



The **lwarp** package

ℒ_AT_EX to HTML

v0.38 — 2017/08/27

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

The `lwarp` package allows ℒ_AT_EX to directly produce HTML 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 pdfℒ_AT_EX, Luaℒ_AT_EX, or Xeℒ_AT_EX. 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 a hundred ℒ_AT_EX 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 1, Updates.](#)

Note that this is still an “alpha” 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.

Contents

| | | |
|----------|--|-----------|
| 1 | lwarp.sty | 26 |
| 1 | Updates | 26 |
| 2 | Introduction | 31 |
| 2.1 | Supported packages and features | 32 |
| 3 | Alternatives | 36 |
| 3.1 | Internet class | 36 |
| 3.2 | TeX4ht | 36 |
| 3.3 | Translators | 36 |
| 3.4 | AsciiDoc | 37 |
| 3.5 | Pandoc | 37 |
| 3.6 | Word processors | 37 |
| 3.7 | Commercial systems | 38 |
| 3.8 | Comparisons | 38 |
| 4 | Installation | 39 |
| 4.1 | Installing the lwarp package | 39 |
| 4.2 | Installing the lwarpmk utility | 42 |
| 4.2.1 | Using a local copy of lwarpmk | 43 |
| 4.3 | Installing additional utilities | 44 |
| 5 | Tutorial | 46 |
| 5.1 | Starting a new project | 46 |
| 5.2 | Compiling the print version with lwarpmk | 50 |
| 5.3 | Compiling the HTML version with lwarpmk | 51 |
| 5.4 | Generating the SVG images | 52 |
| 5.5 | Using MathJax for math | 53 |
| 5.6 | Changing the CSS style | 54 |
| 5.7 | Customizing the HTML output | 55 |
| 5.8 | Using latexmk | 60 |
| 5.9 | Using XeLaTeX or LuaLaTeX | 61 |
| 5.10 | Using a glossary | 62 |
| 5.11 | Cleaning auxiliary files | 63 |
| 5.12 | Cleaning auxiliary and output files | 63 |
| 5.13 | Processing multiple projects in the same directory | 63 |
| 5.14 | Using the make utility | 63 |
| 6 | Additional details | 64 |
| 6.1 | Font and UTF-8 support | 64 |
| 6.1.1 | Indexes and UTF-8 | 65 |
| 6.2 | lwarp package loading and options | 65 |
| 6.3 | Selecting the operating system | 66 |

| | | |
|----------|--|-----------|
| 6.4 | Selecting actions for print or HTML output | 67 |
| 6.5 | Commands to be placed into the warpprint environment | 68 |
| 6.6 | Title page | 68 |
| 6.7 | HTML page meta descriptions | 69 |
| 6.8 | HTML page meta author | 70 |
| 6.9 | CSS | 70 |
| 6.10 | Modifying xindy index processing | 70 |
| 7 | Special cases and limitations | 71 |
| 7.1 | Formatting | 71 |
| 7.1.1 | Text formatting | 71 |
| 7.1.2 | Horizontal space | 71 |
| 7.1.3 | Text alignment | 72 |
| 7.1.4 | Accents | 72 |
| 7.1.5 | Textcomp | 72 |
| 7.1.6 | Superscripts and other non-math uses of math mode | 72 |
| 7.1.7 | Empty \item followed by a new line of text or a nested list: | 72 |
| 7.1.8 | Filenames in lists | 72 |
| 7.1.9 | resize package | 72 |
| 7.2 | Boxes and minipages | 73 |
| 7.2.1 | Save Boxes | 73 |
| 7.2.2 | Minipages | 73 |
| 7.2.3 | Side-by-side minipages | 74 |
| 7.2.4 | Framed minipages and other environments | 74 |
| 7.2.5 | fancybox package | 75 |
| 7.2.6 | mdframed package | 76 |
| 7.3 | Cross-references | 77 |
| 7.3.1 | Page references | 77 |
| 7.3.2 | cleveref and varioref packages | 77 |
| 7.3.3 | Hyperlinks, hyperref, and url | 78 |
| 7.3.4 | Footnotes and page notes | 78 |
| 7.4 | Front and back matter | 78 |
| 7.4.1 | Starred chapters and sections | 78 |
| 7.4.2 | abstract package | 79 |
| 7.4.3 | appendix package | 79 |
| 7.4.4 | pagenote package | 79 |
| 7.4.5 | endnotes package | 79 |
| 7.4.6 | glossaries package | 79 |
| 7.4.7 | Index and the tocbibind package | 80 |
| 7.5 | Math | 81 |
| 7.5.1 | ntheorem package | 83 |
| 7.5.2 | siunitx package | 83 |
| 7.5.3 | units and nicefrac packages | 84 |
| 7.5.4 | newtxmath package | 84 |
| 7.6 | Graphics | 84 |
| 7.6.1 | grffile package | 85 |

| | | |
|-----------|---|------------|
| 7.6.2 | color package | 85 |
| 7.6.3 | xcolor package | 85 |
| 7.7 | Tabular | 86 |
| 7.7.1 | longtable package | 87 |
| 7.7.2 | float, trivfloat, and/or algorithmicx together | 87 |
| 7.7.3 | caption and subcaption packages | 88 |
| 7.8 | Floats | 88 |
| 7.8.1 | subfig package | 88 |
| 7.8.2 | floatrow package | 88 |
| 7.9 | Miscellaneous | 89 |
| 7.9.1 | verse and memoir | 89 |
| 7.9.2 | newclude package | 89 |
| 7.9.3 | babel package | 90 |
| 8 | EPUB conversion | 91 |
| 9 | Word-processor conversion | 93 |
| 10 | Modifying lwarp | 95 |
| 10.1 | Creating an lwarp version of a package | 95 |
| 10.2 | Adding a package to the lwarp.dtx file | 96 |
| 10.3 | Testing lwarp | 96 |
| 10.4 | Modifying lwarpmk | 96 |
| 11 | Troubleshooting | 98 |
| 11.1 | Using the lwarp.sty package | 98 |
| 11.1.1 | Debug tracing output | 100 |
| 11.2 | Compiling the lwarp.dtx file | 100 |
| 12 | Implementation | 101 |
| 13 | Stack depths | 101 |
| 14 | Source Code | 103 |
| 15 | Detecting the T_EX Engine — pdf\LaTeX, lua\LaTeX, x\LaTeX | 104 |
| 16 | Unicode Input Characters | 104 |
| 17 | Early package requirements | 105 |
| 18 | Operating-System portability | 106 |
| 18.1 | Common portability code | 106 |
| 18.2 | Unix, Linux, and Mac OS | 106 |
| 18.3 | MS-Windows | 106 |
| 19 | Package options | 107 |
| 19.1 | Conditional compilation | 110 |

| | | |
|-----------|---|------------|
| 20 | Misplaced packages | 111 |
| 21 | Required packages | 115 |
| 22 | Loading packages | 120 |
| 23 | File handles | 122 |
| 24 | Include a file | 123 |
| 25 | Copying a file | 124 |
| 26 | Debugging messages | 125 |
| 27 | HTML-conversion output modifications | 125 |
| 28 | Remembering original formatting macros | 126 |
| 29 | Textcomp | 128 |
| 30 | Accents | 128 |
| 31 | Configuration Files | 130 |
| 31.1 | project_html.tex | 130 |
| 31.2 | lwarpmk.conf | 130 |
| 31.3 | project.lwarpmkconf | 131 |
| 31.4 | lwarp.css | 132 |
| 31.5 | lwarp_sagebrush.css | 158 |
| 31.6 | lwarp_formal.css | 163 |
| 31.7 | sample_project.css | 167 |
| 31.8 | lwarp.xdy | 168 |
| 31.9 | lwarp_mathjax.txt | 168 |
| 31.10 | lwarpmk option | 170 |
| 32 | Stacks | 181 |
| 32.1 | Assigning depths | 182 |
| 32.2 | Closing actions | 182 |
| 32.3 | Closing depths | 183 |
| 32.4 | Pushing and popping the stack | 183 |
| 33 | Data arrays | 185 |
| 34 | HTML entities | 185 |
| 35 | HTML filename generation | 186 |
| 36 | Homepage link | 189 |

| | | |
|-----------|--|------------|
| 37 | \PrintStack diagnostic tool | 190 |
| 38 | Closing stack levels | 190 |
| 39 | PDF pages and styles | 191 |
| 40 | HTML tags, spans, divs, elements | 192 |
| 40.1 | Mapping \LaTeX Sections to HTML Sections | 192 |
| 40.2 | Babel-French | 193 |
| 40.3 | HTML tags | 194 |
| 40.4 | Block tags and comments | 196 |
| 40.5 | Div class and element class | 197 |
| 40.6 | Single-line elements | 198 |
| 40.7 | HTML5 semantic elements | 199 |
| 40.8 | High-level block and inline classes | 199 |
| 40.9 | Closing HTML tags | 200 |
| 41 | Paragraph handling | 201 |
| 42 | Paragraph start/stop handling | 204 |
| 43 | Page headers and footers | 207 |
| 44 | CSS | 208 |
| 45 | HTML meta description and author | 209 |
| 46 | Footnotes | 209 |
| 46.1 | Regular page footnotes | 210 |
| 46.2 | Minipage footnotes | 210 |
| 46.3 | Titlepage thanks | 210 |
| 46.4 | Regular page footnote implementation | 210 |
| 46.5 | Minipage footnote implementation | 212 |
| 47 | Marginpars | 213 |
| 48 | Splitting HTML files | 213 |
| 49 | Sectioning | 219 |
| 49.1 | User-level starred section commands | 219 |
| 49.2 | Book class commands | 220 |
| 49.3 | Sectioning support macros | 220 |
| 49.4 | \section and friends | 226 |
| 50 | Starting a new file | 228 |
| 51 | Starting HTML output | 231 |

| | | |
|-----------|---|------------|
| 52 | Ending HTML output | 234 |
| 53 | Titles and the titling package | 235 |
| 53.1 | Setting the title, etc. | 237 |
| 53.2 | Changes to HTML titlepage and titlingpage | 239 |
| 53.3 | Printing the title, etc. in HTML | 240 |
| 53.4 | Printing the title, etc. in print form | 243 |
| 53.5 | \maketitle for print output | 244 |
| 53.6 | \maketitle for HTML output | 246 |
| 54 | Abstract | 249 |
| 55 | Quote and verse | 250 |
| 55.1 | Citations and attributions | 250 |
| 55.2 | Quotes, quotations | 251 |
| 55.3 | Verse | 251 |
| 56 | Verbatim | 252 |
| 57 | Theorems | 254 |
| 58 | Lists | 255 |
| 58.1 | Itemize | 256 |
| 58.2 | Enumerate | 257 |
| 58.3 | Description | 257 |
| 59 | Tabular | 259 |
| 59.1 | Token lookahead | 260 |
| 59.2 | Booleans | 261 |
| 59.3 | Handling &, @, and ! | 262 |
| 59.3.1 | Localizing & catcodes | 263 |
| 59.3.2 | Handling & | 263 |
| 59.4 | Handling \\ | 264 |
| 59.5 | Variables | 265 |
| 59.6 | Parsing @, >, <, ! columns | 266 |
| 59.7 | Parsing 'l', 'c', or 'r' columns | 269 |
| 59.8 | Parsing 'p', 'm', or 'b' columns | 269 |
| 59.9 | Parsing 'D' columns | 270 |
| 59.10 | Parsing the column specifications | 270 |
| 59.11 | Starting a new row | 274 |
| 59.12 | Printing at or bang tags | 275 |
| 59.13 | Data opening tag | 275 |
| 59.14 | Midrules | 276 |
| 59.15 | Multicolumns | 277 |
| 59.15.1 | Parsing multicolumns | 277 |
| 59.15.2 | High-level multicolumn interface | 280 |
| 59.15.3 | Longtable captions | 281 |

| | | |
|-----------|--|------------|
| 59.15.4 | Counting HTML tabular columns | 284 |
| 59.15.5 | <code>\tabledatamulticolumntag</code> | 285 |
| 59.16 | <code>Multirow</code> | 286 |
| 59.17 | Utility macros inside a table | 287 |
| 59.18 | Checking for a new table cell | 287 |
| 59.19 | <code>\mrowcell</code> | 288 |
| 59.20 | New <code>\tabular</code> definition | 289 |
| 59.21 | <code>Array</code> | 292 |
| 60 | Cross-references | 292 |
| 60.1 | Setup | 292 |
| 60.2 | Zref setup | 294 |
| 60.3 | Labels | 295 |
| 60.4 | References | 297 |
| 60.5 | Hyper-references | 299 |
| 61 | Floats | 302 |
| 61.1 | Float captions | 303 |
| 61.1.1 | Caption inside a float environment | 304 |
| 61.1.2 | Caption and LOF linking and tracking | 305 |
| 62 | Table of Contents, LOF, LOT | 308 |
| 62.1 | Reading and printing the TOC | 308 |
| 62.2 | High-level TOC commands | 310 |
| 62.3 | Side TOC | 311 |
| 62.4 | Low-level TOC line formatting | 312 |
| 63 | Index and glossary | 315 |
| 64 | Math | 317 |
| 64.1 | Inline and display math | 319 |
| 64.2 | MathJax support | 321 |
| 64.3 | Equation environment | 325 |
| 64.4 | AMS Math environments | 327 |
| 64.4.1 | Support macros | 327 |
| 64.4.2 | Environment patches | 328 |
| 65 | Lateximages | 335 |
| 66 | <code>center</code> , <code>flushleft</code> , <code>flushright</code> | 341 |
| 67 | Siunitx | 342 |
| 68 | Graphics | 343 |
| 68.1 | <code>\graphicspath</code> | 344 |
| 68.2 | Length conversions and graphics options | 344 |
| 68.3 | <code>\includegraphics</code> | 347 |

| | | |
|-----------|---|------------|
| 68.4 | <code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code> | 351 |
| 68.5 | Null functions | 354 |
| 69 | Xcolor defaults | 354 |
| 69.1 | Defaults without xcolor | 354 |
| 69.2 | xcolor print-mode additions | 355 |
| 70 | Cleveref | 358 |
| 71 | Picture | 360 |
| 72 | Boxes and Minipages | 360 |
| 72.1 | Counters and lengths | 361 |
| 72.2 | Footnote handling | 361 |
| 72.3 | Minipage handling | 361 |
| 72.4 | Parbox, makebox, framebox, fbox, raisebox | 365 |
| 73 | Direct formatting | 370 |
| 74 | Skips, spaces, font sizes | 374 |
| 75 | <code>\phantomsection</code> | 382 |
| 76 | <code>\LaTeX</code> and other logos | 382 |
| 76.1 | HTML logos | 382 |
| 76.2 | Print logos | 384 |
| 77 | <code>\AtBeginDocument</code>, <code>\AtEndDocument</code> | 384 |
| 78 | Trademarks | 385 |
| 2 | lwarp-abstract.sty | 386 |
| 79 | Abstract | 386 |
| 3 | lwarp-afterpage.sty | 388 |
| 80 | Afterpage | 388 |
| 4 | lwarp-algorithmicx.sty | 389 |
| 81 | Algorithmicx | 389 |

| | | |
|-----------|----------------------------------|------------|
| 5 | lwarp-alltt.sty | 389 |
| 82 | Alltt | 390 |
| 6 | lwarp-amsthm.sty | 391 |
| 83 | AMSthm | 391 |
| 7 | lwarp-appendix.sty | 394 |
| 84 | Appendix | 394 |
| 8 | lwarp-arabicfront.sty | 395 |
| 85 | Arabicfront | 395 |
| 9 | lwarp-balance.sty | 395 |
| 86 | Balance | 395 |
| 10 | lwarp-bookmark.sty | 396 |
| 87 | Bookmark | 396 |
| 11 | lwarp-booktabs.sty | 396 |
| 88 | Booktabs | 396 |
| 12 | lwarp-boxedminipage.sty | 398 |
| 89 | Boxedminipage | 398 |
| 13 | lwarp-boxedminipage2e.sty | 398 |
| 90 | Boxedminipage2e | 398 |
| 14 | lwarp-caption2.sty | 399 |
| 91 | Caption2 | 399 |

| | | |
|-----------|---------------------------------|------------|
| 15 | lwarp-ccaption.sty | 400 |
| 92 | Ccaption | 400 |
| 16 | lwarp-changepage.sty | 400 |
| 93 | Changepage | 400 |
| 17 | lwarp-chappg.sty | 401 |
| 94 | Chappg | 401 |
| 18 | lwarp-color.sty | 402 |
| 95 | Color | 402 |
| 19 | lwarp-crop.sty | 402 |
| 96 | Crop | 402 |
| 20 | lwarp-cutwin.sty | 403 |
| 97 | Cutwin | 403 |
| 21 | lwarp-dcolumn.sty | 403 |
| 98 | Dcolumn | 404 |
| 22 | lwarp-draftwatermark.sty | 405 |
| 99 | Draftwatermark | 405 |
| 23 | lwarp-ellipsis.sty | 405 |
| 100 | Ellipsis | 405 |
| 24 | lwarp-emptypage.sty | 406 |
| 101 | Emptypage | 406 |

| | | |
|-----------|-----------------------------|------------|
| 25 | lwarp-endnotes.sty | 407 |
| 102 | Endnotes | 407 |
| 26 | lwarp-enumerate.sty | 408 |
| 103 | Enumerate | 408 |
| 27 | lwarp-enumitem.sty | 409 |
| 104 | Enumitem | 409 |
| 28 | lwarp-epigraph.sty | 410 |
| 105 | Epigraph | 410 |
| 29 | lwarp-eso-pic.sty | 411 |
| 106 | Eso-pic | 411 |
| 30 | lwarp-everypage.sty | 412 |
| 107 | Everypage | 412 |
| 31 | lwarp-everyshi.sty | 413 |
| 108 | Everyshi | 413 |
| 32 | lwarp-extramarks.sty | 413 |
| 109 | Extramarks | 413 |
| 33 | lwarp-fancybox.sty | 414 |
| 110 | Fancybox | 414 |
| 34 | lwarp-fancyhdr.sty | 417 |
| 111 | Fancyhdr | 418 |

| | | |
|-----------|----------------------------------|------------|
| 35 | lwarp-fancyvrb.sty | 419 |
| 112 | Fancyvrb | 419 |
| 36 | lwarp-figcaps.sty | 425 |
| 113 | Figcaps | 425 |
| 37 | lwarp-fix2col.sty | 425 |
| 114 | Fix2col | 425 |
| 38 | lwarp-float.sty | 426 |
| 115 | Float and <code>\newfloat</code> | 426 |
| 39 | lwarp-floatflt.sty | 428 |
| 116 | Floatflt | 428 |
| 40 | lwarp-floatpag.sty | 429 |
| 117 | Floatpag | 429 |
| 41 | lwarp-floatrow.sty | 430 |
| 118 | Floatrow | 430 |
| 42 | lwarp-flushend.sty | 436 |
| 119 | Flushend | 436 |
| 43 | lwarp-fncychap.sty | 437 |
| 120 | Fncychap | 437 |
| 44 | lwarp-fontenc.sty | 438 |
| 121 | Fontenc | 438 |

| | | |
|-----------|--------------------------------|------------|
| 45 | lwarp-fontspec.sty | 438 |
| 122 | Fontspec | 438 |
| 46 | lwarp-footmisc.sty | 439 |
| 123 | Footmisc | 439 |
| 47 | lwarp-footnote.sty | 441 |
| 124 | Footnote | 441 |
| 48 | lwarp-footnotehyper.sty | 442 |
| 125 | Footnotehyper | 442 |
| 49 | lwarp-framed.sty | 443 |
| 126 | Framed | 443 |
| 50 | lwarp-ftnright.sty | 446 |
| 127 | Ftnright | 446 |
| 51 | lwarp-fullpage.sty | 446 |
| 128 | Fullpage | 446 |
| 52 | lwarp-geometry.sty | 447 |
| 129 | Geometry | 447 |
| 53 | lwarp-glossaries.sty | 448 |
| 130 | Glossaries | 448 |
| 54 | lwarp-graphics.sty | 450 |
| 131 | Graphics | 450 |

| | | |
|-----------|------------------------------|------------|
| 55 | lwarp-graphicx.sty | 450 |
| 132 | Graphicx | 450 |
| 56 | lwarp-grffile.sty | 451 |
| 133 | Grffile | 451 |
| 57 | lwarp-hyperref.sty | 452 |
| 134 | Hyperref | 452 |
| 58 | lwarp-hyperxmp.sty | 455 |
| 135 | Hyperxmp | 455 |
| 59 | lwarp-idxlayout.sty | 456 |
| 136 | Idxlayout | 456 |
| 60 | lwarp-indentfirst.sty | 457 |
| 137 | Indentfirst | 457 |
| 61 | lwarp-inputenc.sty | 457 |
| 138 | Inputenc | 457 |
| 62 | lwarp-keyfloat.sty | 458 |
| 139 | Keyfloat | 458 |
| 63 | lwarp-layout.sty | 460 |
| 140 | Layout | 460 |

| | | |
|-----------|------------------------------|------------|
| 64 | lwarp-letterspace.sty | 460 |
| 141 | Letterspace | 460 |
| 65 | lwarp-lettrine.sty | 461 |
| 142 | Lettrine | 461 |
| 66 | lwarp-lips.sty | 462 |
| 143 | Lips | 462 |
| 67 | lwarp-listings.sty | 463 |
| 144 | Listings | 463 |
| 68 | lwarp-longtable.sty | 467 |
| 145 | Longtable | 467 |
| 69 | lwarp-lscape.sty | 469 |
| 146 | Lscape | 469 |
| 70 | lwarp-ltcaption.sty | 469 |
| 147 | Ltcaption | 469 |
| 71 | lwarp-marginfit.sty | 470 |
| 148 | Marginfit | 470 |
| 72 | lwarp-marginfix.sty | 470 |
| 149 | Marginfix | 470 |

| | | |
|-----------|-----------------------------|------------|
| 73 | lwarp-marginnote.sty | 471 |
| 150 | Marginnote | 471 |
| 74 | lwarp-mcaption.sty | 471 |
| 151 | Mcaption | 471 |
| 75 | lwarp-mdframed.sty | 472 |
| 152 | Mdframed | 472 |
| 76 | lwarp-metalogo.sty | 481 |
| 153 | Metalogo | 481 |
| 77 | lwarp-microtype.sty | 482 |
| 154 | Microtype | 482 |
| 78 | lwarp-moreverb.sty | 483 |
| 155 | Moreverb | 483 |
| 79 | lwarp-mparhack.sty | 485 |
| 156 | Mparhack | 485 |
| 80 | lwarp-multicol.sty | 486 |
| 157 | Multicol | 486 |
| 81 | lwarp-multirow.sty | 488 |
| 158 | Multirow | 488 |
| 82 | lwarp-nameref.sty | 488 |
| 159 | Nameref | 488 |

| | | |
|-----------|---------------------------------|------------|
| 83 | lwarp-needspace.sty | 489 |
| 160 | Needspace | 489 |
| 84 | lwarp-newclude.sty | 489 |
| 161 | Newclude | 489 |
| 85 | lwarp-newunicodechar.sty | 490 |
| 162 | Newunicodechar | 490 |
| 86 | lwarp-nextpage.sty | 490 |
| 163 | Nextpage | 490 |
| 87 | lwarp-nonumonpart.sty | 491 |
| 164 | Nonumonpart | 491 |
| 88 | lwarp-nopageno.sty | 491 |
| 165 | Nopageno | 491 |
| 89 | lwarp-nowidow.sty | 492 |
| 166 | Nowidow | 492 |
| 90 | lwarp-ntheorem.sty | 493 |
| 167 | Ntheorem | 493 |
| 91 | lwarp-pagenote.sty | 504 |
| 168 | Pagenote | 504 |
| 92 | lwarp-paralist.sty | 504 |
| 169 | Paralist | 504 |

| | | |
|------------|-------------------------------------|------------|
| 93 | lwarp-parskip.sty | 505 |
| 170 | Parskip | 505 |
| 94 | lwarp-pdfscape.sty | 505 |
| 171 | Pdfscape | 505 |
| 95 | lwarp-pdfsync.sty | 506 |
| 172 | Pdfsync | 506 |
| 96 | lwarp-placeins.sty | 506 |
| 173 | Placeins | 506 |
| 97 | lwarp-prelim2e.sty | 507 |
| 174 | Prelim2e | 507 |
| 98 | lwarp-ragged2e.sty | 508 |
| 175 | Ragged2e | 508 |
| 99 | lwarp-realscripts.sty | 509 |
| 176 | Realscripts | 509 |
| 100 | lwarp-reysize.sty | 510 |
| 177 | Reysize | 510 |
| 101 | lwarp-romanbar.sty | 512 |
| 178 | Romanbar | 512 |
| 102 | lwarp-romanbarpagenumber.sty | 512 |
| 179 | Romanbarpagenumber | 512 |

| | | |
|------------|----------------------------|------------|
| 103 | lwarp-rotating.sty | 513 |
| 180 | Rotating | 513 |
| 104 | lwarp-rotfloat.sty | 514 |
| 181 | Rotfloat | 514 |
| 105 | lwarp-savetrees.sty | 515 |
| 182 | Savetrees | 515 |
| 106 | lwarp-scalefnt.sty | 515 |
| 183 | Scalefnt | 515 |
| 107 | lwarp-setspace.sty | 516 |
| 184 | Setspace | 516 |
| 108 | lwarp-shadow.sty | 517 |
| 185 | Shadow | 517 |
| 109 | lwarp-showidx.sty | 518 |
| 186 | Showidx | 518 |
| 110 | lwarp-showkeys.sty | 518 |
| 187 | Showkeys | 518 |
| 111 | lwarp-sidecap.sty | 519 |
| 188 | Sidecap | 519 |
| 112 | lwarp-sidenotes.sty | 520 |
| 189 | Sidenotes | 520 |

| | | |
|------------|---------------------------------|------------|
| 113 | lwarp-siunitx.sty | 522 |
| 190 | Siunitx | 522 |
| 114 | lwarp-soul.sty | 523 |
| 191 | Soul | 523 |
| 115 | lwarp-subfig.sty | 525 |
| 192 | Subfig | 525 |
| 116 | lwarp-syntonly.sty | 531 |
| 193 | Syntonly | 531 |
| 117 | lwarp-tabularx.sty | 532 |
| 194 | Tabularx | 532 |
| 118 | lwarp-tabulary.sty | 533 |
| 195 | Tabulary | 533 |
| 119 | lwarp-textcomp.sty | 534 |
| 196 | Textcomp | 534 |
| 120 | lwarp-textpos.sty | 536 |
| 197 | Textpos | 536 |
| 121 | lwarp-theorem.sty | 537 |
| 198 | Theorem | 537 |
| 122 | lwarp-threeparttable.sty | 541 |
| 199 | Threeparttable | 541 |

| | | |
|------------|------------------------------|------------|
| 123 | lwarp-tikz.sty | 542 |
| 200 | Tikz | 542 |
| 124 | lwarp-titleps.sty | 544 |
| 201 | Titleps | 544 |
| 125 | lwarp-titleref.sty | 545 |
| 202 | Titleref | 545 |
| 126 | lwarp-titlesec.sty | 546 |
| 203 | Titlesec | 546 |
| 127 | lwarp-titletoc.sty | 548 |
| 204 | Titletoc | 548 |
| 128 | lwarp-titling.sty | 549 |
| 205 | Titling | 549 |
| 129 | lwarp-tocbibind.sty | 550 |
| 206 | Tocbibind | 550 |
| 130 | lwarp-tocloft.sty | 551 |
| 207 | Tocloft | 551 |
| 131 | lwarp-transparent.sty | 556 |
| 208 | Transparent | 556 |
| 132 | lwarp-trivfloat.sty | 557 |
| 209 | Trivfloat | 557 |

| | | |
|------------|--|------------|
| 209.1 | Combining <code>\newfloat</code> , <code>\trivfloat</code> , and <code>algorithmicx</code> | 557 |
| 133 | lwarp-ulem.sty | 559 |
| 210 | Ulem | 559 |
| 134 | lwarp-upref.sty | 561 |
| 211 | Upref | 561 |
| 135 | lwarp-verse.sty | 562 |
| 212 | Verse | 562 |
| 136 | lwarp-wallpaper.sty | 564 |
| 213 | Wallpaper | 564 |
| 137 | lwarp-wrapfig.sty | 565 |
| 214 | Wrapfig | 565 |
| 138 | lwarp-xcolor.sty | 567 |
| 215 | Xcolor | 567 |
| 215.1 | Xcolor definitions: location and timing | 567 |
| 215.2 | Code | 569 |
| 139 | lwarp-xfrac.sty | 576 |
| 216 | Xfrac | 576 |
| 140 | lwarp-xltxtra.sty | 579 |
| 217 | Xltxtra | 579 |
| 141 | lwarp-xmpinl.sty | 580 |
| 218 | Xmpinl | 580 |

Change History and Index

581

List of Figures

| | | |
|---|--------------------------------|----|
| 1 | tutorial.tex listing | 47 |
|---|--------------------------------|----|

List of Tables

| | | |
|---|---|-----|
| 1 | \LaTeX -HTML generation — lwarp package — Supported functions . . | 32 |
| 2 | Required software programs | 40 |
| 3 | Files created along with the print version | 49 |
| 4 | Package options | 66 |
| 5 | Section depths and HTML headings | 102 |
| 6 | Tabular baseline | 271 |
| 7 | Tabular HTML column conversions | 271 |
| 8 | Cross-referencing data structures | 293 |
| 9 | Float data structures | 302 |

Package 1

lwarp.sty

1 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 [581](#).

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 [7.4.1](#).

updated tutorial

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

packages

- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `relsize`, `scalegnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.
- Added `grffile`, `romanbar`.
- Added `arabicfront`, `chappg`, `nonumonpart`, `nopageno`, `romanbarpagenumber`.

front & back matter

- Docs: Improved description of the use of front/back matter. See section [7.4](#).
- Fix: color requests `xcolor`.
- Fix: `\part` for `article` class.

v0.37:

`\include for HTML`

`latexmk`

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

`babel-french`

- Improved support for \TeX accents, `textcomp`, `siunitx` symbols.
- Improved `babel-french` handling for load order and `~` tilde.

v0.36:

- Recorganized the documentation section regarding special cases and limitations. (Section 7)
- 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](#)
 - 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` formatting. See section 7.2.2. `\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 7.2.4. `\fbox` and minipages now often work in `SVG` math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- [framing minipages](#)

lateximage, svg math,
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- Improved compatibility between lateximage and minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox, tabular, booktabs.
- Improved font control for lateximagees and svg math.
- Added the eqnarray environments.
- fancyvrb is no longer required (preloaded), but is still supported.
- Added verbatim and moreverb.
- Added fancybox, boxedminipage2e and shadow.
- enumitem is no longer required, but is still supported.
- Added enumerate and paralist.
- titles is no longer required, but is still supported.
- Added crop.
- Added rotfloat, marginfit, and several minor packages; see the change log.
- 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 59.3.1. 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: **lwrap-newproject**

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

To modify existing documents, remove from the document source:

```
\usepackage{lwrap-newproject}
```

The lwrap package now produces the configuration files during print output, and also accepts the option lwrapmk 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 |
|----------------------|------------------------------|
| lwrapmklang (new) | IndexLanguage 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 lwrap, such as:

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

- The file lwrap_html.xdy has been renamed lwrap.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 lwrap.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 6.10.
- 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`.
- Boolean `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.
- 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.
- 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 `printglossary` and `htmlglossary` actions.

2 Introduction

The lwarp project aims to allow a rich \LaTeX document to be converted to a reasonable HTML interpretation. 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 HTML, CSS, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

`pdflatex`, `xelatex`, or `lualatex` is 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.

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments, as this format has better browser and e-book support than MathML (as of this writing), while still allowing for the high-quality display and printing of images (again, subject to potentially bug-ridden¹ 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.² 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, one per math expression. lwarp maintains \LaTeX control for cross-referencing and equation numbering / formatting.

The lwarp package allows \LaTeX to directly generate HTML tags from a \LaTeX source document, with only minor intervention on the user’s part. 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 HTML tags, to a number of HTML plain-text files and accompanying images.

¹Firefox has had an on-again/off-again bug for quite some time regarding printing svgs at high resolution.

²There 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!

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 usual \TeX version. These `lwarp-packagename.sty` files contain code used to emulate or replace functions for HTML output.

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 32.)

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

2.1 Supported packages and features

Supported classes include book, report, and article. memoir is planned, but in the meantime many of the packages used by memoir are already supported.

Table 1 lists some of the various \TeX features which may be used. *Supported* means that the package or macro may be used as-is, perhaps with minor limitations. *Emulated* means that the original package or macro is not used, but similar functionality is provided in a way which is intended to be compatible with the user's \TeX code.

Table 1: \TeX -HTML generation — lwarp package — Supported functions

| Category | Status |
|--------------|--|
| Engines: | pdf \TeX , X \TeX , Lua \TeX |
| Classes: | book, report, or article. memoir is planned. |
| Page layout: | Emulates fancyhdr, titleps. |

lwarp Supported Functions — continued

| Category | Status |
|-------------------------------------|--|
| Sectioning: | Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>FileDepth</code> 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 <code>titlesec</code> , <code>fncychap</code> . |
| Table of contents, figures, tables: | Supported, with hyperlinks. Supports <code>tocbibind</code> . Emulates <code>titletoc</code> and <code>tocloft</code> . |
| Title page: | <code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> . Optional titling-based commands for <code>published</code> and <code>subtitle</code> . |
| Front & back matter: | <code>abstract</code> , <code>appendix</code> |
| Cross-references: | Emulated, with hyperlinks. <code>hyperref</code> , <code>cleveref</code> , <code>varioref</code> , <code>url</code> . |
| Bibliography | Supported text-only. |
| Languages | <code>babel</code> . (<code>polyglossia</code> is untested.) |
| Margin notes: | <code>marginfit</code> , <code>marginfix</code> . |
| Footnotes: | <code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> . |
| Indexing: | <code>texindy</code> is used, with hyperlinks. <code>idxlayout</code> is emulated. |
| Glossary: | <code>glossaries</code> and <code>xindy</code> are used. |
| Bibliography: | Supported, without hyperlinks so far. |
| 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 conversion, due to native \LaTeX processing. |
| Theorems: | Support for native \LaTeX theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> . |
| Add'l math: | <code>delarray</code> , <code>bm</code> , math fonts via SVG images. |
| Units and fractions: | <code>siunitx</code> , <code>xfrac</code> , <code>nicefrac</code> , <code>units</code> |

lwarp Supported Functions — continued

| Category | Status |
|-------------------|---|
| Floats: | Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, capt-of, placeins, trivfloat, floatrow, keyfloat, wrapfig, cutwin, floatflt. |
| Tabular | tabular environment, array, tabularx, tabulary, threeparttable, multirow, longtable, 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. grffile is emulated. |
| rotating: | Emulated. All objects are displayed unrotated. |
| picture and tikz: | Converted to an SVG image. |
| xcolor: | Supported. Full package color names, any color models, and mixing . <code>\textcolor</code> , <code>\colorbox</code> , <code>\fcolorbox</code> . Enhanced for HTML compatibility. |
| Lists: | Standard L ^A T _E X environments are supported, along with enumitem, enumerate, paralist. Spacing is still controlled by CSS. |
| Environments: | Standard L ^A T _E X environments are supported. |
| minipage: | Supported with some HTML-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page. |
| Quotations: | verse, csquotes, epigraph |
| Verbatim: | verbatim, moreverb, fancyvrb (except for verbatim footnotes). |
| Frames: | framed, fancybox, mdframed, boxedminipage2e, shadow. |
| multicol: | Emulated, with CSS. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported. |

lwarp Supported Functions — continued

| Category | Status |
|---------------------------|--|
| Direct formatting: | <code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported. <code>lettrine</code> , <code>ulem</code> , <code>soul</code> , <code>reysize</code> , <code>scalegnt</code> , and <code>realscripts</code> are supported. |
| Ordinals: | <code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported. |
| Text ligatures: | Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> 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 <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> . |
| Rules: | <code>\rule</code> with <code>width</code> , <code>height</code> , <code>raise</code> , <code>text color</code> . |
| HTML reserved characters: | <code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities. |
| Fonts: | Used as-is. Appear in math expressions or embedded image environments. |
| Symbols: | Native \TeX diacriticals, <code>textcomp</code> . |
| Working as-is: | Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> . |
| Where: | |
| Supported: | The existing \TeX package is used. |
| Emulated: | The \TeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility. |

3 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.

3.1 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.

3.2 TeX4ht

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

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 some cases, lwarp provides a better HTML conversion, and it supports a different set of packages. TeX4ht produces several other forms of output beyond HTML.

3.3 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` **H^Ev^Ea**: <http://hevea.inria.fr/> (not on CTAN)

Prog `TtH` **T_TH**: <http://hutchinson.belmont.ma.us/tth/>

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

Prog `LaTeXML` **\LaTeX XML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog `LaTeX2HTML` **\LaTeX 2HTML**: <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

Prog `TeX2page` **T_EX2page**: <http://ds26gte.github.io/tex2page/index.html>

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

Prog GladTeX **GladTeX:** <http://humenda.github.io/GladTeX/>

3.4 AsciiDoc

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 AsciiDoc **AsciiDoc:** <http://asciidoc.org/> (The original version.)

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

Asciidoctor-LaTeX:

Prog Asciidoctor-LaTeX <http://www.noteshare.io/book/asciidoctor-latex-manual>
<https://github.com/asciidoctor/asciidoctor-latex>

3.5 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.)

3.6 Word processors

Prog Word 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.
Prog LibreOffice
Prog OpenOffice

3.7 Commercial systems

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™.

3.8 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.

4 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.

4.1 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.

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

```
Enter ⇒ tlmgr install lwarp
```

MiK \TeX : If using MiK \TeX , try using the package installer to install the package `lwarp`. Also update the package `miktex-misc`, which will install 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
```

MiK \TeX :

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
```

Table 2: Required software programs

Provided by your \LaTeX distribution:

From \TeX Live: <http://tug.org/texlive/>.

\LaTeX : pdf \LaTeX , xel \LaTeX , or lua \LaTeX .

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 pdf \LaTeX 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 MiK \TeX 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 MiK \TeX 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.

perl.org, strawberryperl.com

Automatically downloaded from the internet as required:

MathJax: Optionally used to display math. Automatically loaded from the MathJax website when needed.

From: mathjax.org

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 T_EX 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 T_EX 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 4.2.1 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 4.2.

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!

4.2 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 4.2.1.)

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 4.2.1.
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 T_EX 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 4.2.1.)

4. Create lwarpmk:

Unix: Create a symbolic link and make it executable:

- (a) Locate the T_EX 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 T_EX Live: Create a new lwarpmk.exe file:

- (a) Locate the T_EX 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.

4.2.1 Using a local copy of lwarpmk

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

1. When compiling the tutorial in section 5, 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.

4.3 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 40.

```
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 T_EX utilities xindy, latexmk, and pdfcrop may be provided by your operating system's package manager, and are also provided by the CTAN archive:

```
http://ctan.org/pkg/xindy
http://ctan.org/pkg/latexmk
http://ctan.org/pkg/pdftocrop
```

```
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 40.
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\
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 Strawberry 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.

5 Tutorial

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

5.1 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:

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

```
Enter ⇒ texdoc -l lwarp_tutorial.txt
```

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`.

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

⚠ 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
\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{lwrap_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}
\fbbox{\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.

5.2 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.³

5. Process the index.⁴⁵

```
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.

³Although, when using the utility `latexmk` (introduced later), the changed date is ignored and an actual change in contents must occur to cause a recompile.

⁴A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 7.4.6.

⁵Also see section 7.4.7 for index options.

5.3 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.html⁶, Some-math.html, and the document's index in _Index.html.⁷

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 5.4 and 5.5.

3. Force a recompile:

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

4. Process the HTML index and recompile:⁸⁹

```
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
```

⁶First-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.

⁷index.html is commonly used as a homepage, so the document index is in _Index.html.

⁸A lwarpmk htmlglossary command is also available to process a glossary produced with the glossaries package. See section 7.4.6.

⁹Also see section 7.4.7 for index options.

5.4 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 Tikz lwarp uses the same mechanism for `picture` and `Tikz` 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 `Tikz` 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.

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 5.5 to use MathJax instead.

 **Lots of files!**

5.5 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.

5.6 Changing the CSS style

`\CSSFilename` `\CSSFilename` may be used to choose which `.css` file is used to display each section of the web page. 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.

The styles provided by `lwarp` include:

`lwarp.css`: A default style if `\CSSFilename` is not used. This style is comparable to a plain \TeX 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 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.

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

`HomeHTMLFilename=index`

[filename underscores](#)

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:

`\item [\href{file_name.pdf}{text}] \`

Opt `HTMLFilename` **HTMLFilename:** A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.

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

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

Opt `mathjax` **mathjax:** Selects MathJax for math output.

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

Ctrl `tocdepth` **tocdepth:** Sectioning depth of the table of contents. See section 13 for a list of L^AT_EX stack depths.

Ctr SideTOCDepth

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

sideroc

Each subpage of the website has its own small table of contents on the side (the “sideroc”). Its depth is set by SideTOCDepth. This sideroc 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 sideroc 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
```

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



Ctr FileDepth

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

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. 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.

Inaccessible pages!

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

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.

HTML filenames

Example HTML filenames:

Numbered HTML nodes:

Example: Homepage index.html, and node-1, node-2. (See \SetHTMLFileNumber to number grouped by chapter, for example.)

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

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, etc.

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

\abstractname

\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

\CSSFilename: $\{\langle filename.css \rangle\}$ 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 \TeX 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 pages of the website.

| | |
|--------------------------------|--|
| <code>\HTMLLanguage</code> | <code>\HTMLLanguage:</code> The HTML file's <code>html lang</code> tag. Defaults to <code>en-US</code> . |
| <code>\HTMLAuthor</code> | <code>\HTMLAuthor:</code> The HTML header's meta author. Defaults to <code>\theauthor</code> . |
| <code>\HTMLDescription</code> | <code>\HTMLDescription: {\langle description \rangle}</code> Sets the HTML <code>description</code> tag for the following files. May be changed before each sectioning command which would cause a file split. |
| <code>\HTMLFirstPageTop</code> | <code>\HTMLFirstPageTop: {\langle contents \rangle}</code> A user-definable custom action applied to the top of the home page. Useful for logos, etc. Defaults empty. Ignored in print output. |
| <code>\HTMLPageTop</code> | <code>\HTMLPageTop: {\langle contents \rangle}</code> A user-definable custom action applied to the top of pages other than the home page. Useful for logos, etc. Defaults empty. <code>\LinkHome</code> may be used to place a link back to the homepage. Ignored in print output. |
| <code>\HTMLPageBottom</code> | <code>\HTMLPageBottom: {\langle contents \rangle}</code> A user-definable custom action applied to the bottom of each web page. Useful for authors, copyright notices, contact information, etc. Defaults empty. <code>\LinkHome</code> 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 <code>warpHTML</code> | 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. |
| <code>\warpprintonly</code> | \warpprintonly: <code>{\langle contents \rangle}</code> A macro version of the <code>warpprint</code> environment. |
| <code>\warpHTMLonly</code> | \warpHTMLonly: <code>{\langle contents \rangle}</code> A macro version of the <code>warpHTML</code> environment. |

5.8 Using latexmk

latexmk is a \TeX 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
```

5.9 Using XeLaTeX or LuaLaTeX

X_εLaTeX 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. X_εLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

5.10 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.

5.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

5.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

5.13 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 `lwarpmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

5.14 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.

6 Additional details

6.1 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 bit-mapped fonts.

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.



X_YTeX and LuaTeX 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 X_YTeX or LuaTeX:

- `fontspec` and font choices

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

- `fontenc`

- `inputenc`

- `newunicodechar`

- `\input glyphtounicode.tex`

Pkg `fontspec`

ligatures

Pkg `lmodern`

Pkg `fontenc`

Pkg `inputenc`

Pkg `newunicodechar`

File `glyphtounicode`

| | | |
|-----|-----------|---|
| | | – \input glyphtounicode-cmr.tex% from the pdfx package |
| | | – \pdfgentounicode=1 |
| Pkg | cmap | – cmap |
| Pkg | textcomp | – textcomp |
| Pkg | microtype | – 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. |
| | ligatures | |

3. `\usepackage{lwarp}` (section 6.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

6.1.1 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 \TeX . This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. \XeTeX or \LuaTeX are recommended for non-Latin languages.

For an older distribution of \TeX , 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 10.4.

6.2 lwarp package loading and options

lwarp supports `book`, `report`, and `article` classes.

| | | |
|-----|-----------|---|
| Pkg | lwarp | Load the lwarp package immediately after the font and UTF-8 setup commands. |
| Opt | warpprint | Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarp is loaded in print mode, it creates <code><project>_html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code><project>_html.tex</code> will create an HTML version. |
| Opt | warpHTML | |
| Opt | mathsvg | For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 7.5. |
| Opt | mathjax | |

Table 4: Package options

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

See table 4 for the full list of options.

6.3 Selecting the operating system

| | |
|-----------------|--|
| Prog Unix | 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 Mac OS | |
| Prog Linux | |
| Prog MS-Windows | If WINDOWS is not correctly detected, use the <code>lwarp</code> option <code>OSWindows</code> . |
| Prog Windows | When detected or specified, the operating-system path separator used by <code>lwarp</code> is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use. |
| Opt OSWindows | |

6.4 Selecting actions for print or HTML output

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

For most of built-in \TeX 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 HTML 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` $\{ \langle actions \rangle \}$

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

`\warpHTMLonly` $\{ \langle actions \rangle \}$

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

6.5 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 11: [Troubleshooting](#).

6.6 Title page

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

```
\title{Document Title} % One line only
\subtitle{Optional Document Subtitle \\ with optional multiple lines}
\author{Author One\affiliation{Affiliation One} \and
        Author Two\affiliation{Affiliation Two} }
\date{Optional date}
\published{Optional Journal Name \\ Optional multiple lines}
```

The title is used in the meta tags in the HTML files, and the rest are used in `\maketitle`.

| | |
|------------------------------|---|
| <code>\maketitle</code> | Use <code>\maketitle</code> just after the <code>\begin{document}</code> , as this will establish the title of the homepage. Optionally, use a <code>titlepage</code> environment instead. |
| Env <code>titlepage</code> | The <code>titlepage</code> environment may be used to hold a custom title page. The titlepage will be set in a <code><div></code> class <code>titlepage</code> , and <code>\printtitle</code> , etc. may be used inside this environment. |
| Env <code>titlingpage</code> | Another form of custom title page, where <code>\maketitle</code> is allowed, and additional information may be included as well. |
| <code>\title</code> | <code>{<title>}</code> |

⚠ Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use the `\subtitle` command instead. In HTML, the title will appear in a heading `<h1>`.

`\author` `{\langle author \rangle}`

⚠ In `\author`, use `\protect` before formatting commands such as `\textsc`. In HTML, the author will appear in a `<div>` class `author`. `\affiliation` is a new addition to lwarp.

`\date` `{\langle date \rangle}`

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

`\subtitle` `{\langle subtitle \rangle}`

A new command which sets a subtitle. Newlines are allowed. The default is empty. In HTML, this will appear in a `<div>` class `subtitle`.

`\published` `{\langle published \rangle}`

A new command which sets a publisher. The default is empty. In HTML, this will appear in a `<div>` class `published`.

`\thanks` `{\langle text \rangle}`

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

6.7 HTML page meta descriptions

`\HTMLDescription` `{\langle A description of the web page. \rangle}` The default is no description.

limitations 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 (`"`).

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 description 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{}`

6.8 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.

6.9 CSS

File `lwarp.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`. `project.css` should refer to `lwarp.css` as follows:

File `project.css`

File `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 --- ) */
```

An example file called `sample_project.css` is provided, and may be renamed `project.css`.

`\CSSFilename` For each section at which HTML files are split, `\CSSFilename` may be used before the sectioning command to select a css file for that and all following sections. 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}
...
```

6.10 Modifying xindy index processing

Prog `xindy` `lwarpmk` uses the file `lwarp.xdy` to process the index. This file is over-written by

File `lwarp.xdy`

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`.

7 Special cases and limitations

Also see section [11: Troubleshooting](#).

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

7.1 Formatting

7.1.1 Text formatting

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

7.1.2 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 `\quad`.

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

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

7.1.3 Text alignment

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

7.1.4 Accents

Native \TeX accents such as `\'` will work, but many more kinds of accents are available when using Unicode-aware \XeTeX and \LuaTeX .

7.1.5 Textcomp

 **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.

7.1.6 Superscripts and other non-math uses of math mode

Use `x` instead of `x`

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

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

7.1.8 Filenames in lists

filename underscore Escape underscores in the filenames:

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

7.1.9 relsize package

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 temporarily restored, and so should work as expected.

7.2 Boxes and minipages

7.2.1 Save Boxes

\TeX 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.

7.2.2 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 suppressed 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 `<div>` 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 \TeX PDF output they do not. Use a `flushleft` or similar environment in the child

minipage to force a text alignment.

7.2.3 Side-by-side minipages

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

7.2.4 Framed minipages and other environments

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

`\fbox` For a framed object, options include:
`\fboxBlock`
 Env `fminipage`

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.

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 `` 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.

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 \TeX code, the result in HTML is problematic since a table is a `<div>` not a ``, 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.

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

For inline minipage and lists:

For display `tabular`,
minipages, and lists:

that it is also framed. For HTML output, this forms a `<div>` 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 215 for xcolor’s `\colorbox` and `\fcolorbox`, and lwarp’s additional `\colorboxBlock` and `\fcolorboxBlock`.

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 110 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}
```

7.2.5 fancybox package

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
}
```

△ Misplaced alignment
tab character &

- 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.

7.2.6 mdframed package

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.

7.3 Cross-references

`\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.

7.3.1 Page references

The printed page does not translate to the HTML page, so references to page numbers are converted to parentheses containing `\pagerefPageFor`, which defaults to “see”, followed by a hyperlink to the appropriate object. Ex: “Sec. 1.23 on page (see sec. 1.23)”. `\pagerefPageFor` may be redefined to “page for”, empty, etc.

7.3.2 `cleveref` and `varioref` packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 7 to redefine the message which is printed for page number references.

7.3.3 Hyperlinks, hyperref, and 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.

7.3.4 Footnotes and page notes

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

7.4 Front and back matter

7.4.1 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.

`\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.

7.4.2 abstract package

 **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.

7.4.3 appendix package

 **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.

7.4.4 pagenote package

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

7.4.5 endnotes package

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
```

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

Opt `lwarpmk htmlglossary`

7.4.7 Index and the `tocbibind` package

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.

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 [63](#).

7.5 Math

 **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.

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

- SVG math option** For svg math, math is rendered as usual by \LaTeX into the initial PDF file using the current font¹⁰, 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 files** As currently implemented, each instance of math creates a new SVG file. 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 checksumming may be used to remove the need for duplicate files.
- SVG inline** Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.
- PNG files** Others 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 also larger as well.
- 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 8 regarding EPUB output with MathJax.
- MathJax math option** The popular MathJax alternative (mathjax.org) may be used to display math.
- Prog MathJax 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** Limitations when using MathJax include:
- Prog MathJax ¹⁰See section 216 regarding fonts and fractions.

- 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 `Tikz` 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.

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/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/mathjax/MathJax-third-party-extensions/tree/master/siunitx>
- \LaTeX macros
 - MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.
- custom MathJax macros For an example of using custom \LaTeX macros with MathJax, see page 319.

7.5.1 ntheorem package

- ⚠ 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.

7.5.2 siunitx package

- Pkg `siunitx` Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.
- ⚠ per-mode Some units will require that the expression be placed inside math mode.

7.5.3 units and nicefrac packages

Pkg `units` 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`.

7.5.4 newtxmath package

Pkg `newtxmath` The proper load order is:

⚠ loading sequence

1. ...
2. `\usepackage{lwarp}`
3. ...
4. `\usepackage{amsthm}`
5. `\usepackage{newtxmath}`
6. ...

7.6 Graphics

⚠ **graphics vs. graphicx** 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.

⚠ viewports

⚠ **`\graphicspath`** `\graphicspath` only works for a single directory; all `graphics` must be in this directory.

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`. Using 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.

⚠ **image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` refer-

ences to `.pdf` will automatically be changed to `.svg` for HTML output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.

`\rotatebox` `\rotatebox` accepts the optional origin key.

⚠ **browser support**

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

7.6.1 grffile package

⚠ **matching PDF and SVG encodings**

`grffile` is partially emulated. 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 version for HTML. No attempt is made to convert input or filename encodings.

7.6.2 color package

`color` is superseded 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.

7.6.3 xcolor package

`\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 `<div>` instead of a ``. 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.

7.7 Tabular

 `\multirow &`
`\multicolumn`

- When defining environments or macros which include `tabular` plus 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:

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

column types

- Vertical rules are not yet supported.
- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@`, `!`, `>`, and `<` may be used at each column, and they are used in that order.
- `\newcolumnstype` is ignored; unknown column types are set to `l`.
- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.

 `\multirow &`
`\multicolumn`

- `Multirow` and `multicolumn` cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)

 `\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 with rules`

- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.

rule at last row

- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.

 paragraphs

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

`\cmidrule width, trim`

- `\cmidrule` does not support width or trim options due to `css` limitations.

`\cmidrule and`
`\multicolumn`

- `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.

longtable headings

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

⚠ `\warpprintonly`

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` 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 relevant to HTML output.

⚠ `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 \\}
```

```
\warppHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

7.7.1 longtable package

- ⚠ 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 relevant to HTML output.

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

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

or place it inside `\warpingprintonly`.

7.7.2 float, trivfloat, and/or algorithmicx together

- ⚠ **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 209.1.

7.7.3 caption and subcaption packages

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

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

Similarly for subtable, subfigure, and longtable.

7.8 Floats

7.8.1 subfig package

 **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.

7.8.2 floatrow package

 **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.

7.9 Miscellaneous

7.9.1 verse and 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` 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.

`Len \leftmargini`
`Len \TMLvleftskip`
`Len \TMLleftmargini`

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.

7.9.2 newclude package

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



loading

Ex:

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

7.9.3 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 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

 **customized spacing** the user's changes do not affect the HTML output.

8 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:

| | | |
|------|------------|---|
| 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
```

then

```
Enter ⇒ lwarpmk html
```

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 file split.

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 5. 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.

9 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.

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

Bool FormatWordProcessor

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.

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

```
\booltrue{FormatWordProcessor}
```

to the project's source preamble after lwarp is loaded. 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 file split. These will have to be manually moved to their proper place in the document. lwarp does not know where the page breaks will be in the word processor's document, so the footnotes are simply moved to the end of each sectional break.
- 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 boolean may be set to help mark float boundaries:

Bool HTMLMarkFloats

```
Default true. Adds
    === table begin
    ...
    === end
or
    === figure begin
    ...
    === end
```

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

When enabled, markers are placed around each float, helping the user to identify float boundaries for further conversion to word processor frames and captions.

10 Modifying lwarp

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 \TeX output. See section 71 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 40.8.

10.1 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 Dropped, 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 Passed, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 167 (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.

10.2 Adding a package to the `lwarp.dtx` file

When adding a package to `lwarp.dtx` for permanent including 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 20. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

10.3 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.

10.4 Modifying `lwarpmk`

Prog `lwarpmk` 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.

File `lwarpmk.lua`

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`”.

11 Troubleshooting

11.1 Using the `lwarp.sty` package

Also see:

Section 6.5: [Commands to be placed into the `warpprint` environment](#)

Section 7: [Special cases and limitations](#)

Section 7: [Special cases and limitations](#)

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 6.1 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 5.7.

Misplaced alignment character `&`:

- If using `tabular` inside a `\fbox` or other macro, use `\StartDefiningTabulars` before the definition, and `\EndDefiningTabulars` after. See section 7.7

Obscure error messages:

- Be sure that a print version of the document compiles and that your document's \TeX code is correct, before attempting to generate an HTML version.

Missing sections:

- See section 5.7 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 5.7.

Missing / incorrect cross-references:

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- `\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` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 7 to redefine the message which is printed for page number references.

Em-dashes or En-dashes in listing captions and titles:

Use \XeTeX or \LuaTeX .

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]`.

HTML document contains a single unformatted print document:

- Be sure that the document selects `\usepackage[warpHTML]{lwarp}` instead of `[warpprint]`.

Images are appearing in strange places:

- `lwarpmk limages` to refresh the `lateximage` images.

⚠ Adding/removing

SVG images: When a math expression, `picture`, or `Tikz` 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 \TeX 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 5.5 to use `MathJax` instead.

“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.

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 5.7.
- 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.

11.1.1 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.

11.2 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}`

12 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 relevant 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.

Packages with graphical output are allowed as-is, but must be nested inside a `lateximage` environment to preserve the graphics.

There is still room to improve the factoring of the code, and doing so will become important if support for other output formats is added. Rather than wait until the code is pristine, the author felt it best to publish early and accept input before pushing on towards a perhaps less-than-ideal solution.

Testing has primarily been done with the Iceweasel/Firefox browser.

13 Stack depths

Stacks are created to track depth inside the \LaTeX document structure. This depth is translated to HTML headings as shown in table 5. “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.

Table 5: 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 | |
| subparagraph | 5 | |
| listitem | 7 | new for this package, used for list items |

It is possible to use HTML <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.

14 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 margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are

index entries 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”.

 **warnings** Special warnings are marked with a warning icon.

for HTML output: 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.
for PRINT output:
for HTML & PRINT:

15 Detecting the T_EX Engine — pdf_latex, lua_latex, xe_latex

```

1 \RequirePackage{iftex}
2
3 \ifLuaTeX
4 \RequirePackage{luatex85}% until the geometry package is updated
5 \fi

```

16 Unicode Input Characters

for HTML & PRINT:

If using pdf_latex, 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.

```

6
7 \RequirePackage{newunicodechar}
8
9 \newunicodechar{×}{\texttimes}
10
11 \ifPDFTeX
12 \newunicodechar{ff}{ff}% the first arguments are ligatures
13 \newunicodechar{fi}{fi}
14 \newunicodechar{fl}{fl}
15 \newunicodechar{ffi}{ffi}
16 \newunicodechar{ffl}{ffl}
17 \newunicodechar{--}{---}
18 \newunicodechar{-}{--}

```

In PDF_T_EX, preserve upright quotes in verbatim text:

```

19 \RequirePackage{upquote}
20 \else
21 \fi

```

17 Early package requirements

Pkg `etoolbox` Provides `\ifbool` and other functions.

```
22 \RequirePackage{etoolbox}[2011/01/03]
23 % requires v2.6 for \BeforeBeginEnvironment, etc.
```

Pkg `ifplatform` Provides `\ifwindows` to try to automatically detect Windows OS.

```
24 \RequirePackage{ifplatform}% sense op-system platform
```

Pkg `letltxmacro` Used to redefine `\textbf` and friends.

```
25 \RequirePackage{letltxmacro}
```

18 Operating-System portability

| | | |
|------|------------|--|
| Prog | Unix | 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 | Mac OS | |
| Prog | Linux | |
| Prog | MS-Windows | If WINDOWS is not correctly detected, use the lwarp option OSWindows. |
| Prog | Windows | 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. |
| Opt | OSWindows | |

18.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

```
26 \newbool{usingOSWindows}
27 \boolfalse{usingOSWindows}
```

18.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

```
28 \newcommand*{\OSPathSymbol}{/}
```

18.3 MS-Windows

For MS-Windows:

\LWR@setOSWindows Set defaults for the MS-Windows operating system. lwarp attempts to auto-detect the operating system, and the OSWindows option may also be used to force MS-Windows compatibility.

```
29 \newcommand*{\LWR@setOSWindows}
30 {
31 \booltrue{usingOSWindows}
32 \renewcommand*{\OSPathSymbol}{\@backslashchar}
33 }
```

Test for windows during compile. The user may also specify OSWindows package option in case this test fails.

```
34 \ifwindows
35 \LWR@setOSWindows
36 \fi
```

19 Package options

Pkg `kvoptions` Allows key/value package options.

```
37 \RequirePackage{kvoptions}
38 \SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool `warpingprint`

Bool `warpingHTML` Set to true/false depending on the package option selections for print/HTML/EPUB output and mathsvg/mathjax:

Bool `mathjax`

```
39 \newbool{warpingprint}
40 \newbool{warpingHTML}
41 \newbool{mathjax}
```

[defaults](#) The default is print output, and svg math if the user chose HTML output.

```
42 \booltrue{warpingprint}%
43 \boolfalse{warpingHTML}%
44 \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.

```
45 \DeclareVoidOption{warpprint}{%
46 \PackageInfo{lwarp}{Using option 'warpprint'}
47 \booltrue{warpingprint}%
48 \boolfalse{warpingHTML}%
49 }
```

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.

```
50 \DeclareVoidOption{warpHTML}{%
51 \PackageInfo{lwarp}{Using option 'warpHTML'}%
```

```
52 \booltrue{warpingHTML}%
53 \boolfalse{warpingprint}%
54 }
```

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.

```
55 \DeclareVoidOption{mathsvg}{%
56 \PackageInfo{lwarp}{Using option 'mathsvg'}}
57 \boolfalse{mathjax}%
58 }
```

Opt `mathjax` Option `mathjax` selects MathJax math display: If the `mathjax` option is given, boolean `mathjax` is true, may be used for `\ifbool` tests.

```
59 \DeclareVoidOption{mathjax}{%
60 \PackageInfo{lwarp}{Using option 'mathjax'}}
61 \booltrue{mathjax}%
62 }
```

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.

```
63 \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.

```
64 \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 "}")
```

```
65 \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`:

```

66 \newbool{LWR@creatinglwarpmk}
67 \boolfalse{LWR@creatinglwarpmk}
68
69 \DeclareVoidOption{lwarpmk}{
70 \PackageInfo{lwarp}{Using option 'lwarpmk'}}
71 \booltrue{LWR@creatinglwarpmk}
72 }

```

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.

```

73 \DeclareVoidOption{OSWindows}{
74 \PackageInfo{lwarp}{Using option 'OSWindows'}}
75 \LWR@setOSWindows
76 }

```

Opt `HomeHTMLFilename` 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.

```

77 \DeclareStringOption[] {HomeHTMLFilename}

```

Opt `HTMLFilename` 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.

```

78 \DeclareStringOption[] {HTMLFilename}

```

Opt `latexmk` Option `latexmk` tells `lwarpmk` to use `latexmk` when compiling documents.

```

79 \DeclareBoolOption[false]{latexmk}

```

Execute options Execute the package options, with the defaults which have been set just above:

```

80 \ProcessKeyvalOptions*\relax

```

Assign the `\BaseJobname` if the user hasn't provided one:

```

81 \providecommand*{\BaseJobname}{\LWR@BaseJobname}

```

Defaults unless already over-ridden by the user:

```

82 \ifcsemt{LWR@HomeHTMLFilename}{
83 \newcommand*{\HomeHTMLFilename}{\BaseJobname}

```

```

84 }{
85 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
86 }
87
88 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

19.1 Conditional compilation

`\warpprintonly` `{\langle contents \rangle}`

Only process the contents if producing printed output.

```
89 \newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}
```

`\warpHTMLonly` `{\langle contents \rangle}`

Only process the contents if producing HTML output.

```
90 \newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}
```

`Pkg comment` Provides conditional code blocks.

```
91 \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.

```

92 \ifbool{warpingHTML}{
93 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
94 }{}
95
96 \ifbool{warpingprint}{
97 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
98 }{}
99 \excludecomment{testing}

```

`Env warpall` Anything in the `warpall` environment will be generated for print or HTML outputs.

```
100 \includecomment{warpall}
```

`Env warpprint` Anything in the `warpprint` environment will be generated for print output only.

`Env warpHTML`

For HTML output:

```
101 \ifbool{warpingHTML}{%
102 \includecomment{warpHTML}
103 }
104 {\excludecomment{warpHTML}}%

105 \ifbool{warpingprint}
106 {\includecomment{warpprint}}
107 {\excludecomment{warpprint}}
```

Optionally generate a local copy of `lwarpmk`. Default to no.

```
108 \ifbool{LWR@creatinglwarpmk}
109 {\includecomment{LWR@createlwarpmk}}
110 {\excludecomment{LWR@createlwarpmk}}
```

20 Misplaced packages

Several packages should only be loaded before `lwarp`, and others should only be loaded after.

Packages which should only be loaded before `lwarp` have their own

```
lwarp-<packagename>.sty
```

which will trigger an error if they are loaded after `lwarp`. Examples include `fontspec`, `inputenc`, `fontenc`, and `newunicodechar`.

`\LWR@loadafter` $\langle\textit{packagename}\rangle$ Error if this package was loaded before `lwarp`.

```
111 \newcommand*\LWR@loadafter}[1]{%
112 \@ifpackageloaded{#1}
113 {
114 \PackageError{lwarp}
115 {Package #1, or one which uses #1, must be loaded after lwarp}
116 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
117 Package #1 may also be loaded by something else, which must also be moved
118 after lwarp.}
119 }
120 {}
121 }
```

`\LWR@loadbefore` $\langle\textit{packagename}\rangle$ Error if this package is after `lwarp`.

```

122 \newcommand*{\LWR@loadbefore}[1]{%
123 \@ifpackageloaded{#1}
124 {}
125 {
126 \PackageError{lwarp}
127 {Package #1 must be loaded before lwarp}
128 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
129 }
130 }

```

`\LWR@loadnever` $\langle\textit{badpackagename}\rangle$ $\langle\textit{replacementpkgname}\rangle$

The first packages is not supported, so tell the user to use the second instead.

```

131 \newcommand*{\LWR@loadnever}[2]{%
132 \PackageError{lwarp}
133 {Package #1 is not supported by lwarp's HTML conversion.
134 Package(s) #2 may be useful instead}
135 {Package #1 might conflict with lwarp in some way,
136 or is superceded by another package.
137 For a possible alternative, see package(s) #2.}
138 }

```

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:

```

139 \LWR@loadafter{abstract}
140 \LWR@loadafter{afterpage}
141 \LWR@loadafter{algorithmicx}
142 \LWR@loadafter{alltt}
143 \LWR@loadafter{amsthm}
144 \LWR@loadafter{appendix}
145 \LWR@loadafter{arabicfront}
146 \LWR@loadafter{array}
147 \LWR@loadafter{balance}
148 \LWR@loadafter{bookmark}
149 \LWR@loadafter{booktabs}
150 \LWR@loadafter{boxedminipage}
151 \LWR@loadafter{boxedminipage2e}
152 \LWR@loadafter{ccaption}
153 \LWR@loadafter{changepage}
154 \LWR@loadafter{chappg}
155 \LWR@loadafter{color}
156 \LWR@loadafter{crop}
157 \LWR@loadafter{cutwin}
158 \LWR@loadafter{dcolumn}

```

159 \LWR@loadafter{draftwatermark}
160 \LWR@loadafter{ellipsis}
161 \LWR@loadafter{emptypage}
162 \LWR@loadafter{enumerate}
163 \LWR@loadafter{enumitem}
164 \LWR@loadafter{epigraph}
165 \LWR@loadafter{eso-pic}
166 \LWR@loadafter{everypage}
167 \LWR@loadafter{everyshi}
168 \LWR@loadafter{extramarks}
169 \LWR@loadafter{fancybox}
170 \LWR@loadafter{fancyhdr}
171 \LWR@loadafter{fancyvrb}
172 \LWR@loadafter{figcaps}
173 \LWR@loadafter{fix2col}
174 \LWR@loadafter{float}
175 \LWR@loadafter{floatflt}
176 \LWR@loadafter{floatpag}
177 \LWR@loadafter{floatrow}
178 \LWR@loadafter{flushend}
179 \LWR@loadafter{fncychap}
180 \LWR@loadafter{ftnright}
181 \LWR@loadafter{fullpage}
182 \LWR@loadafter{geometry}
183 \LWR@loadafter{glossaries}
184 % \LWR@loadafter{graphics}% pre-loaded by xunicode
185 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
186 \LWR@loadafter{grffile}
187 \LWR@loadafter{hyperref}
188 \LWR@loadafter{hyperxmp}
189 \LWR@loadafter{idxlayout}
190 \LWR@loadafter{indentfirst}
191 \LWR@loadafter{keyfloat}
192 \LWR@loadafter{layout}
193 \LWR@loadafter{letterspace}
194 \LWR@loadafter{lettrine}
195 \LWR@loadafter{lips}
196 \LWR@loadafter{listings}
197 \LWR@loadafter{longtable}
198 \LWR@loadafter{lscape}
199 \LWR@loadafter{ltcaption}
200 \LWR@loadafter{marginfit}
201 \LWR@loadafter{marginfix}
202 \LWR@loadafter{marginnote}
203 \LWR@loadafter{mcaption}
204 \LWR@loadafter{mdframed}
205 \LWR@loadafter{metalogo}
206 \LWR@loadafter{microtype}
207 \LWR@loadafter{moreverb}
208 \LWR@loadafter{mparhack}

```
209 %\LWR@loadafter{multicol}% loaded by ltxdoc
210 \LWR@loadafter{multirow}
211 \LWR@loadafter{nameref}
212 \LWR@loadafter{needspace}
213 \LWR@loadafter{newtxmath}
214 \LWR@loadafter{nextpage}
215 \LWR@loadafter{nonumonpart}
216 \LWR@loadafter{nopageno}
217 \LWR@loadafter{nowidow}
218 \LWR@loadafter{nththeorem}
219 \LWR@loadafter{pagenote}
220 \LWR@loadafter{paralist}
221 \LWR@loadafter{parskip}
222 \LWR@loadafter{pdfscape}
223 \LWR@loadafter{pdfsync}
224 \LWR@loadafter{placeins}
225 \LWR@loadafter{prelim2e}
226 \LWR@loadafter{ragged2e}
227 \LWR@loadafter{realscripts}
228 \LWR@loadafter{relsize}
229 \LWR@loadafter{romanbar}
230 \LWR@loadafter{romanbarpagenumber}
231 \LWR@loadafter{rotating}
232 \LWR@loadafter{rotfloat}
233 \LWR@loadafter{savetrees}
234 % \LWR@loadafter{scalegnt}% loaded by babel-french
235 \LWR@loadafter{setspace}
236 \LWR@loadafter{shadow}
237 \LWR@loadafter{showidx}
238 \LWR@loadafter{showkeys}
239 \LWR@loadafter{sidecap}
240 \LWR@loadafter{sidenotes}
241 \LWR@loadafter{siunitx}
242 \LWR@loadafter{soul}
243 \LWR@loadafter{subfig}
244 \LWR@loadafter{tabularx}
245 \LWR@loadafter{tabulary}
246 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
247 \LWR@loadafter{textpos}
248 \LWR@loadafter{theorem}
249 \LWR@loadafter{threeparttable}
250 \LWR@loadafter{tikz}
251 \LWR@loadafter{titleps}
252 \LWR@loadafter{titlesec}
253 \LWR@loadafter{titletoc}
254 \LWR@loadafter{tocbibind}
255 \LWR@loadafter{tocloft}
256 \LWR@loadafter{transparent}
257 \LWR@loadafter{trivfloat}
258 \LWR@loadafter{ulem}
```

```

259 \LWR@loadafter{upref}
260 \LWR@loadafter{varioref}
261 \LWR@loadafter{verse}
262 \LWR@loadafter{wallpaper}
263 \LWR@loadafter{wrapfig}
264 \LWR@loadafter{xcolor}
265 \LWR@loadafter{xfrac}
266 \LWR@loadafter{xltextra}
267 \LWR@loadafter{xmpincl}

```

21 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: 268 \begin{warppall}

See: <http://tex.stackexchange.com/a/47579>.

Detects Xe_{La}TeX and Lua_{La}TeX:

```

269 \RequirePackage{iftex}
270 \newif\ifxetexorluatex
271 \ifXeTeX
272   \xetexorluatextrue
273 \else
274   \ifLuaTeX
275     \xetexorluatextrue
276   \else
277     \xetexorluatexfalse
278   \fi
279 \fi

280 \end{warppall}

```

for HTML output: 281 \begin{warpHTML}

```

282 \ifxetexorluatex
283 % ^^A \usepackage[no-math]{fontspec}

```

The monospaced font is used for HTML tags, so turn off its TeX ligatures and common ligatures:

```
284 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
285 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
286 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
287 \else
```

pdf_latex only: Only pre-loaded if pdf_latex is being used.

Pkg microtype

ligatures Older browsers don't display ligatures. Turn off letter ligatures, keeping \TeX dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```
288 \RequirePackage {microtype}
289
290 \microtypesetup{
291   protrusion=false,
292   expansion=false,
293   tracking=false,
294   kerning=false,
295   spacing=false}
296
297 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
298 \fi
299 \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:

```
300 \begin{warpHTML}
301 \RequirePackage[paperheight=190in,paperwidth=20in,%
302 left=2in,right=12in,%
303 top=1in,bottom=1in,%
304 ]{geometry}
```

```
305 \@twosidefalse
306 \@mparswitchfalse
307 \end{warpHTML}
```

for HTML & PRINT: 308 \begin{warpall}

Pkg xparse

L^AT_EX3 command argument parsing

```
309 \RequirePackage{xparse}
310 \end{warpall}
```

for HTML output: 311 \begin{warpHTML}

Pkg expl3

L^AT_EX3 programming

```
312 \RequirePackage{expl3}
```

Pkg gettitlestring

Used to emulate \nameref.

```
313 \RequirePackage{gettitlestring}
```

Pkg everyhook

everyhook is used to patch paragraph handling.

```
314 \RequirePackage{everyhook}
315 \end{warpHTML}
```

for HTML & PRINT: 316 \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>

```
317 \RequirePackage{filecontents}
318
319 \@ifpackagelater{filecontents}{2011/10/09}%
```

```
320 {}
321 {
322 \newwrite\fcwrite
323 \let\LWR@origfilec@ntents\filec@ntents
324 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
325 }

326 \end{warpall}

for HTML output: 327 \begin{warpHTML}

  Pkg  xifthen

          328 \RequirePackage{xifthen}

  Pkg  xstring

          329 \RequirePackage{xstring}

  Pkg  xstring

          330 \RequirePackage{verbatim}

  Pkg  makeidx

          331 \RequirePackage{makeidx}
          332 \makeindex

  Pkg  calc

          333 \RequirePackage{calc}

  Pkg  refcount

          334 \RequirePackage{refcount}

  Pkg  newfloat

          335 \RequirePackage{newfloat}

  Pkg  caption

          336 \RequirePackage{caption}

          337 \end{warpHTML}

for HTML & PRINT: 338 \begin{warpall}
```

Pkg `titling`

Used for `\maketitle` and the title page. See section 53.

```
339 \RequirePackage{titling}
```

Pkg `environ`

Used to encapsulate math environments for re-use in HTML ALT text.

```
340 \RequirePackage{environ}
```

```
341 \end{warpall}
```

for HTML output:

```
342 \begin{warpHTML}
```

Pkg `zref`

Used for cross-references.

```
343 \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.

```
344 \PassOptionsToPackage{leqno}{amsmath}
```

```
345 \RequirePackage{amsmath}
```

Pkg `xfrac`

Patched for HTML use. See section 216.

```
346 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
347 \RequirePackage{printlen}
```

```
348 \end{warpHTML}
```

22 Loading packages

for HTML output: 349 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

```
350 \LetLtxMacro{\LWR@origRequirePackage}{\RequirePackage}
```

`\LWR@requirepackagenames` Stores the list of required package names.

```
351 \newcommand*{\LWR@requirepackagenames}{} 
```

`\LWR@findword` [*⟨1: separator⟩*] [*⟨2: list⟩*] [*⟨3: index⟩*] [*⟨4: destination⟩*]

Note that argument 4 is passed directly to `\StrBetween`.

```
352 \newcommand*\LWR@findword[3][,]{%
353   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
354 }
```

`\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.

```
355 \newcommand*{\LWR@lookforpackagename}[1]{%
```

Find the *n*'th package name from the list:

```
356 \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`.

```
357 \StrSubstitute[100]{\LWR@strresult}{ }{\LWR@strresulttwo}% 
```

See if the package name was found:

```
358 \IfStrEq{\LWR@strresulttwo}{}%
359 {}% no filename
360 {}% yes filename
```

If found, and if an `lwarp-`equivalent name exists, use `lwarp-` instead.

```
361 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
362 {}% lwarp-* file found
363   \StrSubstitute%
```

```

364         {\LWR@requirepackagenames}%
365         {\LWR@strresult}%
366         {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
367 }%
368 {}% no lwarp-* file
369 }% yes filename
370 }

```

`\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 \TeX version.

```

371 \RenewDocumentCommand{\RequirePackage}{o m o}{%

```

Redirect up to nine names:

```

372 \renewcommand*{\LWR@requirepackagenames}{#2}
373 \LWR@lookforpackagename{1}
374 \LWR@lookforpackagename{2}
375 \LWR@lookforpackagename{3}
376 \LWR@lookforpackagename{4}
377 \LWR@lookforpackagename{5}
378 \LWR@lookforpackagename{6}
379 \LWR@lookforpackagename{7}
380 \LWR@lookforpackagename{8}
381 \LWR@lookforpackagename{9}

```

`\RequirePackage` depending on the options and version:

```

382 \IfValueTF{#1}
383 {% options given
384   \IfValueTF{#3}% version given?
385   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
386   {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
387 }
388 {% no options given
389   \IfValueTF{#3}% version given?
390   {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
391   {\LWR@origRequirePackage{\LWR@requirepackagenames}}
392 }
393 }
394 \LetLtxMacro{\usepackage}{\RequirePackage}

```

`\LWR@ProvidesPackagePass` {*⟨pkgname⟩*} [*⟨version⟩*]

Uses the original package, including options.

```

395 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
396 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
397 \IfValueTF{#2}
398 {\ProvidesPackage{lwarp-#1}[#2]}
399 {\ProvidesPackage{lwarp-#1}}
400 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
401 \ProcessOptions\relax
402
403 \IfValueTF{#2}
404 {\LWR@origRequirePackage{#1}[#2]}
405 {\LWR@origRequirePackage{#1}}
406 }

```

`\LWR@ProvidesPackageDrop` $\langle\{pkgname}\rangle$ [$\langle\{version}\rangle$]

Ignores the original package and uses lwarp’s version instead. Drops/discards all options.

```

407 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
408 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
409 \IfValueTF{#2}
410 {\ProvidesPackage{lwarp-#1}[#2]}
411 {\ProvidesPackage{lwarp-#1}}
412 \DeclareOption*{}
413 \ProcessOptions\relax
414 }

415 \end{warpHTML}

```

23 File handles

Defines file handles for writes.

for HTML & PRINT: 416 `\begin{warpall}`

`\LWR@quickfile` For quick temporary use only. This is reused in several places.

```
417 \newwrite\LWR@quickfile%
```

```
418 \end{warpall}
```

for HTML output: 419 `\begin{warpHTML}`

`\LWR@lateximagesfile` For `lateximages.txt`.

```
420 \newwrite\LWR@lateximagesfile
```

```
421 \end{warpHTML}
```

24 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: 422 `\begin{warpHTML}`

```
\include {<filename>}
```

```
\@include {<filename>} Modified to load _html_inc files.
```

```
423 \def\@include#1 {%
424 \immediate\openout\LWR@quickfile #1_html_inc.tex% new
425 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% new
426 \immediate\closeout\LWR@quickfile% new
427 \LWR@origclearpage% \changed
428 \if@filesw
429   \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
430 \fi
431 \@tempswatru
432 \if@partsw
433   \@tempswafalse
434   \edef\reserved@b{#1}%
435   \@for\reserved@a:=\@partlist\do
436     {\ifx\reserved@a\reserved@b\@tempswatru\fi}%
437 \fi
438 \if@tempswa
439   \let\@auxout\@partaux
440   \if@filesw
441     \immediate\openout\@partaux #1_html_inc.aux % changed
442     \immediate\write\@partaux{\relax}%
443   \fi
444   \@input{#1_html_inc.tex}% changed
```

```

445 \LWR@origclearpage% changed
446 \@writeckpt{#1}%
447 \if@filesw
448     \immediate\closeout\@partaux
449 \fi
450 \else
451     \deadcycles\z@
452     \@nameuse{cp@#1}%
453 \fi
454 \let\@auxout\@mainaux%
455 }

456 \end{warpHTML}

```

25 Copying a file

for HTML output: 457 \begin{warpHTML}

\LWR@copyfile $\{ \langle source\ filename \rangle \} \{ \langle destination\ filename \rangle \}$

Used to copy the .toc file to .sidetoc to re-print the TOC in the sideroc navigation pane.

```

458 \newwrite\LWR@copyoutfile % open the file to write to
459 \newread\LWR@copyinfile % open the file to read from
460
461 \newcommand*{\LWR@copyfile}[2]{%
462 \immediate\openout\LWR@copyoutfile=#2
463 \openin\LWR@copyinfile=#1
464 \begingroup\endlinechar=-1
465 \makeatletter
466 \loop\unless\ifeof\LWR@copyinfile
467 \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
468 % \LWR@fileline\par % print the content into the pdf
469 % print the content:
470 \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
471 \repeat
472 \closeout\LWR@copyoutfile
473 \endgroup
474 }

475 \end{warpHTML}

```

26 Debugging messages

```
476 \begin{warpall}
```

Bool `LWR@tracinglwarp` True if tracing is turned on.

```
477 \newbool{LWR@tracinglwarp}
```

`\tracinglwarp` Turns on the debug tracing messages.

```
478 \newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}
```

`\LWR@traceinfo` $\langle text \rangle$ If tracing is turned on, writes the text to the `.log` file.

```
479 \newcommand{\LWR@traceinfo}[1]{%
480 \ifbool{LWR@tracinglwarp}%
481 {%
482   \typeout{*** lwarp: #1}%
483   % \PackageInfo{lwarp}{#1 : }%
484 }%
485 {}}%
486 }
```

Bool `HTMLDebugComments` Default false. Add comments in HTML about closing `<div>`s, sections, etc.

```
487 \newbool{HTMLDebugComments}
488 \boolfalse{HTMLDebugComments}
```

If `\tracinglwarp`, show where preamble hooks occur:

```
489 \AfterEndPreamble{
490 \LWR@traceinfo{AfterEndPreamble}
491 }
492
493 \AtBeginDocument{
494 \LWR@traceinfo{AtBeginDocument}
495 }
```

27 HTML-conversion output modifications

These booleans modify the HTML output in various ways to improve conversion to EPUB or word processor imports.

Bool `FormatEPUB` Default false. Changes HTML output for easy EPUB conversion via an external program.

Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.

```
496 \newbool{FormatEpub}
497 \boolfalse{FormatEpub}
```

Bool `FormatWordProcessor` 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.

```
498 \newbool{FormatWordProcessor}
499 \boolfalse{FormatWordProcessor}
```

Bool `HTMLMarkFloats` Default true. Adds

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

or

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

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames.¹¹

```
500 \newbool{HTMLMarkFloats}
501 \booltrue{HTMLMarkFloats}
```

```
502 \end{warpall}
```

28 Remembering original formatting macros

for HTML output: 503 \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.

```
504 \LetLtxMacro{\LWR@origtextrm}{\textrm}
505 \LetLtxMacro{\LWR@origtextsf}{\textsf}
506 \LetLtxMacro{\LWR@origtexttt}{\texttt}
507 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
```

¹¹Perhaps some day word processors will have HTML import options for identifying `<div>` classes for figures and tables, along with the associated captions.

```
508 \LetLtxMacro{\LWR@origtextbf}{\textbf}
509 \LetLtxMacro{\LWR@origtextmd}{\textmd}
510 \LetLtxMacro{\LWR@origtextit}{\textit}
511 \LetLtxMacro{\LWR@origtextsl}{\textsl}
512 \LetLtxMacro{\LWR@origtextsc}{\textsc}
513 \LetLtxMacro{\LWR@origtextup}{\textup}
514 \LetLtxMacro{\LWR@origemph}{\emph}
515
516 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
517 \LetLtxMacro{\LWR@origsffamily}{\sffamily}
518 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
519 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
520 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
521 \LetLtxMacro{\LWR@origupshape}{\upshape}
522 \LetLtxMacro{\LWR@origslshape}{\slshape}
523 \LetLtxMacro{\LWR@origscshape}{\scshape}
524 \LetLtxMacro{\LWR@origitshape}{\itshape}
525 \LetLtxMacro{\LWR@origem}{\em}
526 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
527
528 \let\LWR@origraggedright\raggedright
529 \let\LWR@origonecolumn\onecolumn
530
531 \let\LWR@origsp\sp
532 \let\LWR@origsb\sb
533 \let\LWR@origtextsuperscript\textsuperscript
534 \let\LWR@origtextsubscript\textsubscript
535
536 \let\LWR@origscriptsize\scriptsize
537
538 \let\LWR@orignewpage\newpage
539
540 \let\LWR@origpagestyle\pagestyle
541 \let\LWR@origthispagestyle\thispagestyle
542 \let\LWR@origpagenumbering\pagenumbering
543
544 \LetLtxMacro{\LWR@origminipage}{\minipage}
545 \let\LWR@origendminipage\endminipage
546 \LetLtxMacro{\LWR@origparbox}{\parbox}
547
548 \let\LWR@orignewline\newline
549
550 \LetLtxMacro{\LWR@origitem}{\item}
551
552 \LetLtxMacro{\LWR@origitemize}{\itemize}
553 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
554 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
555 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
556 \LetLtxMacro{\LWR@origdescription}{\description}
557 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
```

```

558
559
560 \let\LWR@origpar\par
561
562
563 \LetLtxMacro{\LWR@origfootnote}{\footnote}
564 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
565
566 \let\LWR@origclearpage\clearpage
567 \let\clearpage\relax
568 \let\cleardoublepage\relax

569 \end{warpHTML}

```

29 Textcomp

for HTML output: 570 \begin{warpHTML}

If `textcomp` was loaded before `lwarp`, perhaps as part of the font-related packages, explicitly load the `lwarp` patches now:

```

571 \@ifpackageloaded{textcomp}
572 {
573 \LWR@origRequirePackage{lwarp-textcomp}
574 }

```

If `textcomp` is loaded after `warp`, `lwarp` will automatically load `lwarp-textcomp` at that time. Meanwhile, define a null version of `\LWR@restoreorigtextcomp` until `textcomp` is actually loaded, at which point `lwarp-textcomp` will replace `\LWR@restoreorigtextcomp`.

```

575 {
576 \providecommand*\LWR@restoreorigtextcomp{}%
577 }

578 \end{warpHTML}

```

30 Accents

Native \TeX accents such as `\`` will work, but many more kinds of accents are available when using Unicode-aware \XeTeX and \LuaTeX .

for HTML output: 579 \begin{warpHTML}

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
580 \AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only `\t` needs to be redefined:

```
581 \LetLtxMacro{\LWR@origt}{\t}
```

For `pdfLaTeX`, additional work is required:

```
582 \ifPDFTeX
583 \LetLtxMacro{\LWR@origequalaccent}{\=}
584 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
585 \LetLtxMacro{\LWR@origu}{\u}
586 \LetLtxMacro{\LWR@origv}{\v}
587 \LetLtxMacro{\LWR@origc}{\c}
588 \LetLtxMacro{\LWR@origd}{\d}
589 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For `pdfLaTeX`, Unicode diacritical marks are used:

```
590 \renewcommand*{\=} [1]{\#1\HTMLUnicode{0305}}
591 \renewcommand*{\cdot} [1]{\#1\HTMLUnicode{0307}}
592 \renewcommand*{\u} [1]{\#1\HTMLUnicode{0306}}
593 \renewcommand*{\v} [1]{\#1\HTMLUnicode{030C}}
594 \renewcommand*{\c} [1]{\#1\HTMLUnicode{0327}}
595 \renewcommand*{\d} [1]{\#1\HTMLUnicode{0323}}
596 \renewcommand*{\b} [1]{\#1\HTMLUnicode{0331}}
597 \fi
```

For all engines, a Unicode diacritical tie is used:

```
598 \def\LWR@t#1#2{\#1\HTMLUnicode{0361}\#2}
599 \renewcommand*{\t} [1]{\LWR@t#1}
```

`\LWR@restoreorigaccents` Called from `\restoreoriginalformatting` when a `lateximage` is begun.

```
600 \ifPDFTeX
601 \newcommand*{\LWR@restoreorigaccents}{%
602 \LetLtxMacro{\=}{\LWR@origequalaccent}%
603 \LetLtxMacro{\cdot}{\LWR@origdotaccent}%
604 \LetLtxMacro{\u}{\LWR@origu}%
605 \LetLtxMacro{\v}{\LWR@origv}%
606 \LetLtxMacro{\t}{\LWR@origt}%
```

```

607 \LetLtxMacro{\c}{\LWR@origc}%
608 \LetLtxMacro{\d}{\LWR@origd}%
609 \LetLtxMacro{\b}{\LWR@origb}%
610 }
611 \else% XeLaTeX, LuaLaTeX:
612 \newcommand*{\LWR@restoreorigaccents}{%
613 \LetLtxMacro{\t}{\LWR@origt}%
614 }
615 \fi
616 }% AtBeginDocument

617 \end{warpHTML}

```

31 Configuration Files

```

618 \begin{warpprint}
619 \typeout{lwarp: generating configuration files}
620 \end{warpprint}

```

31.1 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.

```

621 \begin{warpprint}
622 \immediate\openout\LWR@quickfile=\jobname_html.tex
623 \immediate\write\LWR@quickfile{%
624 \detokenize{\PassOptionsToPackage}%
625 {warpHTML,BaseJobname=\jobname}{lwarp}%
626 }
627 \immediate\write\LWR@quickfile{%
628 \detokenize{\input}\string{\jobname.tex}\string }%
629 }
630 \immediate\closeout\LWR@quickfile
631 \end{warpprint}

```

31.2 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:

```
632 \begin{warpprint}
633 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
634 \immediate\openout\LWR@quickfile=lwarpmk.conf
635 \ifbool{usingOSWindows}{
636 \immediate\write\LWR@quickfile{opsystem = "Windows"}
637 }{
638 \immediate\write\LWR@quickfile{opsystem = "Unix"}
639 }
640 \ifPDFTeX
641 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
642 \fi
643 \ifXeTeX
644 \immediate\write\LWR@quickfile{latexname = "xelatex"}
645 \fi
646 \ifLuaTeX
647 \immediate\write\LWR@quickfile{latexname = "lualatex"}
648 \fi
649 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
650 \immediate\write\LWR@quickfile{%
651 homehtmlfilename = "\HomeHTMLFilename"%
652 }
653 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
654 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
655 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
656 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
657 \immediate\closeout\LWR@quickfile
658 \end{warpprint}
```

31.3 project.lwarpmkconf

File `project.lwarpmkconf` A project-specific configuration file for `lwarpmk`.

```
659 \begin{warpprint}
660 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
661 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
```

```

662 \ifbool{usingOSWindows}{
663 \immediate\write\LWR@quickfile{opsystem = "Windows"}
664 }{
665 \immediate\write\LWR@quickfile{opsystem = "Unix"}
666 }
667 \ifPDFTeX
668 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
669 \fi
670 \ifXeTeX
671 \immediate\write\LWR@quickfile{latexname = "xelatex"}
672 \fi
673 \ifLuaTeX
674 \immediate\write\LWR@quickfile{latexname = "lualatex"}
675 \fi
676 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
677 \immediate\write\LWR@quickfile{%
678 homehtmlfilename = "\HomeHTMLFilename"%
679 }
680 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
681 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
682 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
683 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
684 \immediate\closeout\LWR@quickfile
685 \end{warpprint}

```

31.4 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.

```

686 \begin{warpprint}
687 \begin{filecontents*}{lwarp.css}
688 /*
689 CSS stylesheet for the LaTeX lwarp package
690 Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
691 */
692
693
694 /* a fix for older browsers: */
695 header, section, footer, aside, nav, main,
696 article, figure { display: block; }
697
698
699 A:link {color:#000080 ; text-decoration: none ; }
700 A:visited {color:#800000 ; }

```

```
701 A:hover {color:#000080 ; text-decoration: underline ;}
702 A:active {color:#800000 ; }
703
704 a.tocpart {display: inline-block ; margin-left: 0em ;
705     font-weight: bold ;}
706 a.tocchapter {display: inline-block ; margin-left: 0em ;
707     font-weight: bold ;}
708 a.tocsection {display: inline-block ; margin-left: 1em ;
709     text-indent: -.5em ; font-weight: bold ; }
710 a.tocsubsection {display: inline-block ; margin-left: 2em ;
711     text-indent: -.5em ; }
712 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
713     text-indent: -.5em ; }
714 a.tocparagraph {display: inline-block ; margin-left: 4em ;
715     text-indent: -.5em ; }
716 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
717     text-indent: -.5em ; }
718 a.tocfigure {margin-left: 0em}
719 a.tocsubfigure {margin-left: 2em}
720 a.tocatable {margin-left: 0em}
721 a.tocsubtable {margin-left: 2em}
722 a.toctheorem {margin-left: 0em}
723 a.toclstlisting {margin-left: 0em}
724
725
726 body {
727     font-family: "DejaVu Serif", "Bitstream Vera Serif",
728         "Lucida Bright", Georgia, serif;
729     background: #FAF7F4 ;
730     color: black ;
731     margin: 0em ;
732     padding: 0em ;
733     font-size: 100% ;
734     line-height: 1.2 ;
735 }
736
737 p {margin: 1.5ex 0em 1.5ex 0em ;}
738
739 /* Holds a section number to add space between it and the name */
740 span.sectionnumber { margin-right: 0em }
741
742 /* Inserted in front of index lines */
743 span.indexitem {margin-left: 0em}
744 span.indexsubitem {margin-left: 2em}
745 span.indexsubsubitem {margin-left: 4em}
746
747 div.hidden, span.hidden { display: none ; }
748
749 kbd {
750     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
```

```
751     "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
752     "Courier New", monospace;
753     font-size: 100% ;
754 }
755
756 pre { padding: 3pt ; }
757
758 span.strong { font-weight: bold; }
759
760 span.textmd { font-weight: normal; }
761
762 span.textsc { font-variant: small-caps; }
763
764 span.textsl { font-style: oblique; }
765
766 span.textup { font-variant: normal; }
767
768 span.textrm {
769     font-family: "DejaVu Serif", "Bitstream Vera Serif",
770     "Lucida Bright", Georgia, serif;
771 }
772
773 span.textsf {
774     font-family: "DejaVu Sans", "Bitstream Vera Sans",
775     Geneva, Verdana, sans-serif ;
776 }
777
778 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
779
780
781
782 /* For realscripts */
783 .supsubscript {
784     display: inline-block;
785     text-align:left ;
786 }
787
788 .supsubscript sup,
789 .supsubscript sub {
790     position: relative;
791     display: block;
792     font-size: .5em;
793     line-height: 1;
794 }
795
796 .supsubscript sup {
797     top: .5em;
798 }
799
800 .supsubscript sub {
```

```
801     top: .5em;
802 }
803
804
805
806
807 span.attribution {
808     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
809 }
810
811 span.citetitle {
812     margin-left: 1em ; font-size: 80% ; font-style: oblique;
813 }
814
815 span.poemtitle {
816     font-size: 120% ; font-weight: bold;
817 }
818
819 blockquote {
820     margin-left: 0px ;
821     margin-right: 0px ;
822 }
823
824 blockquote p {
825     line-height: 1.5;
826     text-align: left ;
827     font-size: .85em ;
828     margin-left: 3em ;
829     margin-right: 3em ;
830 }
831
832 blockquotation {
833     margin-left: 0px ;
834     margin-right: 0px ;
835 }
836
837 blockquotation p {
838     line-height: 1.5;
839     text-align: left ;
840     font-size: .85em ;
841     margin-left: 3em ;
842     margin-right: 3em ;
843 }
844
845 div.epigraph {
846     line-height: 1.2;
847     text-align: left ;
848     padding: 3ex 1em 0ex 1em ;
849 /*     margin: 3ex auto 3ex auto ; /* /* Epigraph centered */
850     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
```

```
851/*    margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
852    font-size: .85em ;
853    max-width: 27em ;
854}
855
856
857
858div.epigraphsource{
859    text-align:right ;
860    margin-left:auto ;
861/*    max-width: 50% ; */
862    border-top: 1px solid #A0A0A0 ;
863    padding-bottom: 3ex ;
864    line-height: 1.2;
865}
866
867div.epigraph p { padding: .5ex ; margin: 0ex ;}
868div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
869
870
871/* lettrine package: */
872span.lettrine { font-size: 3ex ; float: left ; }
873span.lettrinetext { font-variant: small-caps ; }
874
875/* ulem and soul packages: */
876span.uline {
877    text-decoration: underline ;
878    text-decoration-skip ;
879}
880
881span.uuline {
882    text-decoration: underline ;
883    text-decoration-skip ;
884    text-decoration-style: double ;
885}
886
887span.uwave {
888    text-decoration: underline ;
889    text-decoration-skip ;
890    text-decoration-style: wavy ;
891}
892
893span.sout {
894    text-decoration: line-through ;
895}
896
897span.xout {
898    text-decoration: line-through ;
899}
900
```

```
901 span.dashuline {
902     text-decoration: underline ;
903     text-decoration-skip ;
904     text-decoration-style: dashed ;
905 }
906
907 span.dotuline {
908     text-decoration: underline ;
909     text-decoration-skip ;
910     text-decoration-style: dotted ;
911 }
912
913 span.letterspacing { letter-spacing: .2ex ; }
914
915 span.capsspacing {
916     font-variant: small-caps ;
917     letter-spacing: .1ex ;
918 }
919
920 span.highlight { background: #F8E800 ; }
921
922
923
924
925 html body {
926     margin: 0 ;
927     line-height: 1.2;
928 }
929
930
931 body div {
932     margin: 0ex;
933 }
934
935
936 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
937 {
938     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
939         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
940         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
941         "Times New Roman", serif;
942     font-style: normal ;
943     font-weight: bold ;
944     text-align: left ;
945 }
946
947 h1 { /* title of the entire website, used on each page */
948     text-align: center ;
949     font-size: 2.5em ;
950     padding: .4ex 0em 0ex 0em ;
```

```
951 }
952 h2 { font-size: 2.25em }
953 h3 { font-size: 2em }
954 h4 { font-size: 1.75em }
955 h5 { font-size: 1.5em }
956 h6 { font-size: 1.25em }
957 span.paragraph {font-size: 1em ; font-variant: normal ;
958     margin-right: 1em ; }
959 span.subparagraph {font-size: 1em ; font-variant: normal ;
960     margin-right: 1em ; }
961
962
963
964 /* Title of the file */
965 h1 {
966     margin: 0ex 0em 0ex 0em ;
967     line-height: 1.3;
968     text-align: center ;
969 }
970
971 /* Part */
972 h2 {
973     margin: 1ex 0em 1ex 0em ;
974     line-height: 1.3;
975     text-align: center ;
976 }
977
978 /* Chapter */
979 h3 {
980     margin: 3ex 0em 1ex 0em ;
981     line-height: 1.3;
982 }
983
984 /* Section */
985 h4 {
986     margin: 3ex 0em 1ex 0em ;
987     line-height: 1.3;
988 }
989
990 /* Sub-Section */
991 h5 {
992     margin: 3ex 0em 1ex 0em ;
993     line-height: 1.3;
994 }
995
996 /* Sub-Sub-Section */
997 h6 {
998     margin: 3ex 0em 1ex 0em ;
999     line-height: 1.3;
1000 }
```

```
1001
1002
1003 div.titlepage {
1004   text-align: center ;
1005 }
1006
1007 .footnotes {
1008   font-size: .85em ;
1009   margin: 3ex 1em 0ex 1em ;
1010   padding-bottom: 1ex ;
1011   border-top: 1px solid silver ;
1012 }
1013
1014 .marginpar {
1015   max-width:50%;
1016   float:right;
1017   text-align:left;
1018   margin: 1ex 0.5em 1ex 1em ;
1019   padding: 1ex 0.5em 1ex 0.5em ;
1020   font-size: 85% ;
1021   border-top: 1px solid silver ;
1022   border-bottom: 1px solid silver ;
1023   overflow-x: auto;
1024 }
1025
1026 .marginpar br { margin-bottom: 2ex ; }
1027
1028 div.marginblock {
1029   max-width:50%;
1030   float:right;
1031   text-align:left;
1032   margin: 1ex 0.5em 1ex 1em ;
1033   padding: 1ex 0.5em 1ex 0.5em ;
1034   overflow-x: auto;
1035 }
1036
1037 div.marginblock div.minipage {
1038   display: block ;
1039   margin: 0pt auto 0pt auto ;
1040 }
1041
1042 div.marginblock div.minipage p { font-size: 85%}
1043
1044 div.marginblock br { margin-bottom: 2ex ; }
1045
1046
1047 section.textbody div.footnotes{
1048   margin: 3ex 0em 0ex 0em ;
1049   border-bottom: 2px solid silver ;
1050 }
```

```
1051
1052 .footnoteheader {
1053     border-top: 2px solid silver ;
1054     margin-top: 3ex ;
1055     padding-top: 1ex ;
1056     font-weight: bold ;
1057 }
1058
1059 .mpfootnotes {
1060     text-align: left ;
1061     font-size: .85em ;
1062     margin-left: 1em ;
1063     border-top: 1px solid silver ;
1064 }
1065
1066 /* Remove footnote top border in the title page. */
1067 div.titlepage div.mpfootnotes {
1068     border-top: none ;
1069 }
1070
1071
1072
1073 ol {
1074     margin: 1ex 1em 1ex 0em;
1075     line-height: 1.2;
1076 }
1077
1078 ul, body dir, body menu {
1079     margin: 1ex 1em 1ex 0em;
1080     line-height: 1.2;
1081 }
1082
1083 li { margin: 0ex 0em 1ex 0em; }
1084
1085 html {
1086     margin: 0;
1087     padding: 0;
1088 }
1089
1090 .programlisting {
1091     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1092         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1093         "Courier New", monospace;
1094     margin: 1ex 0ex 1ex 0ex ;
1095     padding: .5ex 0pt .5ex 0pt ;
1096     overflow-x: auto;
1097 }
1098
1099 section.textbody>pre.programlisting {
1100 border-top: 1px solid silver ;
```

```
1101 border-bottom: 1px solid silver ;
1102 }
1103
1104
1105 .inlineprogramlisting {
1106   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1107             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1108             "Courier New", monospace;
1109   overflow-x: auto;
1110 }
1111
1112 span.listinglabel {
1113   display: inline-block ;
1114   font-size: 70% ;
1115   width: 4em ;
1116   text-align: right ;
1117   margin-right: 2em ;
1118 }
1119
1120 div.abstract {
1121   margin: 2em 5% 2em 5% ;
1122   padding: 1ex 1em 1ex 1em ;
1123   /* font-weight: bold ; */
1124   font-size: 90% ;
1125 }
1126
1127 div.abstract dl {line-height:1.5;}
1128 div.abstract dt {color:#304070;}
1129
1130 div.abstracttitle{
1131   font-family: "URW Classico", Optima, "Linux Biolinum O",
1132             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1133             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1134   font-weight:bold;
1135   font-size:1.25em;
1136   text-align: center ;
1137 }
1138
1139 span.abstracrunintitle{
1140   font-family: "URW Classico", Optima, "Linux Biolinum O",
1141             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1142             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1143   font-weight:bold;
1144 }
1145
1146
1147 .verbatim {
1148   overflow-x: auto ;
1149 }
1150
```

```
1151 .alltt {
1152     overflow-x: auto ;
1153 }
1154
1155
1156 .bverbatim {
1157     margin: 1ex Opt 1ex Opt ;
1158     padding: .5ex Opt .5ex Opt ;
1159     overflow-x: auto ;
1160 }
1161
1162 .lverbatim {
1163     margin: 1ex Opt 1ex Opt ;
1164     padding: .5ex Opt .5ex Opt ;
1165     overflow-x: auto ;
1166 }
1167
1168 .fancyvrb {
1169     font-size:.85em ;
1170     margin: 3ex Opt 3ex Opt
1171 }
1172
1173 .fancyvrblabel {
1174     font-weight:bold;
1175     text-align: center ;
1176 }
1177
1178
1179 .verse {
1180     font-family: "Linux Libertine Mono O", "Lucida Console",
1181                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1182                 "Liberation Mono", "FreeMono", "Andale Mono",
1183                 "Nimbus Mono L", "Courier New", monospace;
1184     margin-left: 1em ;
1185 }
1186
1187
1188 div.singlespace { line-height: 1.2 ; }
1189 div.onehalfspace { line-height: 1.5 ; }
1190 div.doublespace { line-height: 2 ; }
1191
1192
1193
1194
1195
1196 /* Minipage environments, vertically aligned to top, center, bottom: */
1197 .minipage, .fminipage, .fcolorminipage {
1198     /* display: inline-block ; */
1199     /* Mini pages which follow each other will be tiled. */
1200     margin: .25em .25em .25em .25em;
```

```
1201     padding: .25em .25em .25em .25em;
1202     display: inline-flex;
1203     flex-direction: column ;
1204     overflow: auto;
1205 }
1206
1207 /* Paragraphs in the flexbox did not collapse their margins. */
1208 /* Have not yet researched this. */
1209 .minipage p {margin: .75ex 0em .75ex 0em ;}
1210
1211 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1212 .fcolorbox .minipage, .fcolorboxBlock .minipage
1213     {border: none ; background: none;}
1214
1215 .fbox, .fboxBlock { border: 1px solid black ; }
1216
1217 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1218 .fminipage, .fcolorminipage
1219     {display: inline-block}
1220
1221 .shadowbox, .shabox {
1222     border: 1px solid black;
1223     box-shadow: 3px 3px 3px #808080 ;
1224     border-radius: 0px ;
1225     padding: .4ex .3em .4ex .3em ;
1226     margin: 0pt .3ex 0pt .3ex ;
1227     display: inline-block ;
1228 }
1229
1230 .doublebox {
1231     border: 3px double black;
1232     border-radius: 0px ;
1233     padding: .4ex .3em .4ex .3em ;
1234     margin: 0pt .3ex 0pt .3ex ;
1235     display: inline-block ;
1236 }
1237
1238 .ovalbox, .Ovalbox {
1239     border: 1px solid black;
1240     border-radius: 1ex ;
1241     padding: .4ex .3em .4ex .3em ;
1242     margin: 0pt .3ex 0pt .3ex ;
1243     display: inline-block ;
1244 }
1245
1246 .Ovalbox { border-width: 2px ; }
1247
1248 .framebox {
1249     border: 1px solid black;
1250     border-radius: 0px ;
```

```
1251 padding: .3ex .2em 0ex .2em ;
1252 margin: 0pt .1ex 0pt .1ex ;
1253 display: inline-block ;
1254 }
1255
1256
1257 .mdframed {
1258 /* padding: 0ex ; */
1259 /* border: 1px solid black; */
1260 /* border-radius: 0px ; */
1261 padding: 0ex ;
1262 margin: 3ex 5% 3ex 5% ;
1263 /* display: inline-block ; */
1264 }
1265
1266 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1267
1268 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1269
1270 .mdframedtitle {
1271 padding: .5em ;
1272 display: block ;
1273 font-size: 130%
1274 }
1275
1276 .mdframedsubtitle {
1277 padding: 0ex .5em 0ex .5em ;
1278 display: block ;
1279 font-size: 115% ;
1280 }
1281
1282 .mdframedsubsubtitle {
1283 padding: 0ex .5em 0ex .5em ;
1284 display: block ;
1285 }
1286
1287 .mdtheorem {
1288 padding: 0ex .5em 0ex .5em ;
1289 margin: 3ex 5% 3ex 5% ;
1290 /* display: inline-block ; */
1291 }
1292
1293
1294 /* framed package */
1295 .framed, pre.boxedverbatim, fcolorbox {
1296 margin: 3ex 0em 3ex 0em ;
1297 border: 1px solid black;
1298 border-radius: 0px ;
1299 padding: .3ex 1em 0ex 1em ;
1300 display: block ;
```

```
1301 }
1302
1303 .snugframed {
1304     margin: 3ex 0em 3ex 0em ;
1305     border: 1px solid black;
1306     border-radius: 0px ;
1307     display: block ;
1308 }
1309
1310 .framedleftbar {
1311     margin: 3ex 0em 3ex 0em ;
1312     border-left: 3pt solid black;
1313     border-radius: 0px ;
1314     padding: .3ex .2em .3ex 1em ;
1315     display: block ;
1316 }
1317
1318 .framedtitle {
1319     margin: 0em ;
1320     padding: 0em ;
1321     font-size: 130%
1322 }
1323
1324 .framedtitle p { padding: .3em }
1325
1326
1327
1328 dl {
1329     margin: 1ex 2em 1ex 0em;
1330     line-height: 1.3;
1331 }
1332
1333 dl dt {
1334     margin-top: 1ex;
1335     margin-left: 1em ;
1336     font-weight: bold;
1337 }
1338
1339 dl dd p { margin-top: 0em; }
1340
1341
1342 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1343     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1344         "DejaVu Sans", "Bitstream Vera Sans",
1345         Geneva, Verdana, sans-serif ;
1346     margin-bottom: 4ex ;
1347 }
1348
1349 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1350     line-height: 1.2 ;
```

```
1351     margin-top:.5ex ;
1352     margin-bottom:.5ex;
1353     font-size: .9em ;
1354 }
1355
1356
1357
1358 img, img.hyperimage, img.borderimage {
1359     max-width: 600px;
1360     border: 1px solid silver;
1361     box-shadow: 3px 3px 3px #808080 ;
1362     padding: .5% ;
1363     margin: .5% ;
1364     background: none ;
1365 }
1366
1367 img.inlineimage{
1368     padding: 0px ;
1369     box-shadow: none ;
1370     border: none ;
1371     background: none ;
1372     margin: 0px ;
1373     display: inline-block ;
1374     border-radius: 0px ;
1375 }
1376
1377 img.logoimage{
1378     max-width: 300px ;
1379     box-shadow: 3px 3px 3px #808080 ;
1380     border: 1px solid black ;
1381     background:none ;
1382     padding:0 ;
1383     margin:.5ex ;
1384     border-radius: 10px ;
1385 }
1386
1387
1388 .section {
1389 /*
1390     To have each section float relative to each other:
1391 */
1392 /*
1393     display: block ;
1394     float: left ;
1395     position: relative ;
1396     background: white ;
1397     border: 1px solid silver ;
1398     padding: .5em ;
1399 */
1400     margin: 0ex .5em 0ex .5em ;
```

```
1401 padding: 0 ;
1402 }
1403
1404
1405 figure {
1406     margin: 3ex auto 3ex auto ;
1407     padding: 1ex 1em 1ex 1em ;
1408     overflow-x: auto ;
1409 }
1410
1411
1412 /* To automatically center images in figures: */
1413 /*
1414 figure img.inlineimage {
1415     margin: 0ex auto 0ex auto ;
1416     display: block ;
1417 }
1418 */
1419
1420 /* To automatically center minipages in figures: */
1421 /*
1422 figure div.minipage, figure div.minipage div.minipage {
1423     margin: 1ex auto 1ex auto ;
1424     display: block ;
1425 }
1426 */
1427
1428 figure div.minipage p { font-size: 85% ; }
1429
1430 figure.subfigure, figure.subtable {
1431     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1432 }
1433
1434 figcaption .minipage { margin:0 ; padding: 0 }
1435
1436 div.floatrow { text-align: center; }
1437
1438 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1439
1440 div.floatfoot { font-size: .85em ;
1441     border-top: 1px solid silver ; line-height: 1.2 ; }
1442
1443 figcaption , .lstlistingtitle {
1444     font-size: .85em ;
1445     text-align: center ;
1446     font-weight: bold ;
1447     margin-top: 1ex ;
1448     margin-bottom: 1ex ;
1449 }
1450
```

```
1451 figure.subfigure figcaption, figure.subtable figcaption {
1452     border-bottom: none ; background: none ;
1453 }
1454
1455 div.nonfloatcaption {
1456     margin: 1ex auto 1ex auto ;
1457     font-size: .85em ;
1458     text-align: center ;
1459     font-weight: bold ;
1460 }
1461
1462 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1463 figure div.floatrow div.minipage figcaption {
1464     border: none ;
1465     background: none ;
1466 }
1467
1468
1469 table {
1470     margin: 1ex auto 1ex auto ;
1471     border-collapse: collapse ;
1472     border-spacing: 0px ;
1473     line-height: 1.3 ;
1474     }
1475
1476 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1477     margin-bottom: 0ex ; } /* for \hline */
1478
1479 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1480     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1481
1482 td {padding: 1ex .5em 1ex .5em ;}
1483
1484 table td.tdl { text-align: left ; vertical-align: middle ; }
1485 table td.tdc { text-align: center ; vertical-align: middle ; }
1486 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1487 table td.tdbang { text-align: center ; vertical-align: middle ; }
1488 table td.tdr { text-align: right ; vertical-align: middle ; }
1489 table td.tdp { text-align: left ; vertical-align: bottom ; }
1490 table td.tdm { text-align: left ; vertical-align: middle ; }
1491 table td.tdb { text-align: left ; vertical-align: top ; }
1492 table td.tdP { text-align: center ; vertical-align: bottom ; }
1493 table td.tdM { text-align: center ; vertical-align: middle ; }
1494 table td.tdB { text-align: center ; vertical-align: top ; }
1495 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
1496     vertical-align: middle ; } /* for cmidrule */
1497 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
1498     vertical-align: middle ; }
1499 table td.tdatrule { text-align: center ; border-top: 1px solid silver ;
1500     vertical-align: middle ; padding: 0px ; margin: 0px ; }
```

```
1501 table td.tdbangrule { text-align: center ; border-top: 1px solid silver ;
1502     vertical-align: middle ; } /* for cmidrule */
1503 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
1504     vertical-align: middle ; }
1505 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
1506     vertical-align: bottom ; }
1507 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
1508     vertical-align: middle ; }
1509 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
1510     vertical-align: top ; }
1511 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1512     vertical-align: bottom ; }
1513 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
1514     vertical-align: middle ; }
1515 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1516     vertical-align: top ; }
1517
1518 /* Margins of paragraphs inside table cells: */
1519 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1520     padding-bottom: 1ex ; margin: 0ex ; }
1521 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1522     padding-bottom: 1ex ; margin: 0ex ; }
1523 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1524     padding-bottom: 1ex ; margin: 0ex ; }
1525
1526 td.tdp , td.tdprule , td.tdP , td.tdPrule
1527     { padding: 0ex .5em 0ex .5em ; }
1528 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1529     { padding: 0ex .5em 0ex .5em ; }
1530 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1531     { padding: 0ex .5em 0ex .5em ; }
1532
1533
1534 /* table notes: */
1535 .tnotes {
1536     margin: 0ex 5% 1ex 5% ;
1537     padding: 0.5ex 1em 0.5ex 1em;
1538     font-size:.85em;
1539     text-align: left ;
1540 }
1541
1542 .tnotes dl dt p {margin-bottom:0px;}
1543
1544 .tnoteitemheader {margin-right: 1em;}
1545
1546
1547
1548 /* center, flushleft, flushright environments */
1549 div.center{text-align:center;}
1550 div.center table {margin-left:auto;margin-right:auto;}
```

```
1551 div.flushleft{text-align:left;}
1552 div.flushleft table {margin-left:0em ; margin-right:auto;}
1553 div.flushright{text-align:right;}
1554 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1555
1556
1557 /* Fancybox */
1558 div.Btrivlist table tr td { padding: .2ex 0em ; }
1559
1560
1561 /* program listing callouts: */
1562 span.callout {
1563     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1564     Geneva, Verdana, sans-serif ;
1565     border-radius: .5em;
1566     background-color:black;
1567     color:white;
1568     padding:0px .25em 0px .25em;
1569     margin: 0 ;
1570     font-weight: bold;
1571     font-size:.72em ;
1572 }
1573
1574 div.programlisting pre.verbatim span.callout{
1575     font-size: .85em ;
1576 }
1577
1578
1579
1580
1581
1582 div.published
1583 {
1584     text-align: center ;
1585     font-variant: normal ;
1586     font-style: italic ;
1587     font-size: 1em ;
1588     margin: 3ex 0em 3ex 0em ;
1589 }
1590
1591 div.subtitle
1592 {
1593     text-align: center ;
1594     font-variant: normal ;
1595     font-style: italic ;
1596     font-size: 1.25em ;
1597     margin: 3ex 0em 3ex 0em ;
1598 }
1599
1600 div.subtitle p { margin: 1ex ; }
```

```
1601
1602 div.author
1603 {
1604     font-variant: normal ;
1605     font-style: normal ;
1606     font-size: 1em ;
1607     margin: 3ex 0em 3ex 0em ;
1608 }
1609
1610 div.author table {
1611     margin: 3ex auto 0ex auto ;
1612     background: none ;
1613 }
1614
1615 div.author table tbody tr td { padding: .25ex ; }
1616
1617 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1618
1619 div.titledate {
1620     text-align: center ;
1621     font-size: .85em ;
1622     font-style: italic;
1623     margin: 6ex 0em 6ex 0em ;
1624 }
1625
1626
1627 nav.topnavigation{
1628     text-align: left ;
1629     padding: 0.5ex 1em 0.5ex 1em ;
1630 /*     margin: 2ex 0em 3ex 0em ; */
1631     margin: 0 ;
1632     border-bottom: 1px solid silver ;
1633     border-top: 1px solid silver ;
1634     clear:right ;
1635 }
1636
1637 nav.botnavigation{
1638     text-align: left ;
1639     padding: 0.5ex 1em 0.5ex 1em ;
1640 /*     margin: 3ex 0em 2ex 0em ; */
1641     margin: 0 ;
1642     border-top: 1px solid silver ;
1643     border-bottom: 1px solid silver ;
1644     clear:right ;
1645 }
1646
1647
1648 header{
1649     line-height: 1.2 ;
1650     font-size: 1em ;
```

```
1651/*      border-bottom: 2px solid silver ; */
1652      margin: 0px ;
1653      padding: 0ex 1em 0ex 1em ;
1654      text-align:center ;
1655}
1656
1657header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1658
1659
1660footer{
1661      font-size: .85em ;
1662      line-height: 1.2 ;
1663      margin-top: 1ex ;
1664      border-top: 2px solid silver ;
1665      padding: 2ex 1em 2ex 1em ;
1666      clear:right ;
1667      text-align:left ;
1668}
1669
1670
1671a.linkhome { font-weight:bold ; font-size: 1em ;}
1672
1673
1674div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1675
1676img.lateximage{
1677      padding: 0px 0px 0px 0px ;
1678      box-shadow: none ;
1679      border: none ;
1680      background: none ;
1681      margin: 0px 0px -.15ex 0px ;
1682      /* pdfcrop leaves a slight margin, adjust to baseline */
1683      max-width: 100% ;
1684      border-radius: 0ex ;
1685      border: none ;
1686}
1687
1688
1689
1690nav.sidetoc {
1691      font-family: "DejaVu Serif", "Bitstream Vera Serif",
1692          "Lucida Bright", Georgia, serif;
1693      float:right ;
1694      width: 20%;
1695      border-left: 1px solid silver;
1696      border-top: 1px solid silver;
1697      border-bottom: 1px solid silver;
1698/*      border-top: 2px solid #808080 ; */
1699      background: #FAF7F4 ;
1700      padding: 2ex 0em 2ex 1em ;
```

```
1701     margin: 0ex 0em 2ex 1em ;
1702     font-size:.9em ;
1703     border-radius: 20px 0px 0px 20px ;
1704 }
1705
1706 div.sidetoccontents {
1707 /*     border-top: 1px solid silver ; */
1708     overflow-y: auto ;
1709     width: 100% ;
1710     text-align: left ;
1711 }
1712
1713
1714 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1715     text-indent: 0 ; }
1716
1717 nav.sidetoc p a {color:black ; font-size: .7em ;}
1718
1719 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1720     border-bottom: 1px solid silver ; }
1721
1722 nav.sidetoc a:hover {text-decoration: underline ; }
1723
1724
1725
1726 section.textbody { margin: 0ex 1em 0ex 1em ;}
1727
1728
1729 div.multicolsheading { -webkit-column-span: all;
1730     -moz-column-span: all; column-span: all; }
1731 div.multicols { -webkit-columns: 3 380px ;
1732     -moz-columns: 3 380px ; columns: 3 380px ; }
1733 div.multicols p {margin-top: 0ex}
1734
1735
1736
1737 /* Used to support algorithmicx: */
1738 span.floatright { float: right ; }
1739
1740
1741
1742
1743 /* Native LaTeX theorems: */
1744
1745 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1746 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1747
1748
1749 /* theorem, amsthm, and ntheorem packages */
1750
```

```
1751 span.theoremheader,
1752 span.theoremheaderplain,
1753 span.theoremheaderdefinition,
1754 span.theoremheaderbreak,
1755 span.theoremheadermarginbreak,
1756 span.theoremheaderchangebreak,
1757 span.theoremheaderchange,
1758 span.theoremheadermargin
1759 {
1760     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1761 }
1762
1763 span.amsthmnameplain,
1764 span.amsthmnamedefinition,
1765 span.amsthmnumberplain,
1766 span.amsthmnumberdefinition
1767 {
1768     font-style:normal ; font-weight: bold ;
1769 }
1770
1771
1772 span.amsthmnameremark,
1773 span.amsthmnumberremark
1774 {font-style:italic ; font-weight: normal ; }
1775
1776
1777 span.amsthmnoteplain,
1778 span.amsthmnotedefinition
1779 {font-style:normal ;}
1780
1781
1782 span.theoremheaderremark,
1783 span.theoremheaderproof,
1784 span.amsthmproofname
1785 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1786
1787 span.theoremheadersc
1788 {
1789     font-style:normal ;
1790     font-variant: small-caps ;
1791     font-weight: normal ;
1792     margin-right: 1em ;
1793 }
1794
1795 .theoremdemark {float:right}
1796
1797 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1798 div.theorembodybreak, div.theorembodynonumberbreak,
1799 div.theorembodymarginbreak,
1800 div.theorembodychangebreak,
```

```
1801 div.theorembodychange,
1802 div.theorembodymargin
1803 {
1804     font-style:italic;
1805     margin-top: 3ex ; margin-bottom: 3ex ;
1806 }
1807
1808 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1809 div.theorembodyplainupright, nonumberplainuprightsc,
1810 div.amsthmbodydefinition, div.amsthmbodyremark,
1811 div.amsthmproof
1812 {
1813     font-style: normal ;
1814     margin-top: 3ex ; margin-bottom: 3ex ;
1815 }
1816
1817 span.amsthmnoteremark {}
1818
1819
1820
1821 /*
1822 For CSS LaTeX and related logos:
1823 Based on:
1824 http://edward.oconnor.cx/2007/08/tex-poshlet
1825 http://nitens.org/taraborelli/texlogo
1826 */
1827
1828 .latexlogofont {
1829     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1830         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1831     font-variant: normal ;
1832 }
1833
1834 .latexlogo {
1835     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1836         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1837     letter-spacing: .03em ;
1838     font-size: 1.1em;
1839 }
1840
1841 .latexlogo sup {
1842     text-transform: uppercase;
1843     letter-spacing: .03em ;
1844     font-size: 0.85em;
1845     vertical-align: 0.15em;
1846     margin-left: -0.36em;
1847     margin-right: -0.15em;
1848 }
1849
1850 .latexlogo sub {
```

```
1851 text-transform: uppercase;
1852 vertical-align: -0.5ex;
1853 margin-left: -0.1667em;
1854 margin-right: -0.125em;
1855 font-size: 1em;
1856 }
1857
1858 .xetexlogo {
1859     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1860         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1861     letter-spacing: .03em ;
1862     font-size: 1.1em;
1863 }
1864
1865 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1866 .xetexlogo sub {
1867     text-transform: uppercase;
1868     vertical-align: -0.5ex;
1869     margin-left: -0.0667em;
1870     margin-right: -0.2em;
1871     font-size: 1em;
1872     letter-spacing: .03em ;
1873 }
1874
1875 /* A large gap between Xe and LaTeX v.s. TeX: */
1876 .xelatexlogo sub {
1877     text-transform: uppercase;
1878     vertical-align: -0.5ex;
1879     margin-left: -0.0667em;
1880     margin-right: -.05em;
1881     font-size: 1em;
1882     letter-spacing: .03em ;
1883 }
1884
1885 .amslogo {
1886     font-family: "TeXGyreChorus", "URW Chancery L",
1887         "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1888         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1889         "Hoefler Text", Times, "Times New Roman", serif;
1890     font-style: italic;
1891 }
1892
1893 .lyxlogo {
1894     font-family: "URW Classico", Optima, "Linux Biolinum O",
1895         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1896         Verdana, sans-serif ;
1897 }
1898
1899
1900
```

```
1901
1902 /* Only display top and bottom navigation if a small screen: */
1903 /* Hide the sidetoc if a small screen: */
1904 nav.topnavigation { display:none; }
1905 nav.botnavigation { display:none; }
1906
1907 @media screen and (max-width: 45em) {
1908 /*     nav.sidetoc {display:none;} */
1909     nav.sidetoc {
1910         float: none ;
1911         width: 100% ;
1912         margin: 5ex 0px 5ex 0px ;
1913         padding: 0 ;
1914         border-radius: 0 ;
1915         border-bottom: 1px solid black ;
1916         border-top: 1px solid black ;
1917         box-shadow: none ;
1918     }
1919 /*     nav.topnavigation { display:block } */
1920     nav.botnavigation { display:block }
1921     .marginpar {
1922         max-width: 100%;
1923         float: none;
1924         display:block ;
1925         margin: 1ex 1em 1ex 1em ;
1926     }
1927 }
1928
1929 @media print {
1930     body {
1931         font-family: "Linux Libertine O",
1932             "DejaVu Serif", "Bitstream Vera Serif",
1933             "Liberation Serif", "Nimbus Roman No 9 L",
1934             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1935     }
1936     nav.sidetoc { display:none; }
1937     nav.topnavigation { display: none; }
1938     nav.botnavigation { display: none; }
1939 }
1940
1941 @media handheld {
1942     nav.sidetoc { display:none; }
1943     nav.topnavigation { display:block }
1944     nav.botnavigation { display:block }
1945 }
1946
1947 @media projection {
1948     nav.sidetoc { display:none; }
1949     nav.topnavigation { display:block }
1950     nav.botnavigation { display:block }
```

```
1951 }
1952 \end{filecontents*}
1953 % \end{Verbatim}% for syntax highlighting
1954 \end{warpprint}
```

31.5 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.

```
1955 \begin{warpprint}
1956 \begin{filecontents*}{lwarp_sagebrush.css}
1957 @import url("lwarp.css") ;
1958
1959
1960 A:link {color:#105030 ; text-decoration: none ; }
1961 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1962 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1963 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1964
1965
1966
1967 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1968 {
1969     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1970         "Linux Libertine 0", "Liberation Serif",
1971         "Nimbus Roman No 9 L", "FreeSerif",
1972         "Hoefler Text", Times, "Times New Roman", serif;
1973     font-variant: small-caps ;
1974     font-weight: normal ;
1975     color: #304070 ;
1976     text-shadow: 2px 2px 3px #808080;
1977 }
1978
1979 h1 { /* title of the entire website, used on each page */
1980     font-variant: small-caps ;
1981     color: #304070 ;
1982     text-shadow: 2px 2px 3px #808080;
1983     background-color: #F7F7F0 ;
1984     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1985 }
1986
1987 h1 {
1988     border-bottom: 1px solid #304070;
1989     border-top: 2px solid #304070;
```

```
1990 }
1991
1992 h2 {
1993   border-bottom: 1px solid #304070;
1994   border-top: 2px solid #304070;
1995   background-color: #F7F7F0 ;
1996   background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1997 }
1998
1999
2000
2001 div.abstract {
2002   background: #f5f5eb ;
2003   background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2004
2005   border: 1px solid silver;
2006   border-radius: 1em ;
2007 }
2008
2009 div.abstract dl {line-height:1.5;}
2010 div.abstract dt {color:#304070;}
2011
2012 div.abstracttitle{
2013   font-family: "URW Classico", Optima, "Linux Biolinum 0",
2014     "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2015     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2016   font-weight:bold;
2017   font-variant: small-caps ;
2018   font-size:1.5em;
2019   border-bottom: 1px solid silver ;
2020   color: #304070 ;
2021   text-align: center ;
2022   text-shadow: 1px 1px 2px #808080;
2023 }
2024
2025 span.abstracruntintitle{
2026   font-family: "URW Classico", Optima, "Linux Biolinum 0",
2027     "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2028     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2029   font-weight:bold;
2030 }
2031
2032
2033 div.epigraph {
2034   background: #f5f5eb ;
2035   background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2036
2037   border: 1px solid silver ;
2038   border-radius: 1ex ;
2039   box-shadow: 3px 3px 3px #808080 ;
```

```
2040 }
2041
2042
2043 .example {
2044     background-color: #f5f5eb ;
2045     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2046
2047 }
2048
2049 div.exampletitle{
2050     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2051         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2052         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2053     font-weight:bold;
2054     font-variant: small-caps ;
2055     border-bottom: 1px solid silver ;
2056     color: #304070 ;
2057     text-align: center ;
2058     text-shadow: 1px 1px 2px #808080;
2059 }
2060
2061
2062 .sidebar {
2063     background-color: #f5f5eb ;
2064     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2065
2066 }
2067
2068 div.sidebar{
2069     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2070         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2071         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2072     font-weight:bold;
2073     font-variant: small-caps ;
2074     border-bottom: 1px solid silver ;
2075     color: #304070 ;
2076     text-align: center ;
2077     text-shadow: 1px 1px 2px #808080;
2078 }
2079
2080
2081 .fancyvrblabel {
2082     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2083         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2084         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2085     font-weight:bold;
2086     font-variant: small-caps ;
2087     font-size: 1.5em ;
2088     color: #304070 ;
2089     text-align: center ;
```

```
2090     text-shadow: 1px 1px 2px #808080;
2091 }
2092
2093
2094
2095 .minipage {
2096     background-color: #eeeeee7 ;
2097     border: 1px solid silver ;
2098     border-radius: 1ex ;
2099 }
2100
2101 .framed .minipage , .framedleftbar .minipage {
2102     border: none ;
2103     background: none ;
2104     padding: 0ex ;
2105     margin: 0ex ;
2106 }
2107
2108 figure.figure .minipage, figcaption .minipage { border: none; }
2109
2110 div.marginblock div.minipage { border: none; }
2111
2112 figure , div.marginblock {
2113     background-color: #eeeeee7 ;
2114     border: 1px solid silver ;
2115     border-radius: 1ex ;
2116     box-shadow: 3px 3px 3px #808080 ;
2117 }
2118
2119 figure figure {
2120     border: 1px solid silver ;
2121     margin: 0em ;
2122     box-shadow: none ;
2123 }
2124
2125 /*
2126 figcaption {
2127     border-top: 1px solid silver ;
2128     border-bottom: 1px solid silver ;
2129     background-color: #e8e8e8 ;
2130 }
2131 */
2132
2133
2134 div.table {
2135     box-shadow: 3px 3px 3px #808080 ;
2136 }
2137
2138 /*
2139 .tnotes {
```

```
2140     background: #e8e8e8;
2141     border: 1px solid silver;
2142 }
2143 */
2144
2145
2146 nav.topnavigation{
2147     background-color: #b0b8b0 ;
2148     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2149 }
2150
2151 nav.botnavigation{
2152     background-color: #b0b8b0 ;
2153     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2154 }
2155
2156
2157
2158 header{
2159     background-color: #F7F7F0 ;
2160     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2161 }
2162
2163 footer{
2164     background-color: #F7F7F0 ;
2165     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2166 }
2167
2168
2169
2170 nav.sidetoc {
2171     background-color: #F7F7F0 ;
2172     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2173     box-shadow: 3px 3px 3px #808080 ;
2174     border-radius: 0px 0px 0px 20px ;
2175     }
2176
2177 div.sidetoc title {color: #304070 ; }
2178
2179 nav.sidetoc a: hover {
2180     color:#006000 ;
2181     text-decoration: none ;
2182     text-shadow:0px 0px 2px #a0a0a0;
2183 }
2184
2185
2186 @media screen and (max-width: 45em) {
2187     nav.sidetoc { border-radius: 0 ; }
2188 }
2189
```

```
2190
2191 \end{filecontents*}
2192 % \end{Verbatim}% for syntax highlighting
2193 \end{warpprint}
```

31.6 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.

```
2194 \begin{warpprint}
2195 \begin{filecontents*}{lwarp_formal.css}
2196 @import url("lwarp.css") ;
2197
2198
2199
2200 A:link {color:#802020 ; text-decoration:none; }
2201 A:visited {color:#802020 ; text-shadow:none ;}
2202 A:hover {color:#400000 ; text-shadow:none ;}
2203 A:active {color:#C00000 ; text-shadow:none ;}
2204
2205
2206 body {
2207     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2208         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2209         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2210         "Times New Roman", serif;
2211     background: #fffcf5;
2212 }
2213
2214 span.textrm {
2215     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2216         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2217         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2218         "Times New Roman", serif;
2219 }
2220
2221 span.textsf {
2222     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2223         Geneva, Verdana, sans-serif ;
2224 }
2225
2226
2227
2228 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
```

```
2229 {
2230     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2231         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2232         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2233         "Times New Roman", serif;
2234     color: #800000 ;
2235     text-shadow: none ;
2236 }
2237
2238 h1, h2 {
2239     background-color: #fffcf5 ;
2240     background-image: none ;
2241     border-bottom: 1px solid #808080;
2242     border-top: 2px solid #808080;
2243 }
2244
2245 div.abstracttitle {
2246     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2247         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2248         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2249         "Times New Roman", serif;
2250     color: black ;
2251     text-shadow: none ;
2252 }
2253
2254 span.abstracrunintitle {
2255     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2256         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2257         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2258         "Times New Roman", serif;
2259     color: black ;
2260     text-shadow: none ;
2261 }
2262
2263 div.abstract { font-size: 100% }
2264
2265 .sidebar {
2266     background: #fffcf5;
2267     background-image: none ;
2268     margin: 2em 5% 2em 5%;
2269     padding: 0.5em 1em;
2270     border: none ;
2271     border-top : 1px solid silver;
2272     border-bottom : 1px solid silver;
2273     font-size: 90% ;
2274 }
2275
2276 div.sidebartitle{
2277     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2278         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```
2279     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2280     "Times New Roman", serif;
2281     color: #800000 ;
2282     text-shadow: none ;
2283     border: none ;
2284 }
2285
2286 .example {
2287     background: #fffcf5;
2288     background-image: none ;
2289     margin: 2em 5% 2em 5%;
2290     padding: 0.5em 1em;
2291     border: none ;
2292     border-top : 1px solid silver;
2293     border-bottom : 1px solid silver;
2294 }
2295
2296 div.exampletitle{
2297     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2298     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2299     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2300     "Times New Roman", serif;
2301     color: #800000 ;
2302     text-shadow: none ;
2303     border: none ;
2304 }
2305
2306 div.fancyvrblabel{
2307     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2308     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2309     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2310     "Times New Roman", serif;
2311     color: #800000 ;
2312     text-shadow: none ;
2313     border: none ;
2314 }
2315
2316
2317
2318 .verse {
2319     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2320     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2321     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2322     "Times New Roman", serif;
2323 }
2324
2325
2326 figure {
2327     margin: 3ex 5% 3ex 5% ;
2328     padding: 1ex 1em 1ex 1em ;
```

```
2329     background-color: #fffcf5 ;
2330     overflow-x: auto ;
2331     border: none ;
2332 /*     border-top: 1px solid silver; */
2333 /*     border-bottom: 1px solid silver; */
2334 }
2335
2336
2337 figcaption , .lstlisting {
2338     border: none ;
2339 /*     border-top: 1px solid silver ; */
2340 /*     border-bottom: 1px solid silver ; */
2341     background-color: #fffcf5 ;
2342 }
2343
2344 .tnotes {
2345     background: #fffcf5 ;
2346 }
2347
2348 .theorem {
2349     background: none ;
2350 }
2351
2352 .minipage {
2353     background-color: #fffcf5 ;
2354     border: none ;
2355 }
2356
2357 div.floatrow figure { border: none ; }
2358
2359 figure figure { border: none ; }
2360
2361
2362 nav.toc, nav.lof, nav.lot, nav.lol {
2363     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2364         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2365         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2366         "Times New Roman", serif;
2367 }
2368
2369 nav.sidetoc {
2370     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2371         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2372         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2373         "Times New Roman", serif;
2374     background-image: linear-gradient(to bottom, #fffcf5, #COCOCO);
2375     border-radius: 0px 0px 0px 20px ;
2376 }
2377
2378 div.sidetoc{title{
```

```

2379     color: #800000 ;
2380 }
2381
2382 header{
2383     background-color: #e0e0e0 ;
2384     background-image: linear-gradient(to top, #ffcf5, #b0b0b0);
2385     text-align:center ;
2386 }
2387
2388 footer{
2389     background-color: #e0e0e0 ;
2390     background-image: linear-gradient(to bottom, #ffcf5, #b0b0b0);
2391     padding: 2ex 1em 2ex 1em ;
2392     clear:right ;
2393     text-align:left ;
2394 }
2395
2396 nav.botnavigation {
2397     background: #dedcd5 ;
2398     border-top: 1px solid black ;
2399 }
2400 \end{filecontents*}
2401 % \end{Verbatim}% for syntax highlighting
2402 \end{warpprint}

```

31.7 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.

```

2403 \begin{warpprint}
2404 \begin{filecontents*}{sample_project.css}
2405 /* ( --- Start of project.css --- ) */
2406 /* A sample project-specific CSS file for lwarp --- ) */
2407
2408 /* Load default lwarp settings: */
2409 @import url("lwarp.css") ;
2410 /* or lwarp_formal.css, lwarp_sagebrush.css */
2411
2412 /* Project-specific CSS setting follow here. */
2413 /* . . . */
2414
2415 /* ( --- End of project.css --- ) */
2416 \end{filecontents*}
2417 % \end{Verbatim}% for syntax highlighting

```

```
2418 \end{warpprint}
```

31.8 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.

```
2419 \begin{warpprint}
2420 \begin{filecontents*}{lwarp.xdy}
2421 (require "tex/inputenc/latin.xdy")
2422 (merge-rule "\\PS *" "Postscript")
2423 (require "texindy.xdy")
2424 (require "page-ranges.xdy")
2425 (require "book-order.xdy")
2426 (markup-locref :open "\hyperindexref{" :close "}")
2427 \end{filecontents*}
2428 % \end{Verbatim}% for syntax highlighting
2429 \end{warpprint}
```

31.9 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.

```
2430 \begin{warpprint}
2431 \begin{filecontents*}{lwarp_mathjax.txt}
2432 <!-- https://groups.google.com/forum/#!topic/
2433         mathjax-users/jUteWUcE2bY -->
2434 <script type="text/x-mathjax-config">
2435 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2436     var seteqsectionDefault = {name: "", num: 0};
2437     var seteqsections = {}, seteqsection = seteqsectionDefault;
2438     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2439     var AMS = MathJax.Extension["TeX/AMSmath"];
2440     TEX.Definitions.Add({
2441     macros: {
2442         seteqsection: "mySection",
2443         seteqnumber: "mySetEqNumber"
2444     }
2445     });
```

```
2446
2447 PARSE.Augment({
2448   mySection: function (name) {
2449     seteqsection.num = AMS.number;
2450     var n = this.GetArgument(name);
2451     if (n === "") {
2452       seteqsection = seteqsectionDefault;
2453     } else {
2454       if (!seteqsections["_"+n])
2455         seteqsections["_"+n] = {name:n, num:0};
2456       seteqsection = seteqsections["_"+n];
2457     }
2458     AMS.number = seteqsection.num;
2459   },
2460   mySetEqNumber: function (name) {
2461     var n = this.GetArgument(name);
2462     if (!n || !n.match(/^[0-9]+ *$/))
2463       n = ""; else n = parseInt(n)-1;
2464     <!-- $ syntax highlighting -->
2465     if (n === "" || n < 1)
2466       TEX.Error
2467         ("Argument to "+name+" should be a positive integer");
2468     AMS.number = n;
2469   }
2470 });
2471 MathJax.Hub.Config({
2472   TeX: {
2473     equationNumbers: {
2474       formatTag: function (n)
2475         {return "("+(seteqsection.name+"."+n).replace(/\./, "")+"")"},
2476       formatID: function (n) {
2477         n = (seteqsection.name+'.'+n).replace
2478           (/[:'<>&]/g, "").replace(/\./, "");
2479         return 'mjx-eqn-' + n;
2480       }
2481     }
2482   }
2483 });
2484 });
2485 </script>
2486
2487 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2488 <script type="text/x-mathjax-config">
2489   MathJax.Ajax.config.path["Contrib"] =
2490     "https://cdn.mathjax.org/mathjax/contrib";
2491 </script>
2492
2493 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2494         tree/master/siunitx -->
2495 <script type="text/x-mathjax-config">
```

```

2496 MathJax.Hub.Config({
2497   extensions: ["tex2jax.js","[Contrib]/siunitx/siunitx.js"],
2498   jax: ["input/TeX","output/HTML-CSS"],
2499   tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
2500   TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
2501 });
2502 </script>
2503
2504 <script type="text/x-mathjax-config">
2505 MathJax.Hub.Config({
2506   TeX: {
2507     equationNumbers: {
2508       autoNumber: "AMS"
2509     }
2510   }
2511 });
2512 </script>
2513
2514 <!-- Alternative CDN provider: -->
2515 <script type="text/javascript" async
2516 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2517 </script>
2518
2519 <!-- No longer supported after April 30, 2017: -->
2520 <!--
2521 <script
2522   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2523 </script>
2524 -->
2525
2526 \end{filecontents*}
2527 % \end{Verbatim}% for syntax highlighting
2528 \end{warpprint}

```

31.10 lwarpmk option

The following is only generated if the `lwarpmk` option was given to `lwarp`.

```

2529 \begin{LWR@createlwarpmk}

```

Prog `lwarpmk` Creates a local copy of `lwarpmk`:

```

2530 \begin{filecontents*}{lwarpmk.lua}
2531 #!/usr/bin/env texlua
2532
2533 -- Copyright 2016-2017 Brian Dunn
2534

```

```
2535 -- Print the usage of the lwarpmk command:
2536
2537 printversion = "v0.38"
2538
2539 function printhelp ()
2540 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2541 end
2542
2543 function printusage ()
2544 print ( [[
2545
2546 lwarpmk print [project]: Compile the print version if necessary.
2547 lwarpmk print1 [project]: Forced single compile of the print version.
2548 lwarpmk printindex [project]: Process the index for the print version.
2549 lwarpmk printglossary [project]: Process the glossary for the print version.
2550 lwarpmk html [project]: Compile the HTML version if necessary.
2551 lwarpmk html1 [project]: Forced single compile of the HTML version.
2552 lwarpmk htmlindex [project]: Process the index for the html version.
2553 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2554 lwarpmk again [project]: Touch the source code to trigger recompiles.
2555 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2556 lwarpmk pdftohtml [project]:
2557     For use with latexmk or a Makefile:
2558     Convert project_html.pdf to project_html.html and
2559     individual HTML files.
2560 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2561 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2562 lwarpmk -h: Print this help message.
2563 lwarpmk --help: Print this help message.
2564
2565 ]] )
2566 printconf ()
2567 end
2568
2569 -- Print the format of the configuration file lwarpmk.conf:
2570
2571 function printconf ()
2572 print ( [[
2573 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2574 --
2575 opsystem = "Unix"    (or "Windows")
2576 latexname = "pdflatex" (or "lualatex", or "xelatex")
2577 sourcename = "projectname" (the source-code filename w/o .tex)
2578 homehtmlfilename = "index" (or perhaps the project name)
2579 htmlfilename = "" (or "projectname" - filename prefix)
2580 latexmk = "false" (or "true" to use latexmk to build PDFs)
2581 language = "english" (use a language supported by xindy)
2582 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2583 --
2584 Filenames must contain only letters, numbers, underscore, or dash.
```

```
2585 Values must be in "quotes".
2586
2587 ]] ) ;
2588 end
2589
2590
2591 -- Split one large sourcefile into a number of files,
2592 -- starting with destfile.
2593 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2594
2595 function splitfile (destfile,sourcefile)
2596 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2597 local sfile = io.open(sourcefile)
2598 io.output(destfile)
2599 for line in sfile:lines() do
2600 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|)") ;
2601 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2602 io.output(newfilename) ;
2603 else -- not a splitpoint
2604 io.write (line .. "\n") ;
2605 end
2606 end -- do
2607 io.close(sfile)
2608 end -- function
2609
2610 -- Incorrect value, so print an error and exit.
2611
2612 function cvalueerror ( line, linenum , cvalue )
2613     print ( linenum .. " : " .. line ) ;
2614     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2615     printconf () ;
2616     os.exit(1) ;
2617 end
2618
2619 -- Load settings from the project's "lwarpmk.conf" file:
2620
2621 function loadconf ()
2622 -- Default configuration filename:
2623 local conffile = "lwarpmk.conf"
2624 -- Optional configuration filename:
2625 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2626 -- Default language:
2627 language = "english"
2628 -- Default xdyfile:
2629 xdyfile = "lwarp.xdy"
2630 -- Verify the file exists:
2631 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2632 print("lwarpmk: " .. conffile .. " does not exist.")
2633 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2634 printhelp () ;
```

```
2635 os.exit(1) -- exit the entire lwarpmk script
2636 else -- file exists
2637 -- Read the file:
2638 print ("lwarpmk: Reading " .. conffile .. ".")
2639 local cfile = io.open(conffile)
2640 -- Scan each line:
2641 local linenum = 0
2642 for line in cfile:lines() do -- scan lines
2643 linenum = linenum + 1
2644 i,j,cvarname,cvalue = string.find (line,"([%w-_*]%s*=%s*\"([%w%-_%.]*)\"");
2645 -- Error if incorrect enclosing characters:
2646 if ( i == nil ) then
2647 print ( linenum .. " : " .. line );
2648 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" );
2649 printconf ( );
2650 os.exit(1) ;
2651 end
2652 if ( cvarname == "opssystem" ) then
2653 -- Verify choice of opssystem:
2654 if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2655 opssystem = cvalue
2656 else
2657 cvalueerror ( line, linenum , cvalue )
2658 end
2659 elseif ( cvarname == "latexname" ) then
2660 -- Verify choice of LaTeX compiler:
2661 if (
2662 (cvalue == "pdflatex") or
2663 (cvalue == "xelatex") or
2664 (cvalue == "lualatex")
2665 ) then
2666 latexname = cvalue
2667 else
2668 cvalueerror ( line, linenum , cvalue )
2669 end
2670 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2671 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2672 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2673 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2674 elseif ( cvarname == "language" ) then language = cvalue
2675 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2676 else
2677 print ( linenum .. " : " .. line );
2678 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" );
2679 printconf ( );
2680 os.exit(1) ;
2681 end
2682 end -- do scan lines
2683 io.close(cfile)
2684 end -- file exists
```

```
2685 -- Select some operating-system commands:
2686 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2687   rmname = "rm"
2688   mvname = "mv"
2689   touchnamepre = "touch"
2690   touchnamepost = ""
2691   dirslash = "/"
2692   opquote= "\'"
2693 elseif opsystem=="Windows" then -- For Windows
2694   rmname = "DEL"
2695   mvname = "MOVE"
2696   touchnamepre = "COPY /b"
2697   touchnamepost = "+,,"
2698   dirslash = "\\"
2699   opquote= "\""
2700 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2701 end --- for Windows
2702
2703 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2704 if ( latexname == "pdflatex" ) then
2705   xindycmd = "texindy -C utf8"
2706   glossarycmd = "xindy -C utf8"
2707 else
2708   xindycmd = "xindy -M texindy -C utf8"
2709   glossarycmd = "xindy -C utf8"
2710 end
2711
2712 end -- loadconf
2713
2714
2715 function refreshdate ()
2716 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2717 end
2718
2719
2720 -- Scan the LaTeX log file for the phrase "Rerun to get",
2721 -- indicating that the file should be compiled again.
2722 -- Return true if found.
2723
2724 function reruntoget (filesorce)
2725 local fsource = io.open(filesorce)
2726 for line in fsource:lines() do
2727   if ( string.find(line,"Rerun to get") ~= nil ) then
2728     io.close(fsource)
2729     return true
2730   end
2731 end
2732 io.close(fsource)
2733 return false
2734 end
```

```
2735
2736
2737-- Compile one time, return true if should compile again.
2738-- fsuffix is "" for print, "_html" for HTML output.
2739
2740function onetime (fsuffix)
2741print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2742err = os.execute(
2743--   "echo " ..
2744   latexname .. " " .. sourcename..fsuffix )
2745if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2746return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2747end
2748
2749
2750-- Compile up to five times.
2751-- fsuffix is "" for print, "_html" for HTML output
2752
2753function manytimes (fsuffix)
2754if onetime(fsuffix) == true then
2755if onetime(fsuffix) == true then
2756if onetime(fsuffix) == true then
2757if onetime(fsuffix) == true then
2758if onetime(fsuffix) == true then
2759end end end end end
2760end
2761
2762-- Exit if the given file does not exist.
2763
2764function verifyfileexists (filename)
2765if (lfs.attributes ( filename , "modification" ) == nil ) then
2766print ( "lwarpmk: " .. filename .. " not found." ) ;
2767os.exit (1) ;
2768end
2769end
2770
2771
2772-- Convert <project>_html.pdf into HTML files:
2773
2774function pdftohtml ()
2775   -- Convert to text:
2776   print ("lwarpmk: Converting " .. sourcename
2777         .. "_html.pdf to " .. sourcename .. "_html.html")
2778   os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2779             .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2780   -- Split the result into individual HTML files:
2781   splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2782end
2783
2784
```

```
2785 -- Remove auxiliary files:
2786
2787 function removeaux ()
2788     os.execute ( rmname .. " " ..
2789         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2790         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2791         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2792         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2793         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2794         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2795         sourcename .. ".log " .. sourcename .. "_html.log " ..
2796         sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
2797         "*_html_inc.*"
2798     )
2799 end
2800
2801
2802
2803 -- Create lateximages based on lateximages.txt:
2804 function createlateximages ()
2805 print ("lwarpmk: Creating lateximages.")
2806 local limagesfile = io.open("lateximages.txt")
2807 -- Create the lateximages directory, ignore error if already exists
2808 err = os.execute("mkdir lateximages")
2809 -- Scan lateximages.txt
2810 for line in limagesfile:lines() do
2811 -- lwimgpage is the page number in the PDF which has the image
2812 -- lwimgnum is the sequential lateximage number to assign for the image
2813 i,j,lwimgpage,lwimgnum = string.find (line,"|(.)|(.)|")
2814 -- For each entry:
2815 if ( i~=nil ) then
2816 -- Separate out the image into its own single-page pdf:
2817 err = os.execute(
2818 "pdfseparate -f " .. lwimgpage .. " -1 " ..
2819 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2820 -- Crop the image:
2821 err = os.execute(
2822 "pdfcrop --hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2823 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2824 -- Convert the image to svg:
2825 err = os.execute(
2826 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2827 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2828 -- Move the result into lateximages/:
2829 err = os.execute(
2830 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2831 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2832 -- Remove the temporary files:
2833 err = os.execute(
2834 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
```

```
2835 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2836 end
2837 end -- do
2838 io.close(limagesfile)
2839 end -- function
2840
2841
2842 -- Use latexmk to compile source and index:
2843 -- fsuffix is "" for print, or "_html" for HTML
2844 function compilelatexmk ( fsuffix )
2845     -- The recorder option is required to detect changes in <project>.tex
2846     -- while we are loading <project>_html.tex.
2847     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2848         .. "-e "
2849         .. opquote
2850         .. "$makeindex = q/" -- $
2851         .. xindycmd
2852         .. " -M " .. xdyfile
2853         .. " -L " .. language .. " /"
2854         .. opquote
2855         .. "-pdflatex=\\" .. latexname .. "%O %S\" "
2856         .. sourcename..fsuffix ..".tex" ) ;
2857     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2858 end
2859
2860
2861
2862 -- lwarpmk --version :
2863
2864 if (arg[1] == "--version") then
2865 print ( "lwarpmk: " .. printversion )
2866
2867 else -- not --version
2868
2869 -- print intro:
2870
2871 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2872
2873 -- lwarpmk print:
2874
2875 if arg[1] == "print" then
2876 loadconf ( )
2877 if ( latexmk == "true" ) then
2878     compilelatexmk ( "" )
2879     print ("lwarpmk: Done.")
2880 else -- not latexmk
2881     verifyfileexists (sourcename .. ".tex") ;
2882     -- See if up to date:
2883     if (
2884         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
```

```
2885     (
2886         lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2887         lfs.attributes ( sourcename .. ".pdf" , "modification" )
2888     )
2889 ) then
2890     -- Recompile if not yet up to date:
2891     manytimes("")
2892     print ("lwarpmk: Done." ) ;
2893 else
2894     print ("lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
2895 end
2896 end -- not latexmk
2897
2898 elseif arg[1] == "print1" then
2899     loadconf ()
2900     verifyfileexists (sourcename .. ".tex") ;
2901     onetime("")
2902     print ("lwarpmk: Done." ) ;
2903
2904 -- lwarp printindex:
2905 -- Compile the index then touch the source
2906 -- to trigger a recompile of the document:
2907
2908 elseif arg[1] == "printindex" then
2909 loadconf ()
2910 print ("lwarpmk: Processing the index.")
2911 os.execute(
2912     xindycmd
2913     .. " -M " .. xdyfile
2914     .. " -L " .. language
2915     .. " " .. sourcename .. ".idx")
2916 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2917 refreshdate ()
2918 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2919 print ("lwarpmk: Done.")
2920
2921 -- lwarp printglossary:
2922 -- Compile the glossary then touch the source
2923 -- to trigger a recompile of the document:
2924
2925 elseif arg[1] == "printglossary" then
2926 loadconf ()
2927 print ("lwarpmk: Processing the glossary.")
2928
2929 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2930     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2931     .. sourcename .. ".glo")
2932 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2933 refreshdate ()
2934 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
```

```
2935 print ("lwarpmk: Done.")
2936
2937 -- lwarpmk html:
2938
2939 elseif arg[1] == "html" then
2940 loadconf ()
2941 if ( latexmk == "true" ) then
2942   compilelatexmk ("_html")
2943   pdftohtml ()
2944   print ("lwarpmk: Done.")
2945 else -- not latexmk
2946   verifyfileexists ( sourcename .. ".tex" ) ;
2947   -- See if exists and is up to date:
2948   if (
2949     ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2950     (
2951       lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2952       lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2953     )
2954   ) then
2955     -- Recompile if not yet up to date:
2956     manytimes("_html")
2957     pdftohtml ()
2958     print ("lwarpmk: Done.")
2959   else
2960     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2961   end
2962 end -- not latexmk
2963
2964 elseif arg[1] == "html1" then
2965   loadconf ()
2966   verifyfileexists ( sourcename .. ".tex" ) ;
2967   onetime("_html")
2968   pdftohtml ()
2969   print ("lwarpmk: Done.")
2970
2971 elseif arg[1] == "pdftohtml" then
2972   loadconf ()
2973   pdftohtml ()
2974
2975 -- lwarpmk htmlindex:
2976 -- Compile the index then touch the source
2977 -- to trigger a recompile of the document:
2978
2979 elseif arg[1] == "htmlindex" then
2980 loadconf ()
2981 print ("lwarpmk: Processing the index.")
2982 os.execute(
2983   xindycmd
2984   .. " -M " .. xdyfile
```

```
2985 .. " -L " .. language
2986 .. " " .. sourcename .. "_html.idx"
2987 )
2988 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2989 refreshdate ()
2990 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2991 print ("lwarpmk: Done.")
2992
2993 -- lwarpmk htmlglossary:
2994 -- Compile the glossary then touch the source
2995 -- to trigger a recompile of the document:
2996
2997 elseif arg[1] == "htmlglossary" then
2998 loadconf ()
2999 print ("lwarpmk: Processing the glossary.")
3000
3001 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3002   "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3003   "_html.gls " .. sourcename .. "_html.glo")
3004
3005 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3006 refreshdate ()
3007 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3008 print ("lwarpmk: Done.")
3009
3010 -- lwarpmk limages:
3011 -- Scan the lateximages.txt file to create lateximages,
3012 -- then touch the source to trigger a recompile.
3013
3014 elseif arg[1] == "limages" then
3015 loadconf ()
3016 print ("lwarpmk: Processing images.")
3017 createlateximages ()
3018 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3019 refreshdate ()
3020 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3021 print ("lwarpmk: Done.")
3022
3023 -- lwarpmk again:
3024 -- Touch the source to trigger a recompile.
3025
3026 elseif arg[1] == "again" then
3027 loadconf ()
3028 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3029 refreshdate ()
3030 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3031 print ("lwarpmk: Done.")
3032
3033 -- lwarpmk clean:
3034 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
```

```

3035
3036 elseif arg[1] == "clean" then
3037 loadconf ()
3038 removeaux ()
3039 print ("lwarpmk: Done.")
3040
3041 -- lwarpmk cleanall
3042 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3043 -- and also project.pdf, *.html
3044
3045 elseif arg[1] == "cleanall" then
3046 loadconf ()
3047 removeaux ()
3048 os.execute ( rmname .. " " ..
3049 sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3050 "*.html"
3051 )
3052 print ("lwarpmk: Done.")
3053
3054 -- lwarpmk with no argument :
3055
3056 elseif (arg[1] == nil) then
3057 printhelp ()
3058
3059 -- lwarpmk -h or lwarpmk --help :
3060
3061 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3062 printusage ()
3063
3064 else
3065 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
3066 printhelp ()
3067 end
3068
3069 end -- not --version
3070 \end{filecontents*}
3071 % \end{Verbatim}% for syntax highlighting

3072 \end{LWR@createlwarpmk}

```

32 Stacks

for HTML output: 3073 \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.

32.1 Assigning depths

initial depths for empty stack entries:

```
3074 \newcommand*\LWR@depthnone}{-5}
```

all sectioning depths are deeper than `LWR@depthfinished`:

```
3075 \newcommand*\LWR@depthfinished}{-4}
3076 \newcommand*\LWR@depthpart}{-1}
3077 \newcommand*\LWR@depthchapter}{0}
3078 \newcommand*\LWR@depthsection}{1}
3079 \newcommand*\LWR@depthsubsection}{2}
3080 \newcommand*\LWR@depthsubsubsection}{3}
3081 \newcommand*\LWR@depthparagraph}{4}
3082 \newcommand*\LWR@depthsubparagraph}{5}
```

used by `\itemize`, `\enumerate`, `\description`:

```
3083 \newcommand*\LWR@depthlist}{6}
```

used by `\item`:

```
3084 \newcommand*\LWR@depthlistitem}{7}
```

32.2 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.

```
3085 \newcommand*\LWR@closeone}{}% top of the stack
3086 \newcommand*\LWR@closetwo}{%
3087 \newcommand*\LWR@closethree}{%
```

```

3088 \newcommand*\LWR@closefour}{ }
3089 \newcommand*\LWR@closefive}{ }
3090 \newcommand*\LWR@closesix}{ }
3091 \newcommand*\LWR@closeseven}{ }
3092 \newcommand*\LWR@closeeight}{ }
3093 \newcommand*\LWR@closenine}{ }
3094 \newcommand*\LWR@closeten}{ }
3095 \newcommand*\LWR@closeeleven}{ }
3096 \newcommand*\LWR@closetwelve}{ }

```

32.3 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}

```

```

3097 \newcommand*\LWR@closedepthone{\LWR@depthnone}% top of the stack
3098 \newcommand*\LWR@closedepthtwo{\LWR@depthnone}
3099 \newcommand*\LWR@closedepththree{\LWR@depthnone}
3100 \newcommand*\LWR@closedepthfour{\LWR@depthnone}
3101 \newcommand*\LWR@closedepthfive{\LWR@depthnone}
3102 \newcommand*\LWR@closedepthsix{\LWR@depthnone}
3103 \newcommand*\LWR@closedepthseven{\LWR@depthnone}
3104 \newcommand*\LWR@closedeptheight{\LWR@depthnone}
3105 \newcommand*\LWR@closedepthnine{\LWR@depthnone}
3106 \newcommand*\LWR@closedephten{\LWR@depthnone}
3107 \newcommand*\LWR@closedeptheleven{\LWR@depthnone}
3108 \newcommand*\LWR@closedephtwelve{\LWR@depthnone}

```

32.4 Pushing and popping the stack

```
\pushclose {<action>} {<depth>}
```

Pushes one return action and its \LaTeX depth onto the stacks.

```

3109 \NewDocumentCommand{\pushclose}{m m}
3110 {
3111 \let\LWR@closeone\let\LWR@closeone#1
3112 \let\LWR@closeone\let\LWR@closeone#1
3113 \let\LWR@closeone\let\LWR@closeone#1
3114 \let\LWR@closeone\let\LWR@closeone#1
3115 \let\LWR@closeone\let\LWR@closeone#1
3116 \let\LWR@closeone\let\LWR@closeone#1
3117 \let\LWR@closeone\let\LWR@closeone#1
3118 \let\LWR@closeone\let\LWR@closeone#1
3119 \let\LWR@closeone\let\LWR@closeone#1
3120 \let\LWR@closeone\let\LWR@closeone#1
3121 \let\LWR@closeone\let\LWR@closeone#1
3122 \let\LWR@closeone\let\LWR@closeone#1
3123 \let\LWR@closeone\let\LWR@closeone#1
3124 \let\LWR@closeone\let\LWR@closeone#1
3125 \let\LWR@closeone\let\LWR@closeone#1
3126 \let\LWR@closeone\let\LWR@closeone#1
3127 \let\LWR@closeone\let\LWR@closeone#1
3128 \let\LWR@closeone\let\LWR@closeone#1
3129 \let\LWR@closeone\let\LWR@closeone#1
3130 \let\LWR@closeone\let\LWR@closeone#1
3131 \let\LWR@closeone\let\LWR@closeone#1
3132 \let\LWR@closeone\let\LWR@closeone#1
3133 \let\LWR@closeone\let\LWR@closeone#1
3134 \let\LWR@closeone\let\LWR@closeone#1
3135 }

```

`\popclose` Pops one action and its depth off the stacks.

```

3136 \newcommand*{\popclose}
3137 {
3138 \let\LWR@closeone\let\LWR@closeone#1
3139 \let\LWR@closeone\let\LWR@closeone#1
3140 \let\LWR@closeone\let\LWR@closeone#1
3141 \let\LWR@closeone\let\LWR@closeone#1
3142 \let\LWR@closeone\let\LWR@closeone#1
3143 \let\LWR@closeone\let\LWR@closeone#1
3144 \let\LWR@closeone\let\LWR@closeone#1
3145 \let\LWR@closeone\let\LWR@closeone#1
3146 \let\LWR@closeone\let\LWR@closeone#1
3147 \let\LWR@closeone\let\LWR@closeone#1
3148 \let\LWR@closeone\let\LWR@closeone#1
3149 \let\LWR@closeone\let\LWR@closeone#1
3150 \let\LWR@closeone\let\LWR@closeone#1
3151 \let\LWR@closeone\let\LWR@closeone#1
3152 \let\LWR@closeone\let\LWR@closeone#1

```

```

3153 \let\LWR@closedepthfive\LWR@closedepthsix
3154 \let\LWR@closedepthsix\LWR@closedepthseven
3155 \let\LWR@closedepthseven\LWR@closedeptheight
3156 \let\LWR@closedeptheight\LWR@closedepthnine
3157 \let\LWR@closedepthnine\LWR@closedepthten
3158 \let\LWR@closedepthten\LWR@closedeptheleven
3159 \let\LWR@closedeptheleven\LWR@closedephtwelve
3160 }

3161 \end{warpHTML}

```

33 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: 3162 `\begin{warpHTML}`

```
\LWR@setexparray {<name>} {<index>} {<contents>}
```

```

3163 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3164 \ifthenelse{\isempty{#3}}{%
3165 {\csdef{#1#2}{}}
3166 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3167 }

```

```
\LWR@getexparray {<name>} {<index>}
```

```
3168 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
3169 \end{warpHTML}
```

34 HTML entities

for HTML output: 3170 `\begin{warpHTML}`

HTML entites and HTML Unicode entities:

```
3171 \let\LWR@origampersand\&
```

```
\HTMLentity {<entitytag>}
```

```
3172 \newcommand*\HTMLentity}[1]{%
3173 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3174 \begingroup%
3175 \LWR@FBcancel%
3176 \LWR@origampersand#1;%
3177 \endgroup
3178 % \LWR@traceinfo{HTMLentity done}%
3179 }
```

```
\HTMLunicode {<hex_unicode>}
```

```
3180 \newcommand*\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

```
\&
```

```
3181 \renewcommand*\&{\HTMLentity{amp}}
```

```
\textless
```

```
\textgreater
```

```
3182 \let\LWR@origtextless\textless
3183 \renewcommand*\textless{\HTMLentity{lt}}
3184
3185 \let\LWR@origtextgreater\textgreater
3186 \renewcommand*\textgreater{\HTMLentity{gt}}
```

```
3187 \end{warpHTML}
```

35 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: 3188 \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.

```
3189 \providecommand*\BaseJobname{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 5.7.

```
3190 \providecommand*\HTMLFilename{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 5.7.

```
3191 \providecommand*\HomeHTMLFilename{\BaseJobname}
```

`\SetHTMLFileNumber` $\{\langle number \rangle\}$

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.

```
3192 \newcommand*\SetHTMLFileNumber[1]{%
3193 \setcounter{LWR@htmlfilenumber}{#1}%
3194 }
```

Bool `FileSectionNames` Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
3195 \newbool{FileSectionNames}
3196 \booltrue{FileSectionNames}
```

```
3197 \end{warpall}
```

for HTML output: 3198 `\begin{warpHTML}`

Ctrl `LWR@htmlfilenumber` Records the number of each HTML file as it is being created. Number 0 is the home page.

```
3199 \newcounter{LWR@htmlfilenumber}
3200 \setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` $\{\langle htmlfilenumber \text{ or } name \rangle\}$

Prints the filename for a given section: `\HTMLFilename{}-filenumber/name.html`

```
3201 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3202 \LWR@traceinfo{\LWR@htmlsectionfilename A}%
```

Section 0 or empty is given the home filename. The filename must be detokenized for underscores.

```
3203 \LWR@traceinfo{about to assign temp}%
3204 \edef\LWR@tempone{#1}%
3205 \LWR@traceinfo{about to compare with ??}%
3206 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3207 {\LWR@traceinfo{found ??}}%
3208 {\LWR@traceinfo{not found ??}}%
3209 \LWR@traceinfo{about to compare with zero or empty}%
3210 \ifthenelse{%
3211   \equal{\LWR@tempone}{0}%
3212   \OR \equal{\LWR@tempone}{}}%
3213   \OR \equal{\LWR@tempone}{??}}%
3214 }%
3215 {%
3216   \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3217   \HomeHTMLFilename.html%
3218 }%
```

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`:

```
3219 {%
3220   \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
3221   \ifthenelse{%
3222     \equal{\HTMLFilename}{} \AND
3223     \equal{\LWR@tempone}{Index} \OR
3224     \equal{\LWR@tempone}{index}}%
3225   }%
3226   {%
3227     \LWR@traceinfo{prefixing the index name with an underscore.}%
3228     \_#1.html%
3229   }%
```

Otherwise, create a filename with the chosen prefix:

```
3230   {\HTMLFilename#1.html}%
3231 }%
3232 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
3233 }
```

`\LWR@htmlrefsectionfilename` $\{ \langle label \rangle \}$

Prints the filename for the given label

```

3234 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3235 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
3236 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}}%
3237 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
3238 }

3239 \end{warpHTML}

```

36 Homepage link

for HTML output: 3240 \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.

```

3241 \newcommand*{\LinkHome}{%
3242 \LWR@subhyperrefclass{%
3243 \HomeHTMLFilename.html}%
3244 {Home}{linkhome}%
3245 }

```

`\LWR@topnavigation` Creates a link to the homepage at the top of the page for use when the window is too narrow for the sideroc.

```

3246 \newcommand*{\LWR@topnavigation}{
3247 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3248 }

```

`\LWR@botnavigation` Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sideroc.

```

3249 \newcommand*{\LWR@botnavigation}{
3250 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
3251 }

```

```

3252 \end{warpHTML}

```

37 \PrintStack diagnostic tool



Diagnostics tool: Prints the \TeX nesting depth values for the stack levels. Must have \LWR@startpars active while printing the stack, so \PrintStack may be called from anywhere in the normal text flow.

for HTML output: 3253 $\text{\begin{warpHTML}}$

\PrintStack Prints the closedepth stack.

```
3254 \newcommand*{\PrintStack}{
3255 \LWR@startpars
3256 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3257 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3258 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3259 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3260 }
```

```
3261 \end{warpHTML}
```

38 Closing stack levels

for HTML output: 3262 $\text{\begin{warpHTML}}$

Close one nested level:

```
3263 \newcommand*{\LWR@closeoneprevious}{%
3264
3265 \LWR@closeone{}
3266
3267 \popclose{}
3268 }
```

$\text{\LWR@closeprevious}$ $\{ \langle depth \rangle \}$ Close everything up to the given depth:

```
3269 \newcommand*{\LWR@closeprevious}[1]{
```

Close any pending paragraph:

```
3270 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
3271 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
3272 }
```

```
3273 \end{warpHTML}
```

39 PDF pages and styles

for HTML output: 3274 \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.

```
3275 \newcommand{\LWR@forcenewpage}{%
3276 \LWR@stoppars\LWR@originewpage\LWR@startpars%
3277 }
```

`\pagestyle`, etc. are nullified for HTML output.

`\pagestyle` $\langle style \rangle$

```
3278 \renewcommand*\pagestyle[1]{}
```

`\thispagestyle` $\langle style \rangle$

```
3279 \renewcommand*\thispagestyle[1]{}
```

`\markboth` $\langle left \rangle$ $\langle right \rangle$

```
3280 \renewcommand*\markboth[2]{}
```

`\markright` $\langle right \rangle$

```
3281 \renewcommand*\markright[1]{}
```

`\raggedbottom`

```
3282 \renewcommand*\raggedbottom{}
```

`\flushbottom`

```
3283 \renewcommand*{\flushbottom}{}
```

`\sloppy`

```
3284 \renewcommand*{\sloppy}{}
```

`\fussy`

```
3285 \renewcommand*{\fussy}{}
```

`\pagenumbering` $\{(\textit{commands})\}$

```
3286 \renewcommand*{\pagenumbering}[1]{}

```

```
3287 \end{warpHTML}
```

40 HTML tags, spans, divs, elements

for HTML output: `3288 \begin{warpHTML}`

40.1 Mapping L^AT_EX Sections to HTML Sections

```
3289 \newcommand*{\LWR@tagpart}{h2}
3290 \newcommand*{\LWR@tagpartend}{/h2}
3291 \newcommand*{\LWR@tagchapter}{h3}
3292 \newcommand*{\LWR@tagchapterend}{/h3}
3293 \newcommand*{\LWR@tagsection}{h4}
3294 \newcommand*{\LWR@tagsectionend}{/h4}
3295 \newcommand*{\LWR@tagsubsection}{h5}
3296 \newcommand*{\LWR@tagsubsectionend}{/h5}
3297 \newcommand*{\LWR@tagsubsubsection}{h6}
3298 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3299 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{} }
3300 \newcommand*{\LWR@tagparagraphend}{/span}
3301 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"{} }
3302 \newcommand*{\LWR@tagsubparagraphend}{/span}
3303
3304 \newcommand*{\LWR@tagregularparagraph}{p}
```

40.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdf_latex and xelatex.

(Based on original code by DANIEL FLIPO.)

```

3305 \providecommand*\LWR@FBcancel{}{}
3306
3307 \AtBeginDocument{
3308 \@ifundefined{frenchbsetup}
3309 {}
3310 {
3311   \frenchbsetup{FrenchFootnotes=false}
3312   % OR: redefine \insertfootnotemarkFB?
3313   \LetLtxMacro\LWR@FBcancel\NoAutoSpacing
3314   \renewcommand*\FBcolonspace}{%
3315     \begingroup%
3316     \LWR@FBcancel%
3317     \LWR@origampersand{}nbsp;%
3318     \endgroup%
3319   }
3320   \renewcommand*\FBthinspace}{%
3321     \begingroup%
3322     \LWR@FBcancel%
3323     \LWR@origampersand\#x202f;% \,
3324     \endgroup%
3325   }
3326   \renewcommand*\FBguillspace}{%
3327     \begingroup%
3328     \LWR@FBcancel%
3329     \LWR@origampersand{}nbsp;% ~, for \og xyz \fg{}
3330     \endgroup%
3331   }
3332   \DeclareDocumentCommand\FBmedkern{}{}{%
3333     \begingroup%
3334     \LWR@FBcancel%
3335     \LWR@origampersand\#x202f;% \,
3336     \endgroup%
3337   }
3338   \DeclareDocumentCommand\FBthickkern{}{}{%
3339     \begingroup%
3340     \LWR@FBcancel%
3341     \LWR@origampersand{}nbsp;% ~
3342     \endgroup%
3343   }
3344   \renewcommand*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3345   \iffBunicode
3346   \else

```

```

3347     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
3348     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}
3349     \fi
3350 }
3351 }

```

40.3 HTML tags

`\LWR@htmltagc` $\langle\{tag}\rangle$ Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOF, LOT.

```

3352 \newcommand*{\LWR@htmltagc}[1]{%
3353 {%
3354 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3355 \begingroup%
3356 \LWR@FBcancel%
3357 \ifmmode\else\protect\LWR@origttfamily\fi%
3358 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3359 \endgroup%
3360 }%
3361 }

```

Env `LWR@nestspan` Disable minipage, `\parbox`, and HTML `<div>`s inside a ``.

 `\begin{LWR@nestspan}` must follow the opening `` tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

 `\end{LWR@nestspan}` must follow the `` or a `<p>` may appear inside the span.

```

3362 \newcommand*{\LWR@nestspanitem}{%
3363 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3364 \LWR@origitem%
3365 }
3366
3367 \newenvironment*{LWR@nestspan}
3368 {%
3369 \LWR@traceinfo{LWR@nestspan}%
3370 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3371 {%
3372   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3373 }%
3374 {% not in a lateximage
3375   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3376   \addtocounter{LWR@spandepth}{1}%
3377   \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%

```

```

3378 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3379 \renewcommand{\BlockClassSingle}[2]{##2}%
3380 \renewcommand{\LWR@forcenewpage}{-}%
3381 \renewcommand{\LWR@itemizestart}{-}%
3382 \let\item\LWR@nestspanitem%
3383 }%
3384 \renewcommand{\LWR@itemizeend}{-}%
3385 \renewcommand{\LWR@enumeratestart}{-}%
3386 \let\item\LWR@nestspanitem%
3387 }%
3388 \renewcommand{\LWR@enumerateend}{-}%
3389 \renewcommand{\LWR@descriptionstart}{-}%
3390 \let\item\LWR@nestspanitem%
3391 }%
3392 \renewcommand{\LWR@descriptionend}{-}%
3393 }% not in a lateximage
3394 }%
3395 {%
3396 \ifthenelse{\cntttest{\value{LWR@lateximagedepth}}{>}{0}}{
3397 }{
3398 {\addtocounter{LWR@spandepth}{-1}}%
3399 \LWR@traceinfo{LWR@nestspan: done}%
3400 }
3401
3402 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

`\LWR@htmlspan` $\{ \langle tag \rangle \} \{ \langle text \rangle \}$



`\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.

```

3403 \NewDocumentCommand{\LWR@htmlspan}{m +m}{-}%
3404 \LWR@ensuredoingapar%
3405 \LWR@htmltagc{#1}%
3406 \begin{LWR@nestspan}%
3407 #2%
3408 \LWR@htmltagc{/#1}%
3409 \end{LWR@nestspan}%
3410 }

```

`\LWR@htmlspanclass` $[\langle style \rangle] \{ \langle class \rangle \} \{ \langle text \rangle \}$

```

3411 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{-}%
3412 \LWR@ensuredoingapar%
3413 \LWR@subhtmlclass{span}[#1]{#2}%
3414 \begin{LWR@nestspan}%
3415 #3%

```

```

3416 \LWR@htmltagc{/span}%
3417 \end{LWR@nestspan}%
3418 }

```

`\LWR@htmltag` $\{ \langle tag \rangle \}$

Print an HTML tag: `<tag>`

```

3419 \newcommand*{\LWR@htmltagb}[1]{%
3420 \LWR@htmltagc{#1}%
3421 \endgroup%
3422 }
3423
3424 \newcommand*{\LWR@htmltag}{%
3425 \begingroup\catcode'\_ =12
3426 \LWR@htmltagb%
3427 }

```

40.4 Block tags and comments

In the following, `\origttfamily` breaks ligatures, which may not be used for HTML codes:

`\LWR@htmlopencomment`
`\LWR@htmlclosecomment`

```

3428 \newcommand*{\LWR@htmlopencomment}{%
3429 {%
3430 % \LWR@traceinfo{\LWR@htmlopencomment}%
3431 \begingroup%
3432 \LWR@FBcancel%
3433 \ifmode\else\protect\LWR@origttfamily\fi%
3434 \LWR@origtextless{!}{-}{-}%
3435 \endgroup%
3436 }%
3437 }
3438
3439 \newcommand*{\LWR@htmlclosecomment}{%
3440 {%
3441 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3442 \begingroup%
3443 \LWR@FBcancel%
3444 \ifmode\else\protect\LWR@origttfamily\fi%
3445 {-}{-}\LWR@origtextgreater{ }%
3446 \endgroup%
3447 }%
3448 }

```

`\LWR@htmlcomment` `{<comment>}`

```
3449 \newcommand{\LWR@htmlcomment}[1]{%
3450 \LWR@htmlopencomment{}}%
3451 {%
3452 \LWR@origttfamily% break ligatures
3453 #1%
3454 }%
3455 \LWR@htmlclosecomment{}}
```

`\LWR@htmlblockcomment` `{<comment>}`

```
3456 \newcommand{\LWR@htmlblockcommentb}[1]
3457 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3458
3459 \newcommand{\LWR@htmlblockcomment}
3460 {%
3461 \begingroup\catcode'\_ =12%
3462 \LWR@htmlblockcommentb%
3463 }
```

`\LWR@htmlblocktag` `{<tag>}` print a stand-alone HTML tag

```
3464 \newcommand*\LWR@htmlblocktag[1]{%
3465 \LWR@stoppars%
3466 \LWR@htmltag{#1}%
3467 \LWR@startpars%
3468 }
```

40.5 Div class and element class

`\LWR@subhtmlclass` `{<element>}` [`<style>`] [`<class>`]

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
3469 \NewDocumentCommand{\LWR@subhtmlclass}{m o m}{%
3470 \IfValueTF{#2}%
3471 {% option
3472   \ifthenelse{\equal{#2}{}}%
3473   {\LWR@htmltag{#1 class="#3" }}% empty option
3474   {\LWR@htmltag{#1 class="#3" style="#2" }}% non-empty option
3475 }% option
3476 {\LWR@htmltag{#1 class="#3" }}% no option
3477 }
```

`\LWR@htmlclass {<element>} {<class>} [<style>]`

```
3478 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3479 \LWR@stoppars%
3480 \LWR@subhtmlclass{#1}[#2]{#3}%
3481 \LWR@startpars%
3482 }
```

`\LWR@htmlclassend {<element>} {<class>}`

```
3483 \newcommand*{\LWR@htmlclassend}[2]{%
3484 \LWR@stoppars%
3485 \LWR@htmltag{/#1}%
3486 \ifbool{HTMLDebugComments}{%
3487   \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3488 }{}%
3489 \LWR@startpars%
3490 }
```

`\LWR@htmldivclass [<style>] {<class>}`

```
3491 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3492 \LWR@htmlclass{div}[#1]{#2}%
3493 }
```

`\LWR@htmldivclassend {<class>}`

```
3494 \newcommand*{\LWR@htmldivclassend}[1]{%
3495 \LWR@htmlclassend{div}{#1}%
3496 }
```

40.6 Single-line elements

A single-line element, without a paragraph tag for the line of text:

`\LWR@htmlclassline {<element>} [<style>] {<class>} {<text>}`

```
3497 \NewDocumentCommand{\LWR@htmlclassline}{m o m +m}{%
3498 \LWR@stoppars
3499 \LWR@subhtmlclass{#1}[#2]{#3}%
3500 #4%
3501 \LWR@htmltag{/#1}
3502 \LWR@startpars
3503 }
```

40.7 HTML5 semantic elements

`\LWR@htmllement` $\{ \langle element \rangle \}$

```
3504 \newcommand*\LWR@htmllement}[1]{%
3505 \LWR@htmlblocktag{#1}
3506 }
```

`\LWR@htmllementend` $\{ \langle element \rangle \}$

```
3507 \newcommand*\LWR@htmllementend}[1]{%
3508 \LWR@stoppars
3509 \LWR@htmltag{/#1}
3510 \LWR@startpars
3511 }
3512
3513 \end{warpHTML}
```

40.8 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.

For other direct-formatting commands, see section [73](#).

Env `BlockClass` $[\langle style \rangle] \{ \langle class \rangle \}$ High-level interface for div classes.

Ex: `\begin{BlockClass}{class} text \end{BlockClass}`

for HTML output:

```
3514 \begin{warpHTML}
3515 \NewDocumentEnvironment{BlockClass}{o m}%
3516 {
3517 \LWR@htmldivclass[#1]{#2}
3518 }
3519 {
3520 \LWR@htmldivclassend{#2}
3521 }
3522 \end{warpHTML}
```

for PRINT output:

```
3523 \begin{warpprint}
3524 \NewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3525 \end{warpprint}
```

`\BlockClassSingle` $\{ \langle class \rangle \} \{ \langle text \rangle \}$ A single-line `<div>`, without a paragraph tag for the line of text.

for HTML output:

```

3526 \begin{warpHTML}
3527 \newcommand{\BlockClassSingle}[2]{%
3528 \LWR@html@element@classline{div}{#1}{#2}%
3529 }
3530 \end{warpHTML}

```

for PRINT output:

```

3531 \begin{warpprint}
3532 \newcommand{\BlockClassSingle}[2]{#2}
3533 \end{warpprint}

```

`\InlineClass` [*<style>*] [*<class>*] [*<text>*] High-level interface for inline span classes.

for HTML output:

```

3534 \begin{warpHTML}
3535 \NewDocumentCommand{\InlineClass}{o m +m}{%
3536 \LWR@html@spanclass[#1]{#2}{#3}%
3537 }
3538 \end{warpHTML}

```

for PRINT output:

```

3539 \begin{warpprint}
3540 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
3541 \end{warpprint}

```

40.9 Closing HTML tags

for HTML output:

```

3542 \begin{warpHTML}

```

Sections H1, H2, etc. do not need a closing HTML tag, but we add a comment for readability:

```

3543 \newcommand*\LWR@printclosepart}
3544   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3545 \newcommand*\LWR@printclosechapter}
3546   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3547 \newcommand*\LWR@printclosesection}
3548   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3549 \newcommand*\LWR@printclosesubsection}
3550   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3551 \newcommand*\LWR@printclosesubsubsection}
3552   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3553 \newcommand*\LWR@printcloseparagraph}
3554   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3555 \newcommand*\LWR@printclose subparagraph}
3556   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3557 \newcommand*{\LWR@printcloselistitem}
3558     {\LWR@htmltag{/li}}
3559 \newcommand*{\LWR@printclosedescitem}
3560     {\LWR@htmltag{/dd}}
3561 \newcommand*{\LWR@printcloseitemize}
3562     {\LWR@htmltag{/ul}}
3563 \newcommand*{\LWR@printcloseenumerate}
3564     {\LWR@htmltag{/ol}}
3565 \newcommand*{\LWR@printclosedescription}
3566     {\LWR@htmltag{/dl}}

3567 \end{warpHTML}

```

41 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 42 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: 3568 `\begin{warpHTML}`

Ctrl `LWR@spandepth` Do not create paragraph tags inside of an HTML span.

```

3569 \newcounter{LWR@spandepth}
3570 \setcounter{LWR@spandepth}{0}

```

Bool `LWR@doingstartpars` Tells whether paragraphs may be generated.

```

3571 \newbool{LWR@doingstartpars}
3572 \boolfalse{LWR@doingstartpars}

```

Bool `LWR@doingapar` Tells whether have actually generated and are currently processing paragraph text.

```

3573 \newbool{LWR@doingapar}
3574 \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:

```

3575 \newcommand*{\LWR@ensuredoingapar}{%
3576 \ifbool{LWR@doingstartpars}%
3577 {\global\booltrue{LWR@doingapar}}%
3578 {}%
3579 }

```

\LWR@openparagraph

```

3580 \newcommand*{\LWR@openparagraph}
3581 {%

```

See if paragraph handling is enabled:

```

3582 \ifbool{LWR@doingstartpars}%
3583 {% handling pars

```

See if have already started a lateximage or a . If so, do not generate nested paragraph tags.

```

3584 \ifthenelse{%
3585 \cinttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3586 \cinttest{\value{LWR@spandepth}}{>}{0}%
3587 }% nested par tags?

```

If so: Do nothing if already started a lateximage page. Cannot nest a lateximage. Also do nothing if already inside a . Do not nest paragraph tags inside a .

```

3588 {}% no nested par tags

```

Else: No lateximage or has been started yet, so it's OK to generate paragraph tags.

```

3589 {% yes nest par tags
3590 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```

3591 \ifnumcomp{\@listdepth}{>}{0}%
3592 {%

```

If so, leave some horizontal room in the \LaTeX PDF output for list labels:

```

3593 \LWR@origspace{1in}%
3594 }%
3595 {}%

```

Now have started a paragraph.

```
3596      \global\booltrue{LWR@doingapar}%
```

At the end of each paragraph, generate closing tag and do regular /par stuff. (Attempting to use the everyhook cr hook for \LWR@closeparagraph does not work well.)

```
3597      \let\par\LWR@closeparagraph%
3598    }% end of yes nest par tags
3599}% end of handling pars
3600{% not handling pars
3601 }
```

\LWR@closeparagraph

```
3602 \newcommand*{\LWR@closeparagraph}
3603 {%
```

See if paragraph handling is enabled:

```
3604 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
3605 {% handling pars
```

See if already started a lateximage or a :

```
3606   \ifthenelse{%
3607     \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3608     \cnttest{\value{LWR@spandepth}}{>}{0}%
3609   }%
```

Do nothing if already started a lateximage or a , but add a parbreak if in a span but not a lateximage.

```
3610   {% no nested par tags
3611     \ifthenelse{%
3612       \cnttest{\value{LWR@spandepth}}{>}{0}%
3613       \AND%
3614       \cnttest{\value{LWR@lateximagedepth}}{=} {0}%
3615     }%
3616     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3617     {}%
3618   }% no nested par tags
```

If have not already started a `lateximage` or a ``:

```
3619   {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3620       \unskip%
3621       \LWR@htmltagc{/\LWR@tagregularparagraph}%
3622       \LWR@orignewline%
```

No longer doing a paragraph:

```
3623       \global\boolfalse{LWR@doingapar}%
3624 % Disable the special \env{minipage} \& \cs{hspace} interaction
3625 % until a new minipage is found:
3626 %   \begin{macrocode}
3627       \global\boolfalse{LWR@minipagethispar}%
3628   }% end of yes nest par tags
3629}% end of handling pars
```

Add a parbreak if in a span, but not in a table outside a row:

```
3630 {% not handling pars
3631   \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3632   {\ifbool{LWR@intabularmetadata}{}\{\unskip\LWR@htmltagc{br /}}}%
3633   {}%
3634}% not handling pars
```

Finish with regular paragraph processing

```
3635 \LWR@origpar%
3636 }
```

```
3637 \end{warpHTML}
```

42 Paragraph start/stop handling

These commands allow/disallow the generation of HTML paragraph tags.

Section 41 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: 3638 `\begin{warpHTML}`

`\LWR@startpars` Begin handling HTML paragraphs. This allows an HTML paragraph to start, but one has not yet begun.

```
3639 \newcommand*{\LWR@startpars}%
3640 {%
```

Ignore if inside a span:

```
3641 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3642 {}%
3643 {%
```

See if currently handling HTML paragraphs:

```
3644 \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3645 {}%
```

If not currently in paragraph mode:

```
3646 {%
```

At the start of each paragraph, generate an opening tag:

```
3647 \PushPreHook{par}{\LWR@openparagraph}%
```

At the end of each paragraph, generate closing tag and do regular `/par` actions:

```
3648 \let\par\LWR@closeparagraph
3649
3650 }% an intentionally blank line
```

Are now handling paragraphs, but have not yet actually started one:

```
3651 \global\setbool{LWR@doingstartpars}{true}%
```

No `<par>` tag yet to undo:

```
3652 \global\boolfalse{LWR@doingapar}%
3653 }% nestspan
3654 }
```

`\LWR@stoppars` Stop handling HTML paragraphs. Any currently open HTML paragraph is closed, and no more will be opened.

```
3655 \newcommand*{\LWR@stoppars}%
3656 {%
```

Ignore if inside a span:

```
3657 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3658 {}%
3659 {%
```

See if currently handling HTML paragraphs:

```
3660 \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3661 {%
```

Print a closing tag:

```
3662 \unskip%
3663 \LWR@htmltagc{/LWR@tagregularparagraph}%
3664 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
3665 \global\boolfalse{LWR@doingapar}%
```

Disable the special minipage & \hspace interaction until a new minipage is found:

```
3666 \global\boolfalse{LWR@minipagethispar}
3667
3668 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3669 {}%
```

See if currently allowing HTML paragraphs:

```
3670 \ifbool{LWR@doingstartpars}%
```

If so: clear the par hook to no longer catch paragraphs:

```
3671 {\ClearPreHook{par}}%
```

Else: do nothing

```
3672 {}%
```

no longer in paragraph mode

```
3673 \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
3674 \global\boolfalse{LWR@doingapar}%
```

```
3675 }% nestspan
```

```
3676 }
```

```
3677 \end{warpHTML}
```

43 Page headers and footers

for HTML & PRINT: 3678 \begin{warpall}

In the following, catcode is manually changes back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
3679 \newcommand{\LWR@firstpagetop}{} % for the home page alone
```

```
3680 \newcommand{\LWR@pagetop}{} % for all other pages
```

```
3681 \newcommand{\LWR@pagebottom}{}
3682
```

```
3683 \newcommand{\LWR@setfirstpagetop}[1]{%
```

```
3684 \renewcommand{\LWR@firstpagetop}{#1}
```

```
3685 \catcode'\_ =8
```

```
3686 }
```

\HTMLFirstPageTop {<*text and logos*>}

```
3687 \newcommand{\HTMLFirstPageTop}{%
```

```
3688 \catcode'\_ =12
```

```
3689 \LWR@setfirstpagetop
```

```
3690 }
```

```
3691 \newcommand{\LWR@setpagetop}[1]{%
```

```
3692 \renewcommand{\LWR@pagetop}{#1}
```

```
3693 \catcode'\_ =8
```

```
3694 }
```

\HTMLPageTop {<*text and logos*>}

```
3695 \newcommand{\HTMLPageTop}{%
```

```
3696 \catcode'\_ =12
```

```

3697 \LWR@setpagetopb
3698 }

3699 \newcommand{\LWR@setpagebottomb}[1]{%
3700 \renewcommand{\LWR@pagebottom}{#1}
3701 \catcode'\_ =8
3702 }

```

`\HTMLPageBottom` `{\langle text and logos \rangle}`

```

3703 \newcommand{\HTMLPageBottom}{%
3704 \catcode'\_ =12
3705 \LWR@setpagebottomb
3706 }

```

```

3707 \end{warpall}

```

44 CSS

for HTML output: 3708 `\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.

```

3709 \newcommand*{\LWR@currentcss}{lwarp.css}

```

`\CSSFilename` `{\langle new-css-filename.css \rangle}` Assigns the css file to be used by the following HTML pages.

```

3710 \newcommand*{\LWR@newcssb}[1]{%
3711 \renewcommand*{\LWR@currentcss}{#1}
3712 \catcode'\_ =8
3713 }
3714
3715 \newcommand*{\CSSFilename}{
3716 \catcode'\_ =12
3717 \LWR@newcssb
3718 }
3719 \end{warpHTML}

```

for PRINT output: 3720 `\begin{warpprint}`
3721 `\newcommand*{\CSSFilename}[1]{}`
3722 `\end{warpprint}`

45 HTML meta description and author

for HTML & PRINT: 3723 `\begin{warpall}`

`\HTMLAuthor` `{\langle authorname \rangle}` The author to place into an HTML meta tag.

```
3724 \providecommand{\theauthor}{}
3725 \newcommand{\theHTMLAuthor}{\theauthor}
3726
3727 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3728 \end{warpall}
```

for HTML & PRINT: 3729 `\begin{warpall}`

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` `{\langle New HTML meta description. \rangle}` Assigns the HTML file's description meta tag.

```
3730 \newcommand{\LWR@currentHTMLDescription}{}
3731
3732 \newcommand{\HTMLDescription}[1]{%
3733 \renewcommand{\LWR@currentHTMLDescription}{#1}
3734 }
3735
3736 \end{warpall}
```

46 Footnotes

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

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.

46.1 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 46.4 for the implementation.

46.2 Minipage footnotes

See section 46.5 for how minipage footnotes are gathered. See section 72.3 for how minipage footnotes are placed into the document.

46.3 Titlepage thanks

See section 53.6 for titlepage footnotes.

46.4 Regular page footnote implementation

for HTML output: 3737 `\begin{warpHTML}`

Patch \LaTeX footnotes to use a new `\box` for lwarp footnotes.

3738 `\newbox\LWR@footnotes`

Much of the following has unneeded print-mode formatting removed.

`\@makefntext` `{\langle text \rangle}`

3739 `\long\def\@makefntext#1{\@thefnmark #1}`

`\@makefnmark`

3740 `\def\@makefnmark{\hbox{\@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}`

```

3741 \long\def\LWR@footnotetext#1{%
3742 \global\setbox\LWR@footnotes=\vbox{%

```

Add to any current footnotes:

```

3743 \unvbox\LWR@footnotes%

```

Remember the footnote number for `\ref`:

```

3744 \protected@edef\@currentlabel{%
3745 \csname p@footnote\endcsname\@thefnmark%
3746 }% \@currentlabel

```

Open a group:

```

3747 \color@begingroup%

```

Use HTML superscripts even inside a lateximage:

```

3748 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%

```

Use paragraph tags if in a tabular data cell or a lateximage:

```

3749 \ifthenelse{%
3750 \boolean{LWR@doingstartpars} \AND%
3751 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3752 }%
3753 {}%
3754 {\LWR@htmltagc{\LWR@tagregularparagraph}}%

```

Append the footnote to the list:

```

3755 \@makefntext{#1}%

```

Closing paragraph tag:

```

3756 \ifthenelse{%
3757 \boolean{LWR@doingstartpars} \AND%
3758 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3759 }%
3760 {\par}%

```

```

3761   {%
3762       \LWR@htmltagc{/LWR@tagregularparagraph}%
3763       \LWR@orignewline%
3764   }%

```

Close the group:

```

3765   \color@endgroup%
3766 }% vbox
3767 }%

```

`\@footnotetext` `{\text}`

```

3768 \let\@footnotetext\LWR@footnotetext

```

46.5 Minipage footnote implementation

`\@mpfootnotetext` `{\text}`

```

3769 \long\def\@mpfootnotetext#1{%
3770 \global\setbox\@mpfootins\vbox{%
3771     \unvbox\@mpfootins
3772     \reset@font\footnotesize
3773     \hsize\columnwidth
3774     \@parboxrestore
3775     \protected@edef\@currentlabel
3776     {\csname p@mpfootnote\endcsname\@thefnmark}%
3777     \color@begingroup
3778     \@makefntext{%
3779         \ignorespaces#1%
3780     }%

```

Don't add the closing paragraph tag if are inside a lateximage:

```

3781     \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3782     {}%
3783     {%
3784         \LWR@htmltagc{/LWR@tagregularparagraph}%
3785         \LWR@orignewline%
3786     }%
3787     \color@endgroup%
3788 }% vbox
3789 }

```

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

3790 \newcommand*\LWR@printpendingfootnotes}{%
3791 \ifvoid\LWR@footnotes\else
3792   \LWR@forcenewpage
3793   \begin{BlockClass}{footnotes}
3794   \LWR@origmedskip
3795   \unvbox\LWR@footnotes
3796   \setbox\LWR@footnotes=\vbox{}
3797   \end{BlockClass}
3798 \fi
3799 }

```

`\LWR@epubprintpendingfootnotes` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

3800 \newcommand*\LWR@epubprintpendingfootnotes}{%
3801 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3802 {\LWR@printpendingfootnotes}%
3803 }{%
3804 }

3805 \end{warpHTML}

```

47 Marginpars

for HTML output: 3806 `\begin{warpHTML}`

`\marginpar` [*left*] {*right*}

```

3807 \renewcommand{\marginpar}[2] [] {%
3808 \LWR@htmlspanclass{marginpar}{#2}%
3809 }

3810 \end{warpHTML}

```

48 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenameno blanks`.
- `\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: 3811 `\begin{warpall}`

Ctrl FileDepth `{\langle section depth \rangle}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of `-5` produces one large HTML file.

3812 `\newcounter{FileDepth}`

3813 `\setcounter{FileDepth}{-5}`

Bool CombineHigherDepths Combine higher-level sections together into one file?

3814 `\newbool{CombineHigherDepths}`

3815 `\booltrue{CombineHigherDepths}`

3816 `\end{warpall}`

for HTML output: 3817 `\begin{warpHTML}`

`\LWR@thisfilename` The currently-active filename or number.

3818 `\newcommand*{\LWR@thisfilename}{}`

`\LWR@thisnewfilename` The filename being sanitized.

3819 `\newcommand*{\LWR@thisnewfilename}{}`

`\LWR@filenameno blanks` `{\langle filename \rangle}`

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` \TeX commands which appear in section names and TOC captions.

3820 `\newcommand*{\LWR@filenameno blanks}[1]{%`

3821 `\begingroup`

Locally temporarily disable direct-formatting commands, not used in filenames:

3822 `\LWR@nullfonts`

3823 `\renewcommand*{\LWR@htmltagc}[1]{}%`

Replaces common symbols and short words with hyphens:

3824 `\edef\LWR@thisnewfilename{#1}`

3825 `\fullexpandarg`

Convert spaces into hyphens:

```
3826 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
3827 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3828 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3829 \StrSubstitute{\LWR@thisnewfilename}%
3830 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3831 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3832 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3833 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3834 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3835 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3836 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3837 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3838 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3839 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
3840 \StrSubstitute{\LWR@thisnewfilename}%
3841 {\textless}{-}[\LWR@thisnewfilename]
3842 \StrSubstitute{\LWR@thisnewfilename}%
3843 {\textgreater}{-}[\LWR@thisnewfilename]
3844 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3845 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3846 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3847 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3848 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3849 \StrSubstitute{\LWR@thisnewfilename}%
3850 {\textbackslash}{-}[\LWR@thisnewfilename]
3851 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3852 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3853 %    "~{}" for babel
3854 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3855 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3856 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
3857 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3858 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3859 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3860 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3861 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3862 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3863 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3864 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3865 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3866 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
```

```

3867 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3868 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3869 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3870 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3871 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3872 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3873 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3874 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3875 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3876 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3877 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3878 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3879 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3880 %      emdash
3881 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3882 %      endash
3883 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3884 \endgroup
3885 }

```

`\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.

```
3886 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

3887 \LWR@htmlElementclassend{section}{tbody}
3888
3889 \LWR@printpendingfootnotes
3890

```

No footer between files if EPUB:

```

3891 \ifbool{FormatEPUB}
3892 {}
3893 {
3894   \LWR@htmlElement{footer}
3895
3896   \LWR@pagebottom
3897

```

```
3898 \LWR@htmllementend{footer}
3899 }
```

No bottom navigation if are finishing the home page or formatting for EPUB or a word-processor.

```
3900 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3901 {}
3902 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3903 \LWR@stoppars
3904 \LWR@htmltag{/body}\LWR@orignewline
3905 \LWR@htmltag{/html}\LWR@orignewline
3906 \LWR@orignewpage
3907
3908 \addtocounter{LWR@htmlfilename}{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.

```
3909 \ifbool{FileSectionNames}%
3910 {\LWR@filenamenooblanks{#1}}
3911 {\renewcommand*\LWR@thisfilename{\theLWR@htmlfilename}}
```

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.

```
3912 \LWR@htmlblockcomment{%
3913 |Start file|%
3914 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
3915 }
```

At the top of the starting file:

```
3916 \LWR@stoppars
3917
3918 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
3919
```

No navigation between files if formatting for an EPUB or word processor:

```
3920 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
```

```
3921 {}
3922 {\LWR@topnavigation}
3923
```

No header if between files if formatting for an EPUB or word processor:

```
3924 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3925 {}
3926 {
3927   \LWR@htmlElement{header}
3928
3929   \LWR@pagetop
3930
3931   \LWR@htmlElementend{header}
3932 }
3933
```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```
3934 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3935 {}
3936 {\ifcvoid{thetitle}{}\LWR@printthetitle}}
3937
```

No sidetoc if formatting for an EPUB or word processor:

```
3938 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3939 {}
3940 {\LWR@sidetoc}
3941
```

Start of the <textbody>:

```
3942 \LWR@htmlElementclass{section}{textbody}
3943
```

Keep paragraph tags disabled for now:

```
3944 \LWR@stoppars
3945
```

Track the page numbers:

```
3946 \setcounter{LWR@latestautopage}{\value{page}}
3947 }
```

```
3948 \end{warpHTML}
```

49 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing \TeX 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 `pdf \TeX` to generate characters and `pdf totext` to read them. If extra symbols appear in the text, it may be that `pdf \TeX` is actually producing a symbol over or under a character, resulting in `pdf totext` picking up the accent symbol separately.



$\Xe $\TeX$$ and $\text{Lua}\mathbb{X}$ directly support accented section and file names.

for HTML output: 3949 `\begin{warphTML}`

49.1 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.

```
3950 \newbool{LWR@forcinghtmlpage}
3951 \boolfalse{LWR@forcinghtmlpage}
3952
3953 \newcommand*{\ForceHTMLPage}{%
3954 \global\booltrue{LWR@forcinghtmlpage}%
3955 }
```

`\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.

```
3956 \newbool{LWR@forcinghtmltoc}
3957 \boolfalse{LWR@forcinghtmltoc}
3958
3959 \newcommand*{\ForceHTMLTOC}{%
3960 \global\booltrue{LWR@forcinghtmltoc}%
3961 }
```

```
3962 \end{warpHTML}
```

for PRINT output:

```
3963 \begin{warpprint}
3964 \newcommand*{\ForceHTMLPage}{}
3965 \newcommand*{\ForceHTMLTOC}{}
3966 \end{warpprint}
```

for HTML output:

```
3967 \begin{warpHTML}
```

49.2 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.

```
3968 \newbool{LWR@mainmatter}
3969 \DeclareDocumentCommand{\mainmatter}{}{}%
3970 \booltrue{LWR@mainmatter}%
3971 }
```

`\frontmatter` Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
3972 \DeclareDocumentCommand{\frontmatter}{}{}%
3973 \boolfalse{LWR@mainmatter}%
3974 }
```

`\backmatter` Declare the back matter section of the document. Does not reset the page number.

```
3975 \DeclareDocumentCommand{\backmatter}{}{}%
3976 \boolfalse{LWR@mainmatter}
3977 }
```

49.3 Sectioning support macros

`\LWR@sectionnumber` $\langle section\ type \rangle$

Typeset a section number and its trailing space with CSS formatting:

```
3978 \newcommand*{\LWR@sectionnumber}[1]{}%
3979 \InlineClass{sectionnumber}{{#1}}%
3980 }
```

`autosec` A tag used by the TOC and index.

`\LWR@createautosec` $\langle section\ type\rangle$

Create an autosection tag.

```
3981 \newcommand*\LWR@createautosec[1]{%
3982 \LWR@htmltag{#1 id="autosec-\thepage"{}%
3983 }
```

`\LWR@pushoneclose` $\langle depth\rangle$ $\langle printclose\rangle$ 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.

```
3984 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

`\LWR@startnewdepth` $\langle depth\rangle$ $\langle printclose\rangle$

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.

```
3985 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
3986 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
3987 \LWR@pushoneclose{#1}{#2}%
3988 }
```

Ctrl `\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.

```
3989 \newcounter{\LWR@prevFileDepth}
3990 \setcounter{\LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\@secntformat`

```
3991 \def\@secntformat#1{\csname the#1\endcsname\protect\quad}
```

`\LWR@section` * [$\langle TOC\ name\rangle$] $\langle name\rangle$ $\langle sectiontype\rangle$

The common actions for the high-level sectioning commands.

```

3992 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
3993 \LWR@traceinfo{LWR@section|#2|#3|}%
3994 % \ifthenelse{%
3995 % \equal{#2}{\NoValue-}\AND%
3996 % \isempty{#3}%
3997 % }%
3998 % {% empty section name
3999 % \LWR@traceinfo{LWR@section: empty section}%
4000 % }% empty section name
4001 % {% not an empty section name
4002 \LWR@traceinfo{LWR@section: not an empty section}%
4003 \LWR@stoppars%

```

Cancel special minipage horizontal space interaction:

```

4004 \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 roc and index page number points to the section:

```

4005 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4006 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{%
4007 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{%
4008 \ifthenelse{%
4009   \(%
4010     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
4011     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
4012     \(\boolean{LWR@forcinghtmlpage}\)\)%
4013   \)%
4014   \AND%
4015   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4016   \AND%
4017   \(%
4018     \NOT\boolean{CombineHigherDepths}\OR%
4019     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4020   \)%
4021   \AND%
4022   \NOT\equal{#3}{}}% phantomsection
4023 }%

```

If so: start a new HTML file:

```

4024 {% new file
4025   \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```
4026 \IfNoValueTF{#2}%
```

If no optional entry

```
4027 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4028 {\LWR@newhtmlfile{#2}}%
```

```
4029}% new file
```

Else: No new HTML file:

```
4030 {% not new file
```

Generate a new \LaTeX page so that toc and index page number points to the section:

```
4031 \LWR@traceinfo{LWR@section: not a new HTML file}%
```

```
4032 \LWR@orignewpage%
```

```
4033
```

```
4034}% not new file
```

Remember this section's name for \nameref:

```
4035 \IfValueTF{#3}{%
```

```
4036 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
```

```
4037 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
```

```
4038 }{}%
```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```
4039
```

```
4040 \ifbool{HTMLDebugComments}{%
```

```
4041 \LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
```

```
4042 }{}%
```

```
4043
```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
4044 \ifthenelse{%
```

```
4045 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
```

```
4046 }%
```

```
4047 {\LWR@startpars}%
```

```
4048 }%
```

Create the opening tag with an autosec:

```
4049 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

Check if starred:

```
4050 \IfBooleanTF{#1}%
4051 {%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4052 \ifbool{LWR@forcinghtmltoc}%
4053 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4054 {}%
4055 }% starred
```

Not starred, so step counter and add to TOC:

```
4056 {% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
4057 \ifthenelse{%
4058 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4059 }%
4060 {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```
4061 \LWR@traceinfo{LWR@section: about to test main matter}%
4062 \ifbool{LWR@mainmatter}%
4063 {%
4064 \LWR@traceinfo{LWR@section: yes mainmatter}%
4065 \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
4066 \LWR@traceinfo{LWR@section: about to addcontentsline}%
4067 \addcontentsline{toc}{#4}%
4068 {%
4069 \protect\numberline{\csuse{the#4}}%
4070 {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4071 }%
4072 \LWR@traceinfo{LWR@section: finished addcontentsline}%
4073 }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```

4074      {% not main matter
4075          \LWR@traceinfo{LWR@section: no main matter}%
4076          \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4077      }% end of not main matter
4078  }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

4079  {%
4080      \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4081  }%

```

For part, print the section type:

```

4082  \ifbool{LWR@mainmatter}%
4083  {%
4084      \ifthenelse{%
4085          \(<\cnttest{\csuse{LWR@depth#4}}{<=})%
4086          {\value{secnumdepth}}\ ) \AND%
4087          \(<\cnttest{\csuse{LWR@depth#4}}{<=}{LWR@depthpart}\ )%
4088      }%
4089      {\csuse{#4name}~{}}%
4090  }%

```

Print the section number:

```

4091      \LWR@traceinfo{LWR@section: about to print section number}%
4092      \ifthenelse{%
4093          \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4094      }%
4095      {\protect\LWR@sectionnumber{\@secntformat{#4}}}%
4096      {}%
4097      \LWR@traceinfo{LWR@section: finished print section number}%
4098  }{}%
4099 }% end of not starred

```

Print the section name:

```
4100 #3%
```

Close the heading tag, such as /H2:

```
4101 \LWR@htmltag{\csuse{LWR>tag#4end}}%
```

Generate a \LaTeX label:

```
4102 \label{autopage-\thepage}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
4103 \ifthenelse{%
4104     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
4105 }%
4106 {\LWR@startpars}%
4107 {}%
```

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.

```
4108 \ifthenelse{%
4109     \NOT\equal{#1}{\BooleanTrue}\OR%
4110     \cnttest{\csuse{LWR@depth#4}}{=} {\LWR@depthpart}%
4111 }%
4112 {% not starred
4113     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4114 }% not starred
4115 {}%
```

Reset to defaults if not a phantomsection:

```
4116 \ifstrempty{#3}%
4117 {}%
4118 {%
4119 \global\boolfalse{LWR@forcinghtmlpage}%
4120 \global\boolfalse{LWR@forcinghtmltoc}%
4121 }%
4122 %
4123 % }% not an empty section name
4124 \LWR@traceinfo{LWR@section: done}%
4125 }
```

49.4 `\section` and friends

`\part` * [*(TOC name)*] {*(name)*}

```
4126 \DeclareDocumentCommand{\part}{s o m}{%
4127 \LWR@epubprintpendingfootnotes%
```

```

4128 \LWR@stoppars%
4129
4130 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4131
4132 \LWR@section{#1}{#2}{#3}{part}%
4133 }

```

`\chapter` * [*TOC name*] {*name*}

```

4134 \@ifundefined{chapter}
4135 {}
4136 {%
4137 \DeclareDocumentCommand{\chapter}{s o m}{%
4138 \LWR@traceinfo{chapter #3}%
4139 \LWR@epubprintpendingfootnotes%
4140 \LWR@stoppars%
4141
4142 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4143
4144 \LWR@section{#1}{#2}{#3}{chapter}%
4145 }
4146 }

```

`\section` * [*TOC name*] {*name*}

```

4147 \DeclareDocumentCommand{\section}{s o m}{%
4148 \LWR@epubprintpendingfootnotes%
4149 \LWR@stoppars%
4150
4151 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4152
4153 \LWR@section{#1}{#2}{#3}{section}%
4154 }

```

`\subsection` * [*TOC name*] {*name*}

```

4155 \DeclareDocumentCommand{\subsection}{s o m}{%
4156 \LWR@epubprintpendingfootnotes%
4157 \LWR@stoppars%
4158
4159 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4160
4161 \LWR@section{#1}{#2}{#3}{subsection}%
4162 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

4163 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4164 \LWR@epubprintpendingfootnotes%
4165 \LWR@stoppars%
4166
4167 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4168 {\LWR@printclosesubsubsection}%
4169
4170 \LWR@section{#1}{#2}{#3}{subsubsection}%
4171 }

```

`\paragraph` * [*(TOC name)*] {*(name)*}

```

4172 \DeclareDocumentCommand{\paragraph}{s o m}{%
4173 \LWR@epubprintpendingfootnotes%
4174 \LWR@stoppars%
4175
4176 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4177
4178 \LWR@section{#1}{#2}{#3}{paragraph}%
4179 }

```

`\subparagraph` * [*(TOC name)*] {*(name)*}

```

4180 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4181 \LWR@epubprintpendingfootnotes%
4182 \LWR@stoppars%
4183
4184 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4185
4186 \LWR@section{#1}{#2}{#3}{subparagraph}%
4187 }

4188 \end{warpHTML}

```

50 Starting a new file

for HTML & PRINT: 4189 `\begin{warpall}`

`\HTMLLanguage` Default language for the HTML lang tag.

```

4190 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4191
4192 \newcommand*{\HTMLLanguage}[1]{%
4193 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4194 }

```

```
4195 \end{warpall}
```

for HTML output: 4196 \begin{warpHTML}

```
\LWR@filestart {<title_suffix>}
```

Creates the opening HTML tags.

```
4197 \newcommand*{\LWR@filestart}[1]{
4198 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4199 \begingroup
4200 \LWR@traceinfo{\LWR@filestart: A}
4201 \LWR@nullfonts
4202 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4203 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4204 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4205 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
4206 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4207 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
4208 \ifcempty{theHTMLAuthor}{-}{
4209 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline
4210 }
```

lwarp is the generator:

```
4211 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
4212 \LWR@orignewline
```

If there is a description, add it now:

```
4213 \ifdefempty{\LWR@currentHTMLDescription}{-}{-}{%
4214 \LWR@htmltag{%
4215 meta name="description" content="\LWR@currentHTMLDescription" /}%
4216     \LWR@orignewline
4217 }%
```

Mobile-friendly viewport:

```
4218 \LWR@htmltag{meta name="viewport" %
4219 content="width=device-width, initial-scale=1.0" /}%
4220     \LWR@orignewline
```

IE patch:

```
4221 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4222 \LWR@htmltag{%
4223 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{-}}%
4224 \LWR@htmltag{/script}\LWR@orignewline
4225 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4226 \ifcvoid{thetitle}{-}{-}{%
4227 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
4228 }%
```

The page's stylesheet:

```
4229 \LWR@htmltag{%
4230 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4231 \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.

```
4232 \ifbool{mathjax}%
4233 {%
4234     \boolfalse{LWR@verbtags}
4235     \verbatiminput{lwarp_mathjax.txt}%
4236     \booltrue{LWR@verbtags}
4237     \LWR@stoppars
4238 }% end of mathjax
4239 {}%
```

End of the header:

```
4240 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4241 \LWR@htmltag{body}\LWR@orignewline
4242 \endgroup
4243 \LWR@traceinfo{LWR@filestart: done}
4244 }
```

```
4245 \end{warpHTML}
```

51 Starting HTML output

for HTML output: 4246 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```
4247 \catcode'\$=\active
4248 \newcommand*\LWR@LwarpStart{
4249 {%
4250 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4251 \ifbool{FormatWordProcessor}{%
4252   \setcounter{FileDepth}{-5}%
4253   \boolfalse{HTMLDebugComments}%
4254 }{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
4255 \edef\LWR@strresult{\HomeHTMLFilename}
4256 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4257 \edef\LWR@strresult{\HTMLFilename}
4258 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4259 \LWR@origonecolumn%
4260 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4261 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
4262 \LWR@origraggedright%
4263 \LetLtxMacro{\}\{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
4264 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4265 \setlength{\parindent}{0pt}
4266 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4267 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4268 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4269 \renewcommand{\ps@plain}{} 
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 4270 \renewcommand*{\centering}{}
4271 \renewcommand*{\raggedleft}{}
4272 \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:

```
4273 \LetLtxMacro{\LWR@origtabular}{\tabular}
4274 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4275 \LetLtxMacro{\tabular}{\LWR@tabular}
4276 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
4277 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in amsmath environments and is also patched by cleveref.

[Label in HTML](#)

```
4278 \let\LWR@origltx@label\ltx@label
4279 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4280 \let\TeX\LWR@TeX
4281 \let\LaTeX\LWR@LaTeX
4282 \let\LuaTeX\LWR@LuaTeX
4283 \let\LuaLaTeX\LWR@LuaLaTeX
4284 \let\XeTeX\LWR@XeTeX
4285 \let\XeLaTeX\LWR@XeLaTeX
4286 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
4287 \LetLtxMacro{\rotatebox}{\LWR@rotatebox}
4288 \LetLtxMacro{\scalebox}{\LWR@scalebox}
4289 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
4290 \global\boolfalse{LWR@doingapar}
4291 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4292 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4293 \LWR@filestart{}
4294 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4295 \LWR@htmltag{header}\LWR@orignewline
4296 \LWR@startpars
4297 \LWR@firstpagetop
4298 \LWR@stoppars
4299 \LWR@htmltag{/header}\LWR@orignewline
4300 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4301 \LWR@htmltag{section class="textbody"{} }
```

Document and page settings:

```
4302 \mainmatter
4303 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section [53.6](#).

```
4304 \if@titlepage
4305   \thanksmarkseries{arabic}
4306 \else
4307   \thanksmarkseries{fnsymbol}
4308 \fi
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4309 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4310 \catcode'\$=\active
```

Allow HTML paragraphs to begin:

```
4311 \LWR@startpars
```

```
4312 \LWR@traceinfo{LWR@lwarpStart: done}
```

```
4313 }
```

```
4314 \catcode'\$=3% math shift until lwarp starts
```

```
4315 \end{warpHTML}
```

52 Ending HTML output

for HTML output: 4316 `\begin{warpHTML}`

`\LWR@requesttoc` `{\boolean}` `{\<suffix>}` Requests that a toc, lof, or lot be generated.

```
4317 \newcommand*\LWR@requesttoc}[2]{%
```

```
4318 \ifbool{#1}
```

```
4319 {
```

```
4320   \expandafter\newwrite\csuse{tf@#2}
```

```
4321   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
```

```
4322 }{}
```

```
4323 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
4324 \newcommand*\LWR@LwarpEnd
```

```
4325 {
```

```
4326 \LWR@stoppars
```

```
4327 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the `tbody`:

```
4328 \LWR@htmlend{section}{tbody}
```

Print any pending footnotes:

```
4329 \LWR@printpendingfootnotes
```

Create the footer:

```
4330 \LWR@html element{footer}
4331
4332 \LWR@pagebottom
4333
4334 \LWR@html elementend{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.

```
4335 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4336 {}
4337 {
4338   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{-}
4339 }
```

```
4340 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4341 \LWR@htmltag{/body}\LWR@orignewline
4342 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4343 \LWR@orignewpage
```

For lateximage commands:

```
4344 \immediate\closeout\LWR@lateximagesfile
4345 }
```

```
4346 \end{warpHTML}
```

53 Titles and the titling package

Supports and extends the titling package.

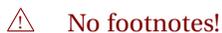
Additional functions include `\published` and `\subtitle`, and the `\author` command has an additional `\affiliation` command to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using `\theauthor` in the main text.

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.

`\printtitle`, `\printauthor`, etc., are provided for use inside the `titlepage` or `titlingpage` environments, and these retain the `\thanks` and `\affiliation`.

Several additional hooks are provided in addition to titling:

| | |
|--------------------------------|--|
| <code>\maketitlehookaa</code> | <code>\maketitlehookaa</code> : Between “published” and the title. |
| <code>\maketitlehookaaa</code> | <code>\maketitlehookaaa</code> : Between the title and the subtitle. |
| <code>\prepublished</code> | <code>\prepublished</code> : Before the “published” field. |
| <code>\postpublished</code> | <code>\postpublished</code> : After the “published” field. |
| <code>\presubtitle</code> | <code>\presubtitle</code> : Before the subtitle. |
| <code>\postsubtitle</code> | <code>\postsubtitle</code> : After the subtitle. |
| <code>\printthanks</code> | <code>\printthanks</code> has been added to force the printing of thanks inside a <code>titlingpage</code> environment when <code>\maketitle</code> is not used. |



No footnotes!

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

```
\begin{titlingpage}
\thanksmarkseries{arabic}
...
```

53.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

| | |
|------------------------------|---|
| <code>\@published</code> | <code>\@title, \@subtitle, \@author, etc.</code> store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included. |
| <code>\@title</code> | |
| <code>\@subtitle</code> | |
| <code>\@author</code> | |
| <code>\@date</code> | |
| <code>\printpublished</code> | <code>\printtitle, \printsubtitle, etc.</code> are user-level macros intended to be used in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included. |
| <code>\printtitle</code> | |
| <code>\printsubtitle</code> | |
| <code>\printauthor</code> | |
| <code>\printdate</code> | |
| <code>\thepublished</code> | <code>\thetitle, \thesubtitle, \theauthor, etc.</code> are user-level sanitized versions which have removed the <code>\thanks</code> and <code>\affiliation</code> , and <code>\and</code> is changed for inline text usage. The author is printed inline without <code>\affiliation</code> or <code>\thanks</code> , with <code>\and</code> 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: |
| <code>\thetitle</code> | |
| <code>\thesubtitle</code> | |
| <code>\theauthor</code> | |
| <code>\thedata</code> | |
| <code>\HTMLPageBottom</code> | <code>{\text}</code> |

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

`\author` `{\author}` While using `\maketitle`, the author is treated as a single-column table and the `\and` feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

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.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

```
\affiliation {<text>}
```

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

```
for HTML output: 4347 \begin{warpHTML}
4348 \newrobustcmd{\affiliation}[1]{\ \ \ \InlineClass{affiliation}{#1}}
4349 \end{warpHTML}
```

```
for PRINT output: 4350 \begin{warpprint}
4351 \newrobustcmd{\affiliation}[1]{\ \ \ \textsc{\small#1}}
4352 \end{warpprint}
```

The following are based on the original titling code:

```
for HTML & PRINT: 4353 \begin{warppall}
```

```
\author {<text>}
```

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`\@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
4354 \renewcommand{\author}[1]{%
4355 \gdef\@author{#1}
4356 \begingroup
4357 \renewcommand{\thanks}[1]{ }
4358 \renewcommand{\and}{\unskip, }
4359 \renewcommand{\thanksmark}[1]{ }
4360 \renewcommand{\thanksgap}[1]{ }
4361 \renewcommand{\affiliation}[1]{ }
4362 \protected@xdef\theauthor{#1}
4363 \endgroup}
```

```
\published {<text>}
```

```
4364 \newcommand{\published}[1]{%
4365 \gdef\@published{#1}
4366 \begingroup
4367 \renewcommand{\thanks}[1]{ }
4368 \renewcommand{\thanksmark}[1]{ }
```

```

4369 \renewcommand{\thanksgap}[1]{
4370 \protected@xdef\thepublished{#1}
4371 \endgroup
4372 }
4373 \newcommand{\@published}{}
4374 \newcommand{\thepublished}{}

```

`\subtitle` $\{ \langle text \rangle \}$

```

4375 \newcommand{\subtitle}[1]{%
4376 \gdef\@subtitle{#1}
4377 \begingroup
4378 \renewcommand{\thanks}[1]{
4379 \renewcommand{\thanksmark}[1]{
4380 \renewcommand{\thanksgap}[1]{
4381 \protected@xdef\thesubtitle{#1}
4382 \endgroup
4383 }
4384 \newcommand{\@subtitle}{}
4385 \newcommand{\thesubtitle}{}

4386 \end{warpall}

```

53.2 Changes to HTML titlepage and titlingpage

for HTML output: 4387 `\begin{warpHTML}`

Env `titlepage` Sets up a titlepage div with a \LaTeX PDF minipage inside.

```

4388 \renewenvironment*{titlepage}
4389 {
4390 \LWR@forcenewpage
4391 \BlockClass{titlepage}\LWR@subminipage
4392 }
4393 {\LWR@endsubminipage\endBlockClass}

```

Env `titlingpage`

```

4394 \renewenvironment*{titlingpage}
4395 {%

```

Start an HTML titlepage div:

```

4396 \begin{titlepage}

```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
4397 \LWR@maketitlesetup
4398 \let\maketitle\LWR@titlingmaketitle
4399 }
4400 {
```

At the end of the environment, end the HTML titlepage div:

```
4401 \end{titlepage}
```

Reset the footnote counter:

```
4402 \@bscontmark
4403 }
```

```
4404 \end{warpHTML}
```

for HTML & PRINT: 4405 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
4406 \newcommand*\printthanks{\@thanks}
```

Env `titlingpage` At the end of the `titlingpage` for both print and HTML, reset footnote markers to zero.

```
4407 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
4408 \end{warpall}
```

53.3 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: 4409 `\begin{warpHTML}`

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
4410 \newcommand\prepublished[1]{%
4411 \def\@bsprepublished{\BlockClass{published}#1}%
4412 }
4413
```

```
4414 \newcommand{\postpublished}[1]{%
4415 \def\@bspostpublished{#1\endBlockClass}%
4416 }
4417
4418 \renewcommand{\pretitle}[1]{%
4419 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}%
4420 }
4421
4422 \renewcommand{\posttitle}[1]{%
4423 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
4424 }
4425
4426 \newcommand{\presubtitle}[1]{%
4427 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
4428 }
4429
4430 \newcommand{\postsubtitle}[1]{%
4431 \def\@bspostsubtitle{#1\endBlockClass}%
4432 }
4433
4434 \renewcommand{\preauthor}[1]{%
4435 \def\@bspreauthor{\BlockClass{author}#1}%
4436 }
4437
4438 \renewcommand{\postauthor}[1]{%
4439 \def\@bspostauthor{#1\endBlockClass}%
4440 }
4441
4442 \renewcommand{\predate}[1]{%
4443 \def\@bspredate{#1\BlockClass{titledate}}%
4444 }
4445
4446 \renewcommand{\postdate}[1]{%
4447 \def\@bspostdate{\endBlockClass#1}%
4448 }
4449
4450 \prepublished{\begin{center}}
4451 \postpublished{\par\end{center}}
4452
4453 \pretitle{\begin{center}}
4454 \posttitle{\par\end{center}}
4455
4456 \presubtitle{\begin{center}}
4457 \postsubtitle{\par\end{center}}
4458
4459 \preauthor{\begin{center}}%
4460 \begin{tabular}[t]{c}%
4461 }
4462 \postauthor{\end{tabular}\par\end{center}}
4463
```

```
4464 \predate{\begin{center}}
4465 \postdate{\par\end{center}}
```

\printpublished

```
4466 \newcommand*{\printpublished}{
4467 \ifthenelse{\equal{\thepublished}{}}
4468 {}
4469 {
4470   \begin{BlockClass}{published}
4471   \@published
4472   \end{BlockClass}
4473 }
4474 }
```

\printtitle

```
4475 \newcommand*{\printtitle}
4476 {
4477 \LWR@stoppars
4478 \LWR@htmltag{h1}%
4479 \@title%
4480 \LWR@htmltag{/h1}
4481 \LWR@startpars
4482 }
```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
4483 \newcommand*{\LWR@printthetitle}
4484 {
4485 \LWR@stoppars
4486 \LWR@htmltag{h1}%
4487 \thetitle%
4488 \LWR@htmltag{/h1}
4489 \LWR@startpars
4490 }
```

\printsubtitle

```
4491 \newcommand*{\printsubtitle}{
4492 \ifthenelse{\equal{\thesubtitle}{}}
4493 {}
4494 {
4495   \begin{BlockClass}{subtitle}
4496   \@subtitle
4497   \end{BlockClass}
```

```
4498 }
4499 }
```

`\printauthor`

```
4500 \newcommand*\printauthor>{
4501 \begin{BlockClass}{author}
4502 \begin{tabular}{c}\@author\end{tabular}
4503 \end{BlockClass}
4504 }
```

`\printdate`

```
4505 \newcommand*\printdate}{%
4506 \begin{BlockClass}{titledate}
4507 \@date
4508 \end{BlockClass}
4509 }
```

```
4510 \end{warpHTML}
```

53.4 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: 4511 `\begin{warpprint}`

`\printpublished`

```
4512 \newcommand*\printpublished>{{\Large\scshape\@published}}
```

`\printtitle`

```
4513 \newcommand*\printtitle>{{\Huge\@title}}
```

`\printsubtitle`

```
4514 \newcommand*\printsubtitle>{{\Large\itshape\@subtitle\bigskip}}
```

`\printauthor`

```
4515 \newcommand*\printauthor
4516   {{{\large\begin{tabular}[t]{c}\@author\end{tabular}}}}
```

`\printdate`

```
4517 \newcommand*{\printdate}{\small\textit{\@date}}
```

53.5 `\maketitle` for print output

`\maketitle` From the titling package, patched to add the publisher and subtitle.

```
4518 \providecommand{\maketitle}{}
4519 \if@titlepage
4520 \renewcommand{\maketitle}{\begin{titlepage}%
4521 \let\footnotesize\small
4522 \let\footnoterule\relax
4523 \LetLtxMacro{\footnote}{\thanks}
4524 \bsmarkseries
4525 \def\@makefnmark{\rlap{\@textsuperscript{%
4526 \normalfont\bsthanksheadpre \tamark \bsthanksheadpost}}}%
4527 \long\def\@makefntext##1{\makethanksmark ##1}
4528 \null\vfil
4529 \vskip 60\p@
4530 \vspace*{\droptitle}
4531 \maketitlehooka
4532 \ifcseempty{@published}
4533 {}
4534 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4535 {\@bspretitle \@title \@bsposttitle}
4536 \ifcseempty{@subtitle}
4537 {}
4538 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4539 \maketitlehookb
4540 {\@bspreauthor \@author \@bspostauthor}
4541 \maketitlehookc
4542 {\@bspredate \@date \@bspostdate}
4543 \maketitlehookd
4544 \par
4545 \thanks
4546 \vfil\null
4547 \end{titlepage}%
4548 \@bscontmark % \setcounter{footnote}{0}%
4549 %%% \bsmttitleempty
4550 } % end titlepage defs
4551 \else
4552 \renewcommand{\maketitle}{\par
4553 \begingroup
4554 \bsmarkseries
4555 \def\@makefnmark{\rlap{\@textsuperscript{%
4556 \normalfont\bsthanksheadpre \tamark \bsthanksheadpost}}}%

```

```

4557     \long\def\@makefntext##1{\makethanksmark ##1}
4558     \if@twocolumn
4559         \ifnum \col@number=\@ne
4560             \@maketitle
4561         \else
4562             \twocolumn[\@maketitle]%
4563         \fi
4564     \else
4565         \newpage
4566         \global\@topnum\z@
4567         \@maketitle
4568     \fi
4569     \thispagestyle{plain}\@thanks
4570 \endgroup
4571 \@bscontmark % \setcounter{footnote}{0}%
4572 %%% \bsmtitleempty
4573 } % end non-titlepage
4574
4575 \def\@maketitle{%
4576     \newpage
4577     \null
4578     \vskip 2em%
4579     \vspace*{\droptitle}
4580     \maketitlehooka
4581     \ifcsemtty{@published}
4582         {}
4583         {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4584     {\@bsprettitle \@title \@bsposttitle}
4585     \ifcsemtty{@subtitle}
4586         {}
4587         {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4588     \maketitlehookb
4589     {\@bspreauthor \@author \@bspostauthor}
4590     \maketitlehookc
4591     {\@bspredate \@date \@bspostdate}
4592     \maketitlehookd
4593     \par
4594     \vskip 1.5em}
4595 \fi
4596
4597 \providecommand{\maketitlehookaa}{}
4598
4599 \providecommand{\maketitlehookaaa}{}
4600
4601 \newcommand{\prepublished}[1]{%
4602 \def\@bsprepublished{#1}%
4603 }
4604
4605 \newcommand{\postpublished}[1]{%
4606 \def\@bspostpublished{#1}%

```

```

4607 }
4608
4609 \newcommand{\presubtitle}[1]{%
4610 \def\@bspresubtitle{#1}%
4611 }

```

`\presubtitle` Hook after printing the subtitle.

```

4612 \newcommand{\postsubtitle}[1]{%
4613 \def\@bspostsubtitle{#1}%
4614 }

```

Initial settings:

```

4615 \if@titlepage
4616 \prepublished{
4617 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
4618 \begin{center}}
4619 \postpublished{\par\end{center}\vskip 2em}
4620
4621 \presubtitle{\unskip\begin{center}\unskip}
4622 \postsubtitle{\par\end{center}\vskip 2em}
4623 \else
4624 \prepublished{\begin{center}}
4625 \postpublished{\par\end{center}\vskip 0.5em}
4626
4627 \presubtitle{\begin{center}\unskip}
4628 \postsubtitle{\par\end{center}\vskip 0.5em}
4629 \fi

4630 \end{warpprint}

```

53.6 `\maketitle` for HTML output

An HTML div of class `titlepage` is created, inside of which a \LaTeX PDF minipage is generated (without HTML tags), allowing the `\thanks` footnotes to be generated immediately at the end of the title page during HTML output. This is desirable when a large table of contents immediately follows the title.

`\thanks` are a form of footnotes used in the title page. See section 46 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4631 `\begin{warppHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use \LaTeX minipage footnotes.

```
4632 \newcommand*{\LWR@maketitlesetup}{%
```

Select which kind of footnote marks to use:

```
4633 \@bsmarkseries
```

```
4634 \@mpbsmarkseries
```

Redefine the footnote mark:

```
4635 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```
\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4636 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4637 \makethanksmark \LWR@origspace{1in}
```

```
\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4638 ##1%
```

```
4639 }%
```

```
4640 }
```

```
\@fnsymbol {⟨counter⟩}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

```
4641 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
```

```
4642 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
```

```
4643 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
```

```
4644 \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}
```

`\maketitle` Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
4645 \renewcommand*{\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
4646 \begin{titlepage}
```

Set up special patches:

```
4647 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4648 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4649 \@thanks
```

Close the HTML titlepage div:

```
4650 \end{titlepage}
```

Reset the footnote counter:

```
4651 \@bscontmark
```

```
4652 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```
4653 \DeclareDocumentCommand{\@maketitle}{-}{-%
4654   \maketitlehooka
4655   \ifcempty{@published}
4656     {}
4657     {{{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
4658     {\@bsprettitle \@title \@bsposttitle}
4659     \ifcempty{@subtitle}
4660     {}
4661     {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
4662   \maketitlehookb
4663   {\@bspreadauthor \@author \@bspostauthor}
4664   \maketitlehookc
4665   {\@bspredate \@date \@bspostdate}
4666   \maketitlehookd
4667 }
```

```
4668 \providecommand{\maketitlehookaa}{}
4669 \providecommand{\maketitlehookaaa}{}
```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
4670 \newcommand*{\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

```
4671 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4672 \@thanks
4673 }
```

`\thanksmarkseries` `{\series}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the \LaTeX PDF minipage which is included with the title page.

```
4674 \renewcommand{\thanksmarkseries}[1]{%
4675 \def\@mpbsmarkseries{%
4676 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}%
4677 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
4678 }
```

```
4679 \end{warpHTML}
```

54 Abstract

The following code replaces the \LaTeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: `4680 \begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the babel package.

```
4681 \providecommand*{\abstractname}{Abstract}
```

Env `abstract`

```

4682 \DeclareDocumentEnvironment{abstract}{}
4683 {
4684 \LWR@forcenewpage
4685 \BlockClass{abstract}
4686 \BlockClassSingle{abstracttitle}{\abstractname}
4687 }
4688 {
4689 \endBlockClass
4690 }

4691 \end{warpHTML}

```

55 Quote and verse

55.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

for HTML output:

```

4692 \begin{warpHTML}
4693 \newcommand{\attribution}[1]{%
4694 \InlineClass{attribution}{--\,#1}}% emdash
4695 \end{warpHTML}

```

for PRINT output:

```

4696 \begin{warpprint}
4697 \newcommand{\attribution}[1]{\textsc{---\,#1}}
4698 \end{warpprint}

```

`\citetitle` for use inside quote, quotation, verse:

for HTML output:

```

4699 \begin{warpHTML}
4700 \newcommand{\citetitle}[1]{%
4701 \InlineClass{citetitle}{--\,#1}}% emdash
4702 \end{warpHTML}

```

for PRINT output:

```

4703 \begin{warpprint}
4704 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
4705 \end{warpprint}

```

55.2 Quotes, quotations

for HTML output: 4706 `\begin{warpHTML}`

Env `quote`

```

4707 \renewenvironment*{quote}
4708 {
4709 \LWR@forcenewpage
4710 \LWR@htmlblocktag{blockquote}
4711 }
4712 {\LWR@htmlblocktag{/blockquote}}
4713
4714 \renewenvironment*{quotation}
4715 {
4716 \LWR@forcenewpage
4717 \LWR@htmlblocktag{blockquotation}
4718 }
4719 {\LWR@htmlblocktag{/blockquotation}}

4720 \end{warpHTML}

```

55.3 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` 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.

Len `\leftmargini`

Len `\TMLvleftskip`

Len `\TMLleftmargini`

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: 4721 `\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.

```
4722 \newlength{\TMLvleftskip}
4723 \setlength{\TMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```
4724 \newlength{\TMLleftmargini}
4725 \setlength{\TMLleftmargini}{4.5em}
```

```
4726 \end{warpall}
```

56 Verbatim

for HTML & PRINT: 4727 `\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.

```
4728 \newlength{\VerbatimHTMLWidth}
4729 \setlength{\VerbatimHTMLWidth}{4in}
4730 \end{warpall}
```

for HTML output: 4731 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4732 \newbool{LWR@verbtags}
4733 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*style*] {*class*}

Encloses a verbatim environment with the given css class.

```
4734 \newcommand*{\LWR@atbeginverbatim}[2] []
4735 {%
```

Avoid excessive space between lines:

```
4736 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4737 \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.

```
4738 \ifbool{LWR@verbtags}{%
4739   \LWR@htmltag{pre class="#2"
4740   \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
4741 }% pre
4742 }{}%
```

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.

Also turn off babel-french extra space before punctuation:

```
4743 \begingroup%
4744 \LWR@origttfamily%
4745 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a verse par. Restore plain \TeX `\hspace` functionality:

```
4746 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4747 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4748 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
4749 \endgroup%
4750 \unskip%
```

At the end of the environment, close the pre:

```

4751 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
4752
4753 }{ }%

```

Resume regular paragraph handling:

```

4754 \LWR@startpars%
4755 }

```

`\verbatiminput` $\langle filename \rangle$

Patch `\verbatiminput` to add HTML tags:

```

4756 \let\LWRV@origverbatim@input\verbatim@input
4757
4758 \renewcommand{\verbatim@input}[2]{%
4759 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{ }%
4760 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4761 \LWRV@origverbatim@input{#1}{#2}%
4762 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4763 }

```

Env `verbatim`

```

4764 \AfterEndPreamble{
4765 \LWR@traceinfo{Patching verbatim.}
4766 \AtBeginEnvironment{verbatim}{%
4767 \LWR@forcenewpage
4768 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4769 }
4770 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4771 }

4772 \end{warpHTML}

```

57 Theorems

`\newtheorem` $\langle \langle text \rangle \rangle$ [$\langle counter \rangle$] -or- [$\langle oldname \rangle$] $\langle \langle text \rangle \rangle$

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: 4773 `\begin{warpHTML}`

`\@begintheorem` $\langle name \rangle$ $\langle number \rangle$

```
4774 \renewcommand{\@begintheorem}[2]{%
4775 \LWR@forcenewpage
4776 \BlockClass{theoremcontents}
4777 \InlineClass{theoremlabel}{#1\ #2\ }
4778 }
```

`\@opargbegintheorem` $\langle name \rangle$ $\langle number \rangle$ $\langle oparg \rangle$

```
4779 \renewcommand{\@opargbegintheorem}[3]{%
4780 \LWR@forcenewpage
4781 \BlockClass{theoremcontents}
4782 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4783 }
```

`\@endtheorem`

```
4784 \renewcommand*\@endtheorem{%
4785 \endBlockClass% theoremcontents
4786 }
```

```
4787 \end{warpHTML}
```

58 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard \TeX 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 \TeX 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}

```

58.1 Itemize

for HTML output: 4788 \begin{warpHTML}

\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.

```

4789 \newcommand*\LWR@itemizeitem}{%
4790 \LWR@stoppars%
4791 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
4792 \LWR@htmltag{li}%
4793 \LWR@startpars%
4794 \LWR@origitem%
4795 }

```

Env itemize [*<options>*]

```

4796 \newcommand*\LWR@itemizestart}{%
4797 \LWR@stoppars%
4798 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4799 \LWR@htmltag{ul style="list-style-type:none"}%
4800 \LWR@startpars%
4801 \let\item\LWR@itemizeitem%
4802 }
4803
4804 \newcommand*\LWR@itemizeend}{%
4805 \LWR@stoppars%
4806 \LWR@closeprevious{\LWR@depthlistitem}%
4807 \LWR@closeoneprevious{}%
4808 \LWR@startpars%
4809 }

```

58.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env `enumerate` [*options*]

```

4810 \newcommand*\LWR@enumeratestart}{%
4811 \LWR@stoppars%
4812 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4813 \LWR@htmltag{ul style="list-style-type:none"}%
4814 \LWR@startpars%
4815 \let\item\LWR@itemizeitem%
4816 }
4817
4818 \newcommand*\LWR@enumerateend}{%
4819 \LWR@stoppars%
4820 \LWR@closeprevious{\LWR@depthlistitem}%
4821 \LWR@closeoneprevious{}%
4822 \LWR@startpars%
4823 }
```

58.3 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```

4824 \newcommand*\LWR@descitem}[1]{}%
4825 {%
4826 \LWR@stoppars%
4827 \LWR@setlatestname{#1}%
4828 \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.

```

4829 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original `\item` code:

```

4830 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```

4831 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
4832 \LWR@originspace{1in}
4833 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4834 \LWR@orignewline%
4835 \LWR@htmltag{dd}%
4836 \LWR@startpars%
4837 }
```

Env description [*options*]

```
4838 \newcommand*\LWR@descriptionstart}{%
4839 \LWR@stoppars%
4840 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4841 \LWR@htmltag{dl}%
4842 \LWR@startpars%
4843 \let\item\LWR@descitem%
4844 }
4845
4846 \newcommand*\LWR@descriptionend}{%
4847 \LWR@stoppars%
4848 \LWR@closeprevious{\LWR@depthlistitem}%
4849 \LWR@closeoneprevious{}}%
4850 \LWR@startpars%
4851 }
```

`\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 \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4852 \newcommand*\LWR@patchlists}{%
4853 \LetLtxMacro{\LWR@origitem}{\item}
4854 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4855 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4856 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4857 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4858 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4859 \AtEndEnvironment{description}{\LWR@descriptionend}
4860 }
```

```
4861 \end{warpHTML}
```

59 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include `tabular` plus 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:

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

column types

- Vertical rules are not yet supported.
- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@`, `!`, `>`, and `<` may be used at each column, and they are used in that order.
- `\newcolumnstype` is ignored; unknown column types are set to `l`.
- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.

⚠ `\multirow &`
`\multicolumn`

- `Multirow` and `multicolumn` cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)

⚠ `\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 with rules`

- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. This may be a browser bug.

rule at last row

- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.

⚠ paragraphs

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

`\cmidrule width, trim`

- `\cmidrule` does not support width or trim options due to CSS limitations.

`\cmidrule and`
`\multicolumn`

- `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.

longtable headings

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

⚠ `\warpprintonly`

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` 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 relevant to HTML output.

⚠ `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 \\}
```

```
\warppHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

59.1 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 4862 `\begin{warppHTML}`

`\LWR@mynexttoken`

```
4863 \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`

```
4864 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
4865 \afterassignment\LWR@fnslone\let\nexttoken= }
4866 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4867 \def\LWR@fnsltwo{%
4868 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
```

```
4869 \else\let\next=\nexttoken\fi\next}
4870 \def\LWR@fnslnthree{\afterassignment\LWR@fnslnone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
4871 \newcommand*\LWR@getmynexttoken}{%
4872 \LWR@traceinfo{\LWR@getmynexttoken}%
4873 % nothing must follow this next line
4874 \LWR@futurenonspacel\let\LWR@mynexttoken\LWR@tabledatacolumn\tag
4875 }
```

59.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
4876 \newbool{LWR@startedrow}
4877 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
4878 \newbool{LWR@doinghline}
4879 \boolfalse{LWR@doinghline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
4880 \newbool{LWR@doingtbrule}
4881 \boolfalse{LWR@doingtbrule}
```

Bool `LWR@tableparcell` True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
4882 \newbool{LWR@tableparcell}
```

Bool `LWR@skippingmrowcell` True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4883 \newbool{LWR@skippingmrowcell}
```

Bool `LWR@skipatbang` True if just finished a `\multicolumn` so should not print the trailing @ or ! columns.

```
4884 \newbool{LWR@skipatbang}
```

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.

```
4885 \newbool{LWR@intabularmetadata}
4886 \boolfalse{LWR@intabularmetadata}
```

59.3 Handling &, @, and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/>

[where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860](http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860)

`\LWR@closetabledatcell` If `LWR@skippingmrowcell` then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
4887 \newcommand*\LWR@closetabledatcell}{%
4888 \global\booltrue\LWR@intabularmetadata}%
4889 \ifbool\LWR@exitingtabular}{}%
4890 {% not exiting tabular
4891   \ifbool\LWR@skippingmrowcell}{}%
4892   {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
4893   \unskip%
4894   \LWR@getexparray\LWR@colafterspec{\theLWR@tablecolspos}%
4895 % \LWR@getexparray\LWR@colatspec{\theLWR@tablecolspos}%
4896 % \LWR@printatbang{at}\theLWR@tablecolspos}%
4897 % \LWR@getexparray\LWR@colbangspec{\theLWR@tablecolspos}%
4898 % \LWR@printatbang{bang}\theLWR@tablecolspos}%
```

Close paragraphs:

```
4899   \ifbool\LWR@tableparcell{\LWR@stoppars}{}%
4900   \global\boolfalse\LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
4901   \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4902   \ifbool\LWR@skipatbang}%
4903   {\boolfalse\LWR@skipatbang}}%
4904   {%
4905     \LWR@printatbang{at}\theLWR@tablecolspos}%
4906     \LWR@printatbang{bang}\theLWR@tablecolspos}%
4907   }% not skipping at or bang
4908 }% not skipping mrowcell
4909 }% not exiting tabular
4910 \global\boolfalse\LWR@skippingmrowcell}%
4911 }
```

`LWR@tabulardepth` tracks whether & is being used inside a tabular.

```
4912 \newcounter\LWR@tabulardepth
```

```
4913 \setcounter{LWR@tabulardepth}{0}
4914
```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```
4915 \let\LWR@origampmacro&
4916 \end{warpHTML}
```

59.3.1 Localizing & catcodes

for HTML & PRINT: 4917 \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.

```
4918 \newcommand{\StartDefiningTabulars}{%
4919 \warpHTMLonly{\catcode'\&=\active}%
4920 }
```

\EndDefiningTabulars Place after defining something with & in it.

```
4921 \newcommand{\EndDefiningTabulars}{%
4922 \warpHTMLonly{\catcode'\&=4}%
4923 }
```

```
4924 \end{warpall}
```

59.3.2 Handling &

for HTML output: 4925 \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.

```

4926 \newcommand*{\LWR@tabularampersand}{%
4927 \LWR@traceinfo{\LWR@tabularampersand}%
4928 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}%
4929 {%

```

If not skipping a multirow cell, close the current data cell.

```

4930 \unskip%
4931 \LWR@closetabledatacell%

```

Move to the next column.

```

4932 \addtocounter{\LWR@tablecolspos}{1}%

```

Look at the next token to decide multi or single column data tag.

```

4933 \LWR@getmynexttoken%
4934 }%

```

If not inside a tabular, performs the original action:

```

4935 {\LWR@origampmacro}%
4936 }

```

& 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.

59.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```

4937 \NewDocumentCommand{\LWR@tabularendofline}{s o}
4938 {%
4939 \LWR@closetabledatacell%

```

Finish the previous row:

```
4940 \LWR@htmltag{/tr}\LWR@orignewline
4941 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
4942 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```
4943 \global\boolfalse{LWR@doinghline}%
4944 \global\boolfalse{LWR@doingtbrule}%
4945 \LWR@clearmidrules%
```

Start at first column:

```
4946 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
4947 \LWR@getmynexttoken%
4948 }
```

59.5 Variables

```
4949 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4950 \newcommand*{\LWR@pposition}{}
4951 \newcommand*{\LWR@pleft}{}
4952 \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 pipe.

This is indexed by the counter `LWR@tabletotalcols`.

```
4953 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
4954 \newcommand*{\LWR@strresult}{}
4955 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

4956 `\newcommand*{\LWR@origcolspec}{}`

Ctr `LWR@tablecolwidth` Holds the width of the table columns specification.

This is the number of tokens, including one for each @ etc. column, and also one each for the parameters of p, @, etc. columns, and three for each D column.

(This is not the total # of \TeX columns in the table.)

4957 `\newcounter{LWR@tablecolwidth}`

Ctr `LWR@tablecolspos` Where are currently looking into the table column specification. Index starts at 1.

4958 `\newcounter{LWR@tablecolspos}`

Ctr `LWR@tabletotalcols` Holds the final number of table columns, not counting @ and ! columns.

4959 `\newcounter{LWR@tabletotalcols}`

Ctr `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

4960 `\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 33.

`LWR@colbangspec` A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 33.

`LWR@colbefore-spec` A data array of specifications for > columns.

`LWR@colafterspec` A data array of specifications for < columns.

59.6 Parsing @, >, <, ! columns

Holds the parsed argument for @, >, <, or ! columns:

4961 `\newcommand*{\LWR@colparameter}{}`

`\LWR@parseatcolumn` Handles `@{text}` columns.

4962 `\newcommand*{\LWR@parseatcolumn}{%`

Move to the next token after the '@':

```
4963 \LWR@traceinfo{at column}%
4964 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4965 \LWR@traceinfo{about to read the next token:}%
4966 \expandarg%
4967 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4968 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4969 \LWR@traceinfo{have now read the next token}%
4970 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
4971 {% left edge of the table:
4972   \LWR@traceinfo{at the left edge}%
4973   \LWR@setexparray{LWR@colatspec}{\leftedge}{\LWR@colparameter}%
4974   \LWR@traceinfo{at the left edge: %
4975   \LWR@getexparray{LWR@colatspec}{\leftedge}}%
4976 }%
4977 {% not at the left edge:
4978   \LWR@traceinfo{not at the left edge}%
4979   \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4980   \LWR@traceinfo{at \theLWR@tabletotalcols: %
4981   \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}%
4982 }%
4983 \let\LWR@colparameter\relax%
4984 \booltrue{LWR@validtablecol}%
4985 }
```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```
4986 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
4987 \LWR@traceinfo{bang column}%
4988 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4989 \LWR@traceinfo{about to read the next token:}%
4990 \expandarg%
4991 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4992 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4993 \LWR@traceinfo{have now read the next token}%
4994 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
4995 {% left edge of the table:
4996   \LWR@traceinfo{at the left edge}%
4997   \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4998 }%
4999 {% not at the left edge:
5000   \LWR@traceinfo{not at the left edge}%
5001   \LWR@setexparray{LWR@colbangspec}{theLWR@tabletotalcols}{\LWR@colparameter}%
5002   \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
5003 }%
5004 \let\LWR@colparameter\relax%
5005 \booltrue{LWR@validtablecol}%
5006 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```
5007 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
5008 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```

5009 \expandarg%
5010 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
5011 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

5012 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@colparameter}%
5013 \let\LWR@colparameter\relax%
5014 \booltrue{LWR@validtablecol}%
5015 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```
5016 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
5017 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
5018 % \StrChar{#1}{\theLWR@tablecolspec}[\LWR@colparameter]
5019 \expandarg%
5020 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspec}[\LWR@colparameter]%
5021 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
5022 \LWR@setexparray{\LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
5023 \let\LWR@colparameter\relax%
5024 \booltrue{\LWR@validtablecol}%
5025 }
```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```
5026 \newcommand*{\LWR@parseskipcolumn}{%
5027 \booltrue{\LWR@validtablecol}%
5028 }
```

59.7 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.

```
5029 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5030 \appto\LWR@tablecolspec{#1}%
5031 \addtocounter{\LWR@tabletotalcols}{1}%
5032 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5033 \LWR@traceinfo{normal column \theLWR@tabletotalcols: #1}%
5034 \LWR@setexparray{\LWR@colatspec}{\theLWR@tabletotalcolsnext}{}%
5035 \LWR@setexparray{\LWR@colbangspec}{\theLWR@tabletotalcolsnext}{}%
5036 \LWR@setexparray{\LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{}%
5037 \LWR@setexparray{\LWR@colafterspec}{\theLWR@tabletotalcolsnext}{}%
5038 \booltrue{\LWR@validtablecol}%
5039 }
```

59.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` `{\thiscolumn}` The width will be ignored.

```
5040 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
5041 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
5042 \addtocounter{LWR@tablecolspos}{1}%
5043 }
```

59.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```
5044 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
5045 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
5046 \addtocounter{LWR@tablecolspos}{3}%
5047 }
```

59.10 Parsing the column specifications

 HTML CSS cannot exactly match the \LaTeX concept of a baseline for a table row. Table 6 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 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

```
5048 \newbool{LWR@validtablecol}
```

`\LWR@parsetablecols` $\{ \langle colspecs \rangle \}$

Scans the column specification left to right.

Table 6: Tabular baseline

| | | | | |
|---|-----|-----|-----|---|
| l | p | m | b | r |
| | | | bot | |
| | | mid | bot | |
| l | par | mid | bot | r |
| | par | mid | | |
| | par | | | |

Table 7: 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 6, \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.
- Unknown:** Converted to 'l'.
- \newcolumn:** Currently treated as unknown.
-

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

```
5049 \newcommand*\LWR@parsetablecols}[1]{%
5050 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
5051 \renewcommand*\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
5052 \renewcommand*\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5053 \setcounter{LWR@tabletotalcols}{0}%
5054 \setcounter{LWR@tabletotalcolsnext}{1}%
5055 \LWR@setexparray{LWR@colatspec}{leftedge}{}%
5056 \LWR@setexparray{LWR@colatspec}{1}{}%
5057 \LWR@setexparray{LWR@colatspec}{2}{}%
5058 \LWR@setexparray{LWR@colatspec}{3}{}%
5059 \LWR@setexparray{LWR@colbangspec}{leftedge}{}%
5060 \LWR@setexparray{LWR@colbangspec}{1}{}%
5061 \LWR@setexparray{LWR@colbangspec}{2}{}%
5062 \LWR@setexparray{LWR@colbangspec}{3}{}%
5063 \LWR@setexparray{LWR@colbeforespec}{1}{}%
5064 \LWR@setexparray{LWR@colbeforespec}{2}{}%
5065 \LWR@setexparray{LWR@colbeforespec}{3}{}%
5066 \LWR@setexparray{LWR@colafterspec}{1}{}%
5067 \LWR@setexparray{LWR@colafterspec}{2}{}%
5068 \LWR@setexparray{LWR@colafterspec}{3}{}%
```

Starting at the first column specification:

```
5069 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
5070 \LWR@traceinfo{about to StrLen}%
5071 \noexpandarg%
5072 \StrLen{#1}[\LWR@strresult]%
5073 \fullexpandarg%
5074 \LWR@traceinfo{finished StrLen}%
5075 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
5076 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%
```

Place the next single-character column type into `\LWR@strresult`:

```
5077 \noexpandarg%
5078 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
5079 \LWR@traceinfo{position \arabic{LWR@tablecolspos}: \LWR@strresult}%
5080 \fullexpandarg%
```

Not yet found a valid column type

```
5081 \boolfalse{LWR@validtablecol}%
```



Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.

```
5082 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
5083 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
5084 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
5085 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
5086 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
5087 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
5088 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
5089 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5090 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5091 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5092 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5093 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5094 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
5095 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5096 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5097 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

From the `dcolumn` package:

```
5098 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the `tabularx` package. `X` column has no parameter, but will be given paragraph tags.

```
5099 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnntype{P}[1]{>\centering\arraybackslash}p{#1}}
```

```

5100 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
5101 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5102 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an 1 column:

```

5103 \ifbool{LWR@validtablecol}{}{}%
5104   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5105   \LWR@parsenormalcolumn{1}%
5106 }%
5107 \addtocounter{LWR@tablecolspos}{1}%
5108 }%
5109 }%

```

59.11 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.

```

5110 \newcommand*{\LWR@maybenewtablerow}
5111 {%
5112   \ifbool{LWR@startedrow}%
5113   {}% started the row
5114   {}% not started the row

```

Remember that now have started the row:

```

5115   \global\booltrue{LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

5116   \global\booltrue{LWR@intabularmetadata}%
5117   \ifbool{LWR@doinghline}%
5118   {\LWR@htmltag{tr class="hline"{} }\LWR@orignewline}%
5119   {% not doing hline
5120     \ifbool{LWR@doingtbrule}%
5121     {\LWR@htmltag{tr class="tbrule"{} }\LWR@orignewline}%
5122     {\LWR@htmltag{tr} \LWR@orignewline}%
5123   }% end of not doing hline
5124 }% end of not started the row
5125 }

```

59.12 Printing at or bang tags

`\LWR@printatbang` $\langle\{at\ -or\ -bang\}\rangle$ $\langle\{index\}\rangle$

```

5126 \newcommand*{\LWR@printatbang}[2]{%
5127 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}
5128 \LWR@traceinfo{atbang: !\LWR@atbangspec!}
5129 \ifdefempty{\LWR@atbangspec}%
5130 % \ifthenelse{\isempty{\LWR@atbangspec}}%
5131 {}%
5132 {%
5133   \LWR@htmltag{td class="td#1%
5134     \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}}%
5135   "%}
5136   \LWR@atbangspec%
5137   \LWR@htmltag{/td}\LWR@orignewline%
5138 }%
5139 }%

```

59.13 Data opening tag

`\LWR@tabledatasinglecolumn` Print a table data opening tag with style for alignment

```

5140 \newcommand*{\LWR@tabledatasinglecolumn}{%
5141 {%
5142 \LWR@maybe newtablerow%

```

Don't start a new paragraph if have already started one:

```

5143 \ifbool{\LWR@intabularmetadata}%
5144 {%

```

If have found the end of tabular command, do not create the next data cell:

```

5145   \ifbool{\LWR@exitingtabular}{}%
5146   {% not exiting tabular

```

Print the @ and ! contents before first column:

```

5147     \ifthenelse{\cinttest{\value{\LWR@tablecolspos}}=1}{%
5148     {%
5149       \LWR@printatbang{at}{leftedge}%
5150       \LWR@printatbang{bang}{leftedge}%
5151     }% left edge
5152     }% not left edge

```

Fetch the current column's alignment character into `\LWR@strresult`:

```
5153      \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```
5154      \LWR@htmltag{td class="td%
```

append this column's spec:

```
5155      \LWR@strresult%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
5156      \ifthenelse{\equal{\LWR@getexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}%
5157      {rule}%
5158      {}%
5159      "{}}%
```

If this is a p, m, b, or X column, allow paragraphs:

```
5160      \ifthenelse{%
5161      \equal{\LWR@strresult}{p}\OR%
5162      \equal{\LWR@strresult}{m}\OR%
5163      \equal{\LWR@strresult}{b}\OR%
5164      \equal{\LWR@strresult}{P}\OR%
5165      \equal{\LWR@strresult}{M}\OR%
5166      \equal{\LWR@strresult}{B}\OR%
5167      \equal{\LWR@strresult}{X}%
5168      }%
5169      {% allow pars
5170      \LWR@startpars%
5171      \global\booltrue{\LWR@tableparcell}%
5172      }% allow pars
5173      {}% no pars
```

Print the > contents:

```
5174      \LWR@getexpparray{\LWR@colbefore-spec}{\theLWR@tablecolspos}%
5175      \global\boolfalse{\LWR@intabularmetadata}%
5176      }% not exiting tabular
5177      {}% in tabular metadata
5178      }%
```

59.14 Midrules

`\LWR@midrules` `\LWR@midrules` is a data array (section 33) of columns containing Y if a midrule should

be created for each column.

Ctrl LWR@midrulecounter Indexes across the LWR@midrules data array.

```
5179 \newcounter{LWR@midrulecounter}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```
5180 \newcommand*{\LWR@clearmidrules}
5181 {%
5182 \setcounter{LWR@midrulecounter}{1}%
5183 \whiledo{%
5184 \cnttst{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
5185 }%
5186 {%
5187 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{}%
5188 \addtocounter{LWR@midrulecounter}{1}%
5189 }%
5190 }
```

\LWR@subcmidrule [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```
5191 \newcommand*{\LWR@subcmidrule}[4]{%
5192 \setcounter{LWR@midrulecounter}{#3}%
5193 \whiledo{\cnttst{\value{LWR@midrulecounter}}{<=}{#4}}%
5194 {%
5195 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
5196 \addtocounter{LWR@midrulecounter}{1}%
5197 }% end of the whiledo
5198 }
```

\LWