWP force explicit border. 348
\LWR@hspace: If FormatWP add \quads. 441
\LWR@htmlmathlabel: If FormatWP print LaTeX expression. 394
\LWR@includegraphicsb: Fix: Filename expansion. 523
If FormatWP, use explicit size. . . 524
\LWR@rememberbtag: Fix: Numbering and naming AMS math environments. 406
\LWR@restoreorigformatting: Improved \ensuremath. 383
\LWR@rule: If FormatWP add \quads. 444
\LWR@singledollar: If FormatWP print LaTeX expression. 388
\LWR@subaddcmidruletrim: 0pt if no rule given. 330
\LWR@tabledatasinglecolumntag: If FormatWP add cell alignment. 326
\LaTeX: If FormatWP use explicit style. 446
\TeX: If FormatWP use explicit style. 446
\listoffigures: Added boolean WPMarkLOFT. 374
\listoftables: Added boolean WPMarkLOFT. 374
\marginpar: If FormatWP emulate a wrapfig. 254
\marginparBlock: If FormatWP emulate a wrapfig. 254
\tableofcontents: Added boolean WPMarkTOC. 373
\underline: If FormatWP, use explicit styles for \underline, etc. 436
General: 2017/10/30 1
\textbf and related: If FormatWP, use explicit styles for \textsc, etc. 431
algorithmicx: If FormatWP add \quads. 456
epigraph: If FormatWP add HTML styles. 481
fancybox: If FormatWP add HTML styles. 485
floatflt: Added width. 500
includegraphics: Fix: Class key. . . 522
keyfloat: If FormatWP add explicit HTML style. 540
moreverb: Simplified formatting of listings. 567
morewrites: Added. 568
multirow: If FormatWP add cell alignment. 571
overpic: Added. 589
realscripts: Fix for subscripts in a lateximage. 595
sidenotes: If FormatWP add explicit HTML style. 612
siunitx: Improved \ensuremath. . 613

soul: If FormatWP, add explicit styles. 614
textcomp: Improved \interrobangdown. 629
wrapfig: If FormatWP add explicit HTML style. 672
Added boolean WPMarkLOFT. . . . 160
Added boolean WPMarkMath. . . . 161
Added boolean WPMarkMinipages. 160
Added boolean WPMarkTOC. 160
Added boolean WPTitleHeading. 161
Docs: Added support page. 2
Docs: Improper \prevdepth. . . . 128
Docs: Reorganized math limitations 102
File: lwarp_mathjax.txt: Updated siunitx script. 206
Fix: Numbering and naming AMS math environments. 405
If FormatWP, shift section headings. 161
tabbing: Added. 294
lateximage: Fix: Numbering and naming AMS math environments. 407
center: If FormatWP use explicit text-align. 411
minipage: Added boolean WPMarkMinipages. 425
If FormatWP add a text frame. . . 423
eqnarray: Fix: Numbering and naming AMS math environments. 397
If FormatWP print LaTeX expression. 396
equation: If FormatWP print LaTeX expression. 392
LWR@BlockClassWP: Added to factor code. 239
LWR@figcaption: If FormatWP forces italic captions. 368
LWR@tabular: If FormatWP force explicit border. 350

v0.43

\LWR@domulticolumn: Fix for vertical rules. 337
Fix: Multicolumn trim. 336
\LWR@maybeprintpendingfootnotes: Added FootnoteDepth. 253
\LWR@nullfonts: Fix: Nullify dollar inside filesnames. 433
\LWR@parsetablecols: Ignore spaces in col spec. 319
\LWR@printmccoltype: Added vertical rules. 333
\LWR@restoreorigformatting: Fix: \ref in math or lateximage. . . 382
Fix: Improved math inside lateximages. 383
\LWR@section: Fix: Expansion in comparison. 265
Fix: Math in section name. 266, 268
Fix: Nullify fonts inside HTML comment. 266
\TabularMacro: \newcommand instead of \relax to fix supertabular and xtab. 344
\href: Made robust. 362
\nameref: Made robust. 360
\nolinkurl: Made robust. 362
\url: Made robust. 362
General: 2017/11/08 1
breakurl: Added. 469
hyperref: Made robust. 533, 534, 536
hyperref: \Gauge added. 536
luatodonotes: Added. 550
todonotes: Added. 663
LWR@currentautosec: Added. . . 264
Added FootnoteDepth. 250
Docs: HTML settings table. . . . 80
Docs: Reorganized HTML customization. 80

v0.44

\@currentlabelname: Adjustment for koma-script. 353
\HTMLTitle: Added. 248
\LWR@addformatwpalignment: Fix for multicolumn alignment if FormatWP. 332
\LWR@backgroundcolor: Added. . . 678
\LWR@filestart: Add \HTMLTitle. 273
Fix \HTMLAuthor. 273
\LWR@listitem: Added list and trivlist. 298
\LWR@patchlists: Added list and trivlist. 301
\LWR@strresult: Fix: \providecommand. 308

\addcontentsline: Automatic \LWR@newfloatanchor. 370
\chapter: Add preamble for koma-script. 270
\marginparBlock: Added. 254
\nopagecolor: xcolor: Fix for \nopagecolor. 679
\part: Add preamble for koma-script. 269
\title: Added \thetitle. 248
General: 2017/11/22 1
AMS environments: Fix: Groups for lateximages. 397
algorithmicx: Improved comment symbol. 456
atbegshi: Added. 463
cancel: Added. 470
changepage: Additional options. 472
easy-todo: Added. 477
fancyref: Added. 489
fixmetodonotes: Added. 498
fixme: Added. 497
fontenc: Allowed after lwarp. . . . 509
hang: Added. 530
ifoddpage: Added. 538
ltxtable: Added. 549
luatodonotes: Improved. 550
lwarp-patch-komascript: Added. . 690
overpic: Fix: Groups for lateximages. 589
pdfsync: Fixes. 591
preview: Added. 593
scrextend: Added. 600
scrhack: Added. 603
scrlayer-notecolumn: Added. . . . 605
scrlayer-scrpage: Added. 606
scrlayer: Added. 604
section: Added. 607
soulpos: Added. 616
soulutf8: Added. 617
supertabular: Fix for caption. . . . 625
tocbasic: Added. 652
tocloft: Added \newlistentry. . 659
tocloft: Improved \newlistof. . . 660
tocstyle: Added. 661
todonotes: Improved. 663
todo: Added. 662
typearea: Added expert commands. 667
watermark: Added. 672
xcolor: Added \LWR@currenttextcolorstyle. 677
xcolor: Added \LWR@findcurrentcolor. . . . 677
xcolor: Added \LWR@textcurrentcolor. . . . 677
xtab: Fix for caption. 687
Adjustment for koma-script. 148, 150
If pdfLaTeX, require T1 and UTF8 encoding. 135
picture: overpic: Fix: Groups for lateximages. 420
tikzpicture: tikz: Fix: Groups for lateximages. 639
list: Added list and trivlist. . 298
LWR@nestspan: Added list and trivlist. 233
v0.45
\@currentHref: Added. 361
\@donoparitem: Modified for HTML. 296
\@item: Modified for HTML. 296
\@mklab: Modified for HTML. 296
\CSSFilename: Improved filenames with underscores. 247
\LWR@LwarpStart: Fix: Lateximages on incorrect pages with Mathjax. 277
\LWR@filenamenoblanks: Fix: Section names with \ 257
Fix: Section names with underscores. 257
\LWR@includegraphicsb: Improved URLs with underscores. 523
\LWR@newautoidanchor: Fix: No anchor if frozen autoid. 366
\LWR@notmemoirloadafter: Added. 143
\LWR@printpendingmpfootnotes: Added. 253
\LWR@startref: Fix: Labels with underscores. 359
\LWR@subhyperref: Improved URLs with underscores. 361
\LWR@subhyperrefclass: Improved URLs with underscores. 362
\LWR@sublabel: Fix: Labels with underscores. 357
\LWR@tabledatacolumntag: Fix: Empty line between rows. . . . 347
\chapter: Add optional heading title for memoir. 270
\newpage: Added. 438

\nolinkurl: Improved URLs with underscores. 362
\normalmarginpar: Added. 255
\reversemarginpar: Added. 255
\section: Add optional heading title for memoir. 270
\tableofcontents: Fix: Empty sidetoc. 373
Fix: Patch \AtBeginDocument. . . 373
\url: Improved URLs with underscores. 362
General: 2018/01/14 1
array: Added. 463
babel-french: Robust commands. 231
backref: Added. 465
changebar: Added. 471
cite: Added. 473
continue: Added. 474
endfloat: Added. 479
epigraph: Support for memoir. . . 482
fancyvrb: Improvements. . . 490, 491
flafter: Added. 498
fltrace: Added. 507
footnpag: Added. 512
fwlw: Added. 516
hanging: Added. 532
hyperref: Fix for underscores in \hyperref. 534
lwarp-patch-memoir: Added. . . . 692
memhfixc: Added. 564
memoir: Added. 449
natbib: Added. 573
pagesel: Added. 590
prettyref: Added. 593
subfigure: Added. 624
subfig: Fix for subcaption end tag. 621
subfig: Fix: Math in subcaptions. 619
textfit: Added. 632
titleref: Added. 643
turnthepage: Added. 666
Allows memoir's preloaded packages. 144
Docs: xparse warnings. 116
Docs: Fix for double hyphens. . . 61
Docs: Improved install instructions. 61
Docs: Improved MiKTEX install instructions. 57
Docs: Moved table so doesn't interfere with install docs. 56
File: lwarp_mathjax.txt: Allow MathJax inside tabbing. 206
File: lwarp_mathjax.txt: Allow MathJax inside verse. 206
Fix: Empty sidetoc. 373
Improved filenames with underscores. 533
Improved URLs with underscores. 469
Improved: Robust \, \ , and \textellipsis commands. . . 437
Separate LWR@thisautoidWP for word processor s. 366
thebibliography: Patched to emphasize titles. 381
minipage: Fix: Improper \prevdepth. 423, 425
v0.46
\LWR@closeparagraph: Fix: Tabular empty lines. 243
\LWR@closeprevious: Fix: Stack unnesting. 229
\LWR@forcenewpage: Fix: Improper \prevdepth. 229
\LWR@lookforpackagename: Fix: Spaces in \usepackage. 154
\LWR@providelength: Added. . . . 136
\LWRPrintStack: Name changed from \PrintStack. 228
\makebox: Fix: Lateximage in a \makebox. 427
\popclose: Fix: Stack unnesting. . . 222
\pushclose: Fix: Stack unnesting. . 221
General: 2018/01/23 1
LWR@tabularpardepth added. . . 307
amsthm: Adapted to trivlist changes. 459
mdframed: Fixes for SVG math or lateximage in title. 558
mdframed: Fixes for footnotes. . . 559
ntheorem: Adapted to trivlist changes. 577
theorem: Adapt to trivlist changes. 635, 637
list: Fix: Stack unnesting. 299
LWR@tabular: Fix: Tabular empty lines. 352

§ 301 # Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\$ *386*
\& *224*, 5358
\(. 7047
\) 7047
\, ... *91*
\@@setcpageref 7774
\@@setcref 7770
\@@setcrefrange 7771
\@author *280*
\@begintheorem 5043
\@captype 6576
\@chapcntformat 4372
\@currentHref 6472
\@currentlabelname 6332
\@date *280*
\@dlbfloat 6548
\@donoparitem 5066
\@endtheorem 5055
\@ensuredmath 7055
\@float 6548
\@fnsymbol 4852
\@footnotetext 4089
\@include 526
\@item 5079
\@makecaption 6578
\@makefnmark 4060
\@makefntext 4059
\@maketitle 49, 4875
\@mklab 5063
\@mpfootnotetext 4091
\@nbitem 5149
\@opargbegintheorem 5049
\@seccntformat 4370
\@starttoc 6692
\@textsubscript 8138
\@textsuperscript 8134
\@title *280*
\@wrglossary 6874
\@wrindex 6865
\\ *438*
$ *387*
$$ *387*
~ 91

A

a4 (package) *451*
a4wide (package) *451*
a5comb (package) *451*
abstract (environment) 4938
abstract (package) *99*, *451*
\abstractname *82*, 4937
accents 261
\addcontentsline 6640
addlines (package) *455*
\AddSubtitlePublished 4900
adjmulticol (package) *454*
Adobe (program) *53*
\affiliation 4755
afterpackage (package) *150*
afterpage (package) *455*
algorithmic
  with newfloat, trivfloat 666
algorithmicx (package) *456*
algorithmx (package) *112*
align (environment) 7322
align* (environment) 7345
alignment tab character &, misplaced 108, 302, 310
alltt (package) *457*
\AmS 8371
AMSmath
  split miss-numbered 105, 577
amsmath (package) *152*
amsthm (package) *457*
\and *280*
anonchap (package) *461*
anysize (package) *462*
appendix (package) *100*, *462*
arabicfront (package) *463*
array (package) *463*
AsciiDoc (program) *53*
AsciiDoctor (program) *53*
Asciidoctor-LaTeX (program) *53*
atbegshi (package) *463*
\attrib *115*, *290*, *670*

\attribution 4948
authblk (package) *100*, *464*
author
  HTML meta tag 90, 248
\author *89*, *280*
autosec *263*

B

babel (package) *116*
\backmatter 4351
backref (package) *465*
balance (package) *465*
\BaseJobname 3470
BaseJobname (option) *79*, *139*
baseline
  tabular 319
\bfseries 8123
\BibTeX 8366
bigdelim (package) *112*, *466*
bigstrut (package) *467*
BlockClass (environment) 3803
\BlockClassSingle 3811
bookmark (package) *467*
booktabs (package) *468*
booleans:
  CombineHigherDepths *82*, *256*
  FileSectionNames *82*, *225*
  FormatEPUB *117*, *159*
  FormatWP *119*, *159*
  HTMLDebugComments *159*
  LWR@amsmultline *394*
  LWR@copiedsidetoc *373*
  LWR@doingapar *241*
  LWR@doingcmidrule *306*
  LWR@doinghline *306*
  LWR@doingstartpars *240*
  LWR@doingtbrule *306*
  LWR@emptyatbang *307*
  LWR@exitingtabular *345*
  LWR@freezethisautoid *366*
  LWR@infloatrow *522*
  LWR@intabularmetadata *307*
  LWR@isstartingequation *405*
  LWR@minipagefullwidth *422*
  LWR@minipagethispar *422*
  LWR@skipatbang *307*
  LWR@skippingmcolrowcell *307*
  LWR@skippingmrowcell *306*
  LWR@starredlongtable *338*
  LWR@startedrow *306*
  LWR@tableparcell *306*
  LWR@tabularcelladded *306*
  LWR@tabularmutemods *345*
  LWR@tracinglwarp *158*
  LWR@validtablecol *319*
  LWR@verbtags *292*
  mathjax *138*
  usingOSWindows *137*
  warpingHTML *138*
  warpingprint *138*
  WPMarkFloats *121*, *159*
  WPMarkLOFT *122*, *160*
  WPMarkMath *122*, *161*
  WPMarkMinipages *121*, *160*
  WPMarkTOC *121*, *160*
  WPTitleHeading *122*, *161*
boxedminipage (package) *468*
boxedminipage2e (package) *469*
\boxframe 164
breakurl (package) *469*
bugs 128
BVerbatim (environment) 196

C

calc (package) *152*
Calibre 117
cancel (package) *470*
capt-of (package) *370*
caption (package) *112*, *152*, *370*
caption2 (package) *471*
\caption@begin 6621
\caption@end 6621
\captionlistentry 6625
\captionof 6654
\CaptionSeparator 6577
ccaption (package) *471*
center (environment) 7607
\centering *276*
changebar (package) *471*
changepage (package) *472*
chappg (package) *473*
\chapter 4529
chngpage (package) *472*
cite (package) *473*
\citetitle 4955
classs:
  internet *52*
  komascript *114*
  memoir *114*, *115*, *670*
\cleardoublepage 8267

\clearpage 8267
cleveref (package) *98*, *418*
cmap (package) *77*
\color *679*
color (package) *107*, *474*
\colorbox 72
\colorboxBlock 82, 7666
CombineHigherDepths (boolean) . *82*, *256*
comment (package) *141*
Computer Modern 76
\ConTeXt 8363
continue (package) *474*
counters:
  FileDepth *82*, *255*
  FootnoteDepth *82*, *250*
  lofdepth *377*
  lotdepth *377*
  LWR@currentautosec *264*
  LWR@externalfilecnt *387*
  LWR@htmlfilenumber *226*
  LWR@latestautopage *368*
  LWR@lateximagedepth *403*
  LWR@lateximagenumber *403*
  LWR@LIpage *403*
  LWR@midrulecounter *327*
  LWR@minipagedepth *421*
  LWR@nextautoid *367*
  LWR@nextautopage *367*
  LWR@nextequation *389*
  LWR@prevFileDepth *263*
  LWR@spandepth *240*
  LWR@startingequation *405*
  LWR@tablecolindex *308*
  LWR@tablecolspecindex *308*
  LWR@tablecolspecwidth *308*
  LWR@tabletotalcols *308*
  LWR@tabletotalcolsnext *308*
  LWR@tabularDepth *307*
  LWR@tabularpardepth *307*
  LWR@thisautoid *366*
  LWR@thisautoidWP *366*
  SideTOCDepth *81*, *375*
  tocdepth *81*
\cpagerefFor 7773
crop (package) *474*
cross-references
  missing or incorrect 129
CSS
  file selection 85
  lwarp.css 86
  per HTML page 85
  project-specific changes 86
\CSSFilename *83*, *85*, 4015
cuted (package) *475*
cutwin (package) *475*

D

danger icon 134
\date *89*
dblfnote (package) *476*
dcolumn (package) *476*
debugging 128
  HTML debug comments 159
\DeclareGraphicsExtensions 2
Deja Vu 76
description
  HTML meta tag 89, 248
description (environment) 5213
displaymath (environment) 7069
\dotfill 8150
draftwatermark (package) *477*

E

easy-todo (package) *477*
ebook (package) *478*
ellipsis (package) *478*
\emph 8020
emptypage (package) *479*
\end@dlbfloat 6564
\end@float 6564
\EndDefiningTabulars 5352
endfloat (package) *479*
endnotes (package) *100*, *480*
\enlargethispage 8266
\enskip *439*, 8219
enumerate (environment) 5195
enumerate (package) *480*
enumitem (package) *481*
environ (package) *152*
environments:
  abstract 4938
  align 7322
  align* 7345
  BlockClass 3803
  BVerbatim 196
  center 7607
  description 5213
  displaymath 7069
  enumerate 5195
  eqnarray 7177

equation 7116
equation* 7140
fcolorminipage 148, 7708
flalign 7368
flalign* 7391
flushleft 7623
flushright 7615
fminipage *94*, 7963, 8011
gather 7276
gather* 7299
itemize 5190
lateximage *402*, 7524
list 5161
longtable 2
LWR@BlockClassWP 3827
LWR@figcaption 6587
LWR@nestspan 3667
LWR@tabular 6210
LWRprint@fminipage 7987
math 7068
minipage 7810
multline 7227
multline* 7251
picture *420*, 7791
quote 4963
tabbing 5031
thebibliography 6890
theindex 6843
tikzpicture 7
titlepage *88*, 4772
titlingpage 14, *89*
verbatim 5021
verse 2
warpall *87*, 107
warpHTML 57, *84*, *87*, 108
warpprint *84*, *87*, 108
epigraph (package) *481*
EPS images 106, 413
epstopdf (package) *106*, *108*, *413*
EPUB
conversion software 117
HTML conversion settings . . . 117, 159
eqnarray (environment) 7177
equation
miss-numbered 105, 577
equation (environment) 7116
equation* (environment) 7140
error messages 128
eso-pic (package) *482*
etoolbox (package) *136*
everyhook (package) *151*
everypage (package) *483*
everyshi (package) *483*
expl3 (package) *150*
export
to word processor 119
extramarks (package) *484*

F

fancybox (package) *95*, *484*
fancyhdr (package) *488*
fancyref (package) *489*
fancyvrb (package) *490*
FAQ 128
fbox
around a minipage 94
\fbox *94*, 7946
\fboxBlock *94*, 7955, 8010
\fcolorbox 94, 7672
\fcolorboxBlock 114, 7707
fcolorminipage (environment) 148, 7708
figcaps (package) *496*
file
inaccessible 81, 82
underscore 81, 92
filecontents (package) *151*
FileDepth (counter) *82*, *255*
files:
glyphtounicode *77*
lwarp.css *86*, *168*
lwarp.xdy *90*, *205*
lwarp_formal.css *200*
lwarp_mathjax.txt *206*
lwarp_sagebrush.css *195*
lwarp_tutorial.txt *63*
lwarpmk.conf *166*
lwarpmk.lua *126*
project.css *86*
project.lwarpmkconf *167*
project_html.tex *166*
sample_project.css *86*, *205*
tutorial.tex *63*
FileSectionNames (boolean) *82*, *225*
fix2col (package) *496*
fixme (package) *116*, *496*
fixmetodonotes (package) *498*
flafter (package) *498*
\flagverse *671*
flalign (environment) 7368
flalign* (environment) 7391

Flare (program) 53
float (package) 112, 499
floatflt (package) 500
floatpag (package) 501
floatrow (package) 113, 501
fltrace (package) 507
\flushbottom 3584
flushend (package) 507
flushleft (environment) 7623
flushright (environment) 7615
fminipage (environment) . 94, 7963, 8011
fncychap (package) 508
fnpos (package) 509
font
  bitmapped 76
  Computer Modern 76
  Deja Vu 76
  ligatures 76, 77
  packages 76
  selection 76
  size — lateximage 103, 385, 403
  size — math, SVG 103, 385, 403
  size — xfrac 683
fontenc (package) 77, 509
fontspec (package) 76, 509
footmisc (package) 509
footnote (package) 511
FootnoteDepth (counter) 82, 250
footnotehyper (package) 512
footnotes 249
footnpag (package) 512
for
  HTML & PRINT 134
  HTML output 134
  PRINT output 134
\ForceHTMLPage 99, 4326
\ForceHTMLTOC 99, 4332
FormatEPUB (boolean) 117, 159
FormatWP (boolean) 119, 159
\framebox 7925
framed (package) 512
framed objects 94
FrameMaker (program) 53
Frequently Asked Questions 128
\frontmatter 4348
ftnright (package) 515
fullpage (package) 515
fullwidth (package) 515
\fup 8142
\fussy 3586
fwlw (package) 516

G

gather (environment) 7276
gather* (environment) 7299
GELLMU (program) 52
generator
  HTML meta tag 273
geometry (package) 149, 516
gettitlestring (package) 150
GladTeX (program) 52
glossaries (package) 100, 517
glossary
  language 74
  processing 74
glyphtounicode (file) 77
graphics
  EPS files 106, 413
  PDF files 106, 413
graphics (package) 106, 518
graphicx (package) 106, 530
grffile (package) 107, 530

H

hang (package) 530
hanging (package) 532
heading, word processor 122
Hevea (program) 52
\hfill 8148
\HomeHTMLFilename 3472
HomeHTMLFilename (option) . . . 79, 81, 140
horizontal space 91
  between minipages 439
\href 6504
\hrulefill 8149
\hskip 91
\hspace 91, 439, 8257
htlatex (program) 52
HTML
  commands for a successful conversion 91
  conversion settings 80
    debug comments 159
    EPUB 117, 159
    word processor 119, 159, 160
  filename generation 84
  headings 132
  in place of SVG images 69, 130
  meta tag
    author 90, 248

description 89, 248
generator 273
Title 248
title 90, 248
viewport 273
tabular column conversion 320
\HTMLAuthor 83, 90, 4041
HTMLDebugComments (boolean) 159
\HTMLDescription 83, 89, 4046
\HTMLentity 3453
\HTMLFilename 3471
HTMLFilename (option) 79, 81, 140
\HTMLFirstPageTop 83, 4003
htmlglossary (option) [lwarpmk] 101, 517
\HTMLLanguage 83, 4603
HTMLleftmargini (length) 115, 290, 291, 670
\HTMLPageBottom 83, 280, 4009
\HTMLPageTop 83, 4006
\HTMLTitle 83, 90, 4036
\HTMLunicode 3461
HTMLvleftskip (length) . 115, 290, 291, 670
\hyperindexref 6881
hyperref (package) 98, 361, 533
\hypertoc 6782
\hypertocfloat 6802
hyperxmp (package) 537

I

icon
warning 134
idxlayout (package) 537
ifoddpage (package) 538
ifplatform (package) 136
images
EPS 106, 413
graphicx package 413
in strange places 129
PDF 106, 413
showing as HTML 69, 130
Improper \prevdepth. 128
\include 526
\includegraphics 232
indentfirst (package) 539
InDesign (program) 53
index
language 74
placement and TOC options 101
processing 67, 68
UTF-8 77
IndexLanguage (option) 74, 79, 101, 139, 517, 652
\InlineClass 3819
inputenc (package) 77, 539
internet (class) 52
item
empty 92, 295
itemize (environment) 5190
\itshape 8128

J

JavaScript
MathJax 102, 384

K

\kern 91
keyfloat (package) 113, 539
komascript (class) 114
kvoptions (package) 138

L

label
in HTML 276
math environment 394
valid characters 97
Label(s) may have changed 128
language
glossary 74
index 74
language HTML metadata 272
\LaTeX 8339
LaTeX2HTML (program) 52
\LaTeXe 8339
lateximage (environment) 402, 7524
\LateximageFontSizeName 7425
lateximages
font size 103, 385, 403
latexmk (option) 79, 81, 140
LaTeXML (program) 52
layout (package) 541
Leaders not followed by proper glue. . . 128
lengths:
HTMLleftmargini . . 115, 290, 291, 670
HTMLvleftskip 115, 290, 291, 670
LWR@minipageheight 421
LWR@minipagewidth 421
LWR@tempheight 440
LWR@tempraise 440
LWR@tempwidth 440
\LWR@cmidrulewidth 328

\LWR@heavyrulewidth 327
\LWR@lightrulewidth 327
\LWR@thiscmidrulewidth 328
\VerbatimHTMLWidth 291
vleftmargini 115, 290, 670
vleftskip 115, 290, 670, 671
letltxmacro (package) 136
letterspace (package) 541
lettrine (package) 542
LibreOffice
conversion recommendations 122
import into 119
section headings 122
LibreOffice (program) 53
ligatures 76, 77, 149
line numbers 134
\linebreak 8260
\LinkHome 3522
Linux (program) 86, 137
lips (package) 543
list (environment) 5161
list item, empty 92, 295
listings (package) 543
\listof 6746
\listoffigures 6720
\listoftables 6733
lmodern (package) 76
lofdepth (counter) 377
longtable (environment) 2
longtable (package) 111, 547
lotdepth (counter) 377
lscape (package) 549
ltcaption (package) 549
ltxtable (package) 549
LuaLaTeX
detection 148
file & section names 261
\LuaLaTeX 8357
LuaLaTeX (program) [requirement] . . . 56
\LuaTeX 8357
luatodonotes (package) 116, 550
lwarp
loading 78
options 78
lwarp (package) 78
lwarp-patch-komascript (package) . 689
lwarp-patch-memoir (package) 691
lwarp.css (file) 86, 168
lwarp.xdy
customizing 90
lwarp.xdy (file) 90, 205
lwarp_formal.css (file) 200
lwarp_mathjax.txt (file) 206
lwarp_sagebrush.css (file) 195
lwarp_tutorial.txt (file) 63
lwarpmk
customizing 126
[lwarpmk]:
htmlglossary (option) 101, 517
printglossary (option) 101, 517
lwarpmk (option) 79, 139
lwarpmk (program) 126, 208
lwarpmk.conf (file) 166
lwarpmk.lua (file) 126
\LWR@addcmidruletrim 5781
\LWR@addcmidrulewidth 5806
\LWR@addformatwpalignment 5813
\LWR@addleftmostbartag 5648
\LWR@addmathjax 7105
\LWR@addrulewidth 5786
\LWR@afterendverbatim 5005
\LWR@amsmathbody 7511
\LWR@amsmathbodynumbered 7517
LWR@amsmultline (boolean) 394
\LWR@atbeginverbatim 4990
\LWR@backgroundcolor 48
\LWR@beginhideamsmath 7162
\LWR@blackborderpadding 7941
LWR@BlockClassWP (environment) . . 3827
\LWR@botnavigation 3530
\LWR@caption@begin 6602
\LWR@caption@end 6612
\LWR@clearmidrules 5721
\LWR@closeparagraph 3908
\LWR@closeprevious 3557
\LWR@closetabledatacell 5315
\LWR@cmidrulewidth (length) 328
LWR@colafterspec 309
LWR@colatspec 308
LWR@colbangspec 308
LWR@colbarspec 309
LWR@colbeforespec 309
LWR@copiedsidetoc (boolean) 373
\LWR@copyfile 561
\LWR@createautosec 4357
LWR@currentautosec (counter) 264
\LWR@currentcss 4014
\LWR@currenttextcolor 8269
\LWR@descitem 5200
\LWR@docmidrule 5752

LWR@doingapar (boolean) *241*
LWR@doingcmidrule (boolean) *306*
LWR@doinghline (boolean) *306*
LWR@doingstartpars (boolean) *240*
LWR@doingtbrule (boolean) *306*
\LWR@domulticolumn 5908
\LWR@doubledollar 7013
LWR@emptyatbang (boolean) *307*
\LWR@endhideamsmath 7170
\LWR@endofline 8202
\LWR@ensuredoingapar 3886
\LWR@equationtag 7483
LWR@exitingtabular (boolean) *345*
LWR@externalfilecnt (counter) *387*
LWR@figcaption (environment) 6587
\LWR@filenamenoblanks 4188
\LWR@filestart 4610
\LWR@findword 448
\LWR@floatbegin 6527
\LWR@floatend 6550
\LWR@footnotetext 4061
\LWR@forceminwidth 7930
\LWR@forcenewpage 3574
LWR@freezethisautoid (boolean) . . . *366*
\LWR@futurenonspacelet 5261
\LWR@FVstyle 37
\LWR@getexparray 3439
\LWR@getmynexttoken 5268
\LWR@heavyrulewidth (length) *327*
\LWR@hidelatexequation 7091
\LWR@hline 6180
\LWR@hspace 8227
\LWR@htmlblockcomment 3753
\LWR@htmlblocktag 3755
\LWR@htmlclosecomment 3725
\LWR@htmlcomment 3746
\LWR@htmldivclass 3780
\LWR@htmldivclassend 3783
\LWR@htmlelement 3793
\LWR@htmlelementclass 3767
\LWR@htmlelementclassend 3772
\LWR@htmlelementclassline 3786
\LWR@htmlelementend 3796
LWR@htmlfilenumber (counter) *226*
\LWR@htmlfileref 6359
\LWR@htmlmathlabel 7147
\LWR@htmlmulticolumn 5944
\LWR@htmlopencomment 3725
\LWR@htmlrefsectionfilename 3515
\LWR@HTMLsanitize 7428
\LWR@HTMLsanitizeexpand 7449
\LWR@htmlsectionfilename 3482
\LWR@htmlspan 3702
\LWR@htmlspanclass 3710
\LWR@htmltag 3720
\LWR@htmltagc 3655
\LWR@HTMLtextstyle 8015
\LWR@includegraphicsb 105
\LWR@indexitem 6853, 6857, 6861
LWR@infloatrow (boolean) *522*
\LWR@instertatbangcols 5307
LWR@intabularmetadata (boolean) . . *307*
LWR@isstartingequation (boolean) . . *405*
\LWR@itemizeitem 5183
LWR@latestautopage (counter) *368*
LWR@lateximagedepth (counter) *403*
\LWR@lateximagedepthref 6363
LWR@lateximagenumber (counter) . . . *403*
\LWR@lateximagenumberref 6366
\LWR@lateximagesfile 523
\LWR@lightrulewidth (length) *327*
LWR@LIpage (counter) *403*
\LWR@listitem 5154
\LWR@loadafter 118
\LWR@loadbefore 132
\LWR@loadnever 141
\LWR@longtabledatacaptiontag . . . 5955
\LWR@lookforpackagename 451
\LWR@LwarpEnd 4732, 8393
\LWR@LwarpStart 4663, 8393
\LWR@maketitlesetup 33, 4844
\LWR@maybenewtablerow 5593
\LWR@maybeprintpendingfootnotes 4128
LWR@midrulecounter (counter) *327*
LWR@midrules *327*
LWR@minipagedepth (counter) *421*
LWR@minipagefullwidth (boolean) . . *422*
LWR@minipageheight (length) *421*
\LWR@minipagestartpars 8207
\LWR@minipagestoppars 8210
LWR@minipagethispar (boolean) *422*
LWR@minipagewidth (length) *421*
\LWR@multicolother 5856
\LWR@multicolpartext 5851
\LWR@multicolskip 5862
\LWR@multirowborder 3
\LWR@mynexttoken 5260
\LWR@myshorttoc 6664
\LWR@nameref 6356
LWR@nestspan (environment) 3667

\LWR@newautoidanchor 6570
\LWR@newhtmlfile 4259
\LWR@newlabel 6397
\LWR@newref 6448
LWR@nextautoid (counter) *367*
LWR@nextautopage (counter) *367*
LWR@nextequation (counter) *389*
\LWR@nohspace 8256
\LWR@notmemoirloadafter 129
\LWR@nullfonts 8085
\LWR@nullifyNoAutoSpacing 6197
\LWR@openparagraph 3891
\LWR@origcolspec 5301
\LWR@originname 246
\LWR@originnames 253
\LWR@parseaftercolumn 5480
\LWR@parseatcolumn 5426
\LWR@parsebangcolumn 5450
\LWR@parsebarcolumn 5489
\LWR@parsebeforecolumn 5471
\LWR@parseDcolumn 5516
\LWR@parsenormalcolumn 5500
\LWR@parsepcolumn 5512
\LWR@parsetablecols 5521
\LWR@patchlists 5221
LWR@prevFileDepth (counter) *263*
\LWR@printatbang 5625
\LWR@printbartag 5615
\LWR@printmccoldata 5865
\LWR@printmccoltype 5830
\LWR@printpendingfootnotes 4118
\LWR@printpendingmpfootnotes . . . 4137
\LWR@printthetitle 4796
\LWR@providelength 26
\LWR@ProvidesPackageDrop 510
\LWR@ProvidesPackagePass 498
\LWR@pushoneclose 4360
\LWR@quickfile 520
\LWR@remembertag 7487
\LWR@requesttoc 4725
\LWR@requirepackagenames 447
\LWR@restoreorigaccents 778
\LWR@restoreorigformatting 6910
\LWR@restoreoriglists 5246
\LWR@restoreorigprintxcolor 7746
\LWR@rotboxorigin 245
\LWR@rotstyle 91
\LWR@rule 8270
\LWR@sanitize 3443
\LWR@sanitized 3442
\LWR@scalestyle 94
\LWR@section 4374
\LWR@sectionumber 4354
\LWR@setexparray 3434
\LWR@setlatestname 6334
\LWR@setOSWindows 36
\LWR@sidetoc 6759
\LWR@singledollar 7031
LWR@skipatbang (boolean) *307*
LWR@skippingmcolrowcell (boolean) . *307*
LWR@skippingmrowcell (boolean) . . . *306*
LWR@spandepth (counter) *240*
\LWR@splabel 6369
LWR@starredlongtable (boolean) . . . *338*
LWR@startedrow (boolean) *306*
LWR@startingequation (counter) . . . *405*
\LWR@startingequationtag 7482
\LWR@startnewdepth 4364
\LWR@startpars 3958
\LWR@startref 6410
\LWR@stoppars 3976
\LWR@stripperiod 6333
\LWR@strresult 5299
\LWR@subaddcmidruletrim 5770
\LWR@subcmidrule 5737
\LWR@subhtmlelementclass 3760
\LWR@subhyperref 6477
\LWR@subhyperrefclass 6491
\LWR@subinlineimage 6520
\LWR@sublabel 6374
\LWR@subnewref 6442
\LWR@subtableofcontents 6682
\LWR@subtabularhtmlcolumns 6022
\LWR@syncmathjax 7071
LWR@tablecolindex (counter) *308*
\LWR@tablecolspec 5298
LWR@tablecolspecindex (counter) . . . *308*
LWR@tablecolspecwidth (counter) . . . *308*
\LWR@tabledatacolumntag 6099
\LWR@tabledatasinglecolumntag . . 5662
LWR@tableparcell (boolean) *306*
LWR@tabletotalcols (counter) *308*
LWR@tabletotalcolsnext (counter) . . *308*
LWR@tabular (environment) 6210
LWR@tabularcelladded (boolean) . . . *306*
LWR@tabularDepth (counter) *307*
\LWR@tabularendofline 5409
\LWR@tabularfinishrow 5370
\LWR@tabularhtmlcolumns 6032
\LWR@tabularleftedge 5653

LWR@tabularmutemods (boolean) *345*
LWR@tabularpardepth (counter) *307*
\LWR@tdaddstyle 5758
\LWR@tdendstyles 5764
\LWR@tdstartstyles 5757
LWR@tempheight (length) *440*
LWR@tempraise (length) *440*
LWR@tempwidth (length) *440*
LWR@thisautoid (counter) *366*
LWR@thisautoidWP (counter) *366*
\LWR@thiscmidrulewidth (length) . . . *328*
\LWR@thiscolspec 5661
\LWR@thisfilename 4186
\LWR@thisnewfilename 4187
\LWR@titlingmaketitle 76, 4892
\LWR@topnavigation 3527
\LWR@traceinfo 589
LWR@tracinglwarp (boolean) *158*
LWR@trimlrules *327*
LWR@trimrrules *327*
LWR@validtablecol (boolean) *319*
LWR@verbtags (boolean) *292*
\LWR@vspace 8258
\LWR@WPcell 5809
LWRprint@fminipage (environment) . 7987
\LWRPrintStack 3535
\LWRsetnextfloat 6581
\LyX 8374

M

Mac OS (program) *86*, *137*
Madcap (program) *53*
\mainmatter 4344
\makebox 7902
makeidx (package) *101*, *151*
\MakeIndex 8366
\maketitle 40, *88*, 4856
margin
  numbers 134
  tags 134
marginfit (package) *552*
marginfix (package) *552*
marginnote (package) *553*
\marginpar *93*, *254*, 4149
\marginparBlock *93*, *254*, 4160, 4177
\markboth 3581
\markright 3582
markup languages 53
math
  font size — SVG 103, 385, 403
  labels — valid characters 97
  mathjax option 139
  MathJax summary 103, 385
  mathsvg option 139
  showing as HTML 69, 130
  SVG summary 103, 384
  word processor conversion 122
math (environment) 7068
MathJax
  custom macros 105, 386
  mathjax option 139
  subequations 104, 385
  summary 103, 385
  tagged equations 104, 385
MathJax (program) *103*, *104*, *385*
MathJax (program) [requirement] *56*
mathjax (boolean) *138*
mathjax (option) *78*, *81*, *139*
mathsvg (option) *78*, *81*, *139*
mcaption (package) *553*
\mcolrowcell 6170
mdframed (package) *97*, *554*
\mdseries 8122
memhfixc (package) *564*
memoir
  options clash 114
  verse 670
memoir (class) *114*, *115*, *670*
meta tag, HTML
  author 90, 248
  description 89, 248
  generator 273
  Title 248
  title 90, 248
  viewport 273
metalogo (package) *565*
microtype (package) *77*, *149*, *565*
midfloat (package) *566*
midpage (package) *566*
\MiKTeX 8373
minipage
  framed 94
  horizontal space between 439
minipage (environment) 7810
\minipagefullwidth 7802
misplaced \noalign 111, 304
misplaced alignment tab character & . .
  . . . . . 108, 302, 310
Missing $ inserted. 128
missing sections 81, 82

moreverb (package) 567
morewrites (package) 568
mparhack (package) 569
\mrowcell 6167
MS-Windows (program) 86, 137
multicol (package) 569
multicolumn
  with multirow 572
\multicolumnrow 35, 6052
multirow
  with multicolumn 572
\multirow 571
multirow (package) 570
multline (environment) 7227
multline* (environment) 7251

N

\Nameref 6469
\nameref 6460
nameref (package) 573
natbib (package) 573
needspace (package) 574
newclude (package) 115, 575
newfloat
  with trivfloat, algorithmic 666
\newfloat 4
newfloat (package) 152
\newline 8200
\newpage 8197
\newtheorem 294
newtxmath (package) 106
newunicodechar (package) 77, 575
nextpage (package) 575
nicefrac (package) 105
\nolinebreak 8261
\nolinkurl 6508
nonumonpart (package) 576
\nopagebreak 8265
\nopagecolor 71
nopageno (package) 576
\normalfont 8130
\normalmarginpar 4174
nowidow (package) 576
ntheorem (package) 105, 577
\numberline 6777
numbers
  left margin 134
numindex (option) [tocbibind] . . 102, 653

O

OpenOffice (program) 53
options clash, memoir 114
options:
  [lwarpmk]:
    htmlglossary 101, 517
    printglossary 101, 517
  [tocbibind]:
    numindex 102, 653
  [tocloft]:
    titles 100
  BaseJobname 79, 139
  HomeHTMLFilename 79, 81, 140
  HTMLFilename 79, 81, 140
  IndexLanguage 74, 79, 101, 139, 517, 652
  latexmk 79, 81, 140
  lwarpmk 79, 139
  mathjax 78, 81, 139
  mathsvg 78, 81, 139
  OSWindows 79, 86, 137, 140
  warpHTML 78, 138
  warpprint 78, 138
  xdyFilename 79, 90, 139
\OSPathSymbol 35
OSWindows (option) 79, 86, 137, 140
overpic (package) 108, 589

P

packages
  required 148
packages:
  a4 451
  a4wide 451
  a5comb 451
  abstract 99, 451
  addlines 455
  adjmulticol 454
  afterpackage 150
  afterpage 455
  algorithmicx 456
  algorithmx 112
  alltt 457
  amsmath 152
  amsthm 457
  anonchap 461
  anysize 462
  appendix 100, 462
  arabicfront 463
  array 463
  atbegshi 463

authblk 100, 464
babel 116
backref 465
balance 465
bigdelim 112, 466
bigstrut 467
bookmark 467
booktabs 468
boxedminipage 468
boxedminipage2e 469
breakurl 469
calc 152
cancel 470
capt-of 370
caption 112, 152, 370
caption2 471
ccaption 471
changebar 471
changepage 472
chappg 473
chngpage 472
cite 473
cleveref 98, 418
cmap 77
color 107, 474
comment 141
continue 474
crop 474
cuted 475
cutwin 475
dblfnote 476
dcolumn 476
draftwatermark 477
easy-todo 477
ebook 478
ellipsis 478
emptypage 479
endfloat 479
endnotes 100, 480
enumerate 480
enumitem 481
environ 152
epigraph 481
epstopdf 106, 108, 413
eso-pic 482
etoolbox 136
everyhook 151
everypage 483
everyshi 483
expl3 150
extramarks 484
fancybox 95, 484
fancyhdr 488
fancyref 489
fancyvrb 490
figcaps 496
filecontents 151
fix2col 496
fixme 116, 496
fixmetodonotes 498
flafter 498
float 112, 499
floatflt 500
floatpag 501
floatrow 113, 501
fltrace 507
flushend 507
fncychap 508
fnpos 509
fontenc 77, 509
fontspec 76, 509
footmisc 509
footnote 511
footnotehyper 512
footnpag 512
framed 512
ftnright 515
fullpage 515
fullwidth 515
fwlw 516
geometry 149, 516
gettitlestring 150
glossaries 100, 517
graphics 106, 518
graphicx 106, 530
grffile 107, 530
hang 530
hanging 532
hyperref 98, 361, 533
hyperxmp 537
idxlayout 537
ifoddpage 538
ifplatform 136
indentfirst 539
inputenc 77, 539
keyfloat 113, 539
kvoptions 138
layout 541
letltxmacro 136
letterspace 541

lettrine 542
lips 543
listings 543
lmodern 76
longtable 111, 547
lscape 549
ltcaption 549
ltxtable 549
luatodonotes 116, 550
lwarp 78
lwarp-patch-komascript 689
lwarp-patch-memoir 691
makeidx 101, 151
marginfit 552
marginfix 552
marginnote 553
mcaption 553
mdframed 97, 554
memhfixc 564
metalogo 565
microtype 77, 149, 565
midfloat 566
midpage 566
moreverb 567
morewrites 568
mparhack 569
multicol 569
multirow 570
nameref 573
natbib 573
needspace 574
newclude 115, 575
newfloat 152
newtxmath 106
newunicodechar 77, 575
nextpage 575
nicefrac 105
nonumonpart 576
nopageno 576
nowidow 576
ntheorem 105, 577
overpic 108, 589
pagenote 100, 590
pagesel 590
paralist 590
parskip 591
pdflscape 591
pdfsync 591
pfnote 592
placeins 592
prelim2e 592
prettyref 593
preview 593
quotchap 593
ragged2e 595
realscripts 595
refcount 152
relsize 92, 596
romanbar 598
romanbarpagenumber 598
rotating 598
rotfloat 599
savetrees 600
scalefnt 600
scrextend 600
scrhack 603
scrlayer 604
scrlayer-notecolumn 605
scrlayer-scrpage 606
section 607
sectsty 608
setspace 608
shadow 610
showidx 610
showkeys 610
sidecap 611
sidenotes 611
siunitx 105, 412, 613
soul 614
soulpos 616
soulutf8 617
stabular 617
subcaption 112
subfig 113, 618
subfigure 624
supertabular 112, 624
syntonly 626
tabls 626
tabularx 627
tabulary 627
textarea 628
textcomp 77, 92, 628
textfit 632
textpos 633
theorem 633
threeparttable 638
tikz 639
titleps 640
titleref 643
titlesec 643

titletoc 645
titling 100, 647
tocbasic 652
tocbibind 101, 102, 652, 653
tocloft 100, 655
tocstyle 661
todo 662
todonotes 116, 663
transparent 664
trivfloat 112, 665
turnthepage 666
typearea 666
ulem 667
units 105
upref 669
url 98
varioref 98
verse 115, 669, 670
wallpaper 671
watermark 672
wrapfig 672
xcolor 107, 414, 674
xfrac 683
xifthen 151
xltxtra 685
xmpincl 686
xparse 116, 150
xpatch 136
xstring 151
xtab 112, 686
zref 152
zwpagelayout 688
page
inacessible 81, 82
\pagebreak 8262
\pagecolor 70
pagenote (package) 100, 590
\pagenumbering 3587
\pageref 6455
\pagerefPageFor 6454
pagesel (package) 590
\pagestyle 3579
Pandoc (program) 53
\par
hooks 244
\paragraph 4585
paralist (package) 590
\parbox 7895
\parsemulticolumnalignment 5892
parskip (package) 591
\part 4516
PDF images 106, 413
pdfcrop (program) [requirement] 56
pdfLaTeX (program) [requirement] . . . 56
pdflscape (package) 591
pdfseparate (program) [requirement]
. . . . . 56, 61
pdfsync (package) 591
pdftocairo (program) 106, 413
pdftocairo (program) [requirement] 56, 61
pdftotext (program) [requirement] 56, 61
Perl 62
perl (program) [requirement] 62
pfnote (package) 592
\phantomsection 8322
picture (environment) 420, 7791
placeins (package) 592
Plastex (program) 52
\popclose 3407
Poppler 56, 61
prelim2e (package) 592
prettyref (package) 593
preview (package) 593
\printauthor 280, 4804, 4823
\printdate 280, 4815, 4825
printglossary (option) [lwarpmk] 101, 517
\printindex 6836
\printthanks 279, 282
\printtitle 280, 4788, 4822
problems 128
programs
utility 55
programs:
[requirement]:
LuaLaTeX 56
MathJax 56
pdfcrop 56
pdfLaTeX 56
pdfseparate 56, 61
pdftocairo 56, 61
pdftotext 56, 61
perl 62
XeLaTeX 56
Adobe 53
AsciiDoc 53
AsciiDoctor 53
Asciidoctor-LaTeX 53
Flare 53
FrameMaker 53
GELLMU 52

GladTeX 52
Hevea 52
htlatex 52
InDesign 53
LaTeX2HTML 52
LaTeXML 52
LibreOffice 53
Linux 86, 137
lwarpmk 126, 208
Mac OS 86, 137
Madcap 53
MathJax 103, 104, 385
MS-Windows 86, 137
OpenOffice 53
Pandoc 53
pdftocairo 106, 413
Plastex 52
TeX2page 52
TeX4ht 52
TeXMaths 122
TtH 52
Unix 86, 137
Windows 86, 137
Word 53
xindy 90
project.css (file) 86
project.lwarpmkconf (file) 167
project_html.tex (file) 166
published 287
\pushclose 3380

Q

\qquad 439, 8218
\quad 439, 8213
quotchap (package) 593
quote (environment) 4963

R

ragged2e (package) 595
\raggedbottom 3583
\raggedleft 276
\raggedright 276
\raisebox 7980
realscripts (package) 595
\ref 6448
refcount (package) 152
\reflectbox 299
relsize (package) 92, 596
[requirement]:
LuaLaTeX (program) 56
MathJax (program) 56
pdfcrop (program) 56
pdfLaTeX (program) 56
pdfseparate (program) 56, 61
pdftocairo (program) 56, 61
pdftotext (program) 56, 61
perl (program) 62
XeLaTeX (program) 56
\RequirePackage 474
\resizebox 304
\ResumeTabular 6086
\reversemarginpar 4173
\rmfamily 8124
romanbar (package) 598
romanbarpagenumber (package) 598
\rotatebox 265
rotating (package) 598
rotfloat (package) 599
\rule 8319

S

sample_project.css (file) 86, 205
savetrees (package) 600
\sb 8132
\scalebox 284
scalefnt (package) 600
scrextend (package) 600
scrhack (package) 603
scrlayer (package) 604
scrlayer-notecolumn (package) 605
scrlayer-scrpage (package) 606
\scshape 8129
section
depths 132
heading, word processor 122
missing 81, 82
\section 4555
section (package) 607
sectsty (package) 608
\SetHTMLFileNumber 3473
setspace (package) 608
settings
css project-specific 86
CSS selection 85
HTML conversion 80
lwarp package options 78
selecting output 87
title page 88
\sffamily 8125
\sfrac 683

shadow (package) *610*
showidx (package) *610*
showkeys (package) *610*
sidecap (package) *611*
sidenotes (package) *611*
SideTOCDepth (counter) *81*, *375*
\sidetocname 6756
\simplechapterdelim 4371
siunitx
  with TeXMaths 122
siunitx (package) *105*, *412*, *613*
\sloppy 3585
soul (package) *614*
soulpos (package) *616*
soulutf8 (package) *617*
\sp 8131
space
  horizontal 91
    between minipages 439
split
  miss-numbered 105, 577
stabular (package) *617*
stack depths 132
\StartDefiningTabulars 5348
subcaption (package) *112*
subequations
  MathJax 104, 385
subfig (package) *113*, *618*
subfigure (package) *624*
\subparagraph 4593
\subsection 4568
\subsubsection 4576
subtitle 287
supertabular (package) *112*, *624*
SVG
  images showing as HTML 69, 130
  math summary 103, 384
  mathsvg option 139
syntonly (package) *626*

T

tabbing (environment) 5031
\tableofcontents *84*, 6701
tabls (package) *626*
tabular
  baseline 319
  HTML columnn conversion 320
  macros inside 110, 304
  misplaced \noalign 111, 304
  misplaced alignment tab character & 108, 302
  multicolumn with multirow 572
  row corruption 110, 304
\TabularMacro 6084
tabularx (package) *627*
tabulary (package) *627*
tagged equations
  Mathjax 104, 385
\TeX 8327
TeX2page (program) *52*
TeX4ht (program) *52*
TeXMaths (program) *122*
textarea (package) *628*
\textbf 8029
\textcolor 61
textcomp (package) *77*, *92*, *628*
textfit (package) *632*
\textgreater 3463
\textit 8054
\textless 3463
\textmd 8023
\textnormal 8071
textpos (package) *633*
\textrm 8032
\textsc 8057
\textsf 8039
\textsl 8064
\textsubscript 8135
\textsuperscript 8133
\texttt 8044
\textup 8047
\tfl@chapter@fix *665*
\thanks *89*
\thanksmarkseries 84
\theauthor *280*
thebibliography (environment) . . . 6890
\thedate *280*
theindex (environment) 6843
theorem (package) *633*
\thetitle *280*
\thispagestyle 3580
threeparttable (package) *638*
tikz
  catcodes 639
  dollar redefined 639
tikz (package) *639*
tikzpicture (environment) 7
Title
  HTML meta tag 248

title
HTML meta tag 90, 248
\title 89, 4025
titlepage
subtitle and published 287
titlepage (environment) 88, 4772
titleps (package) 640
titleref (package) 643
titles (option) [tocloft] 100
titlesec (package) 643
titletoc (package) 645
titling (package) 100, 647
titlingpage (environment) 14, 89
tocbasic (package) 652
[tocbibind]:
numindex (option) 102, 653
tocbibind (package) . . . 101, 102, 652, 653
tocdepth (counter) 81
[tocloft]:
titles (option) 100
tocloft (package) 100, 655
tocstyle (package) 661
todo (package) 662
todonotes (package) 116, 663
\tracinglwarp 130, 588
transparent (package) 664
trivfloat
with newfloat, algorithmic 666
trivfloat (package) 112, 665
troubleshooting 128
HTML debug comments 159
\ttfamily 8126
TtH (program) 52
turnthepage (package) 666
tutorial.tex (file) 63
typearea (package) 666

U

ulem (package) 667
\underline 8143
underscore
filename 81, 92
Unicode
enhanced coverage 76
file & section names 261
input characters 135
selection 76
units (package) 105
Unix (program) 86, 137
\up 8141
upref (package) 669
\upshape 8127
\url 6514
url (package) 98
usingOSWindows (boolean) 137
UTF-8
enhanced coverage 76
file & section names 261
index 77
selection 76
utility
programs 55

V

varioref (package) 98
verbatim (environment) 5021
\VerbatimHTMLWidth (length) 291
\verbatiminput 5013
verse (environment) 2
verse (package) 115, 669, 670
viewport
HTML meta tag 273
vleftmargini (length) 115, 290, 670
vleftskip (length) 115, 290, 670, 671
\vspace 8259

W

wallpaper (package) 671
warning
Improper \prevdepth. 128
Label(s) may have changed. 128
Leaders not followed by proper glue. 128
Missing $ inserted. 128
Options clash, memoir 114
warning icon 134
warpall (environment) 87, 107
warpHTML (environment) . . 57, 84, 87, 108
warpHTML (option) 78, 138
\warpHTMLonly 84, 88, 97
warpingHTML (boolean) 138
warpingprint (boolean) 138
warpprint (environment) 84, 87, 108
warpprint (option) 78, 138
\warpprintonly 84, 88, 96
watermark (package) 672
Windows (program) 86, 137
Word (program) 53
word processor
conversion recommendations 122
HTML conversion settings 119, 159, 160

section headings 122
WPMarkFloats (boolean) *121*, *159*
WPMarkLOFT (boolean) *122*, *160*
WPMarkMath (boolean) *122*, *161*
WPMarkMinipages (boolean) *121*, *160*
WPMarkTOC (boolean) *121*, *160*
WPTitleHeading (boolean) *122*, *161*
wrapfig (package) *672*

X

xcolor (package) *107*, *414*, *674*
xdyFilename (option) *79*, *90*, *139*
XeLaTeX
detection 148
file & section names 261
\XeLaTeX 8359
XeLaTeX (program) [requirement] *56*
\XeTeX 8359
xfrac (package) *683*
\xfracHTMLfontsize 3
xifthen (package) *151*
xindy
customizing 90
xindy (program) *90*
xltxtra (package) *685*
xmpincl (package) *686*
xparse
warnings 116
xparse (package) *116*, *150*
xpatch (package) *136*
xstring (package) *151*
xtab (package) *112*, *686*

Z

zref (package) *152*
zwpagelayout (package) *688*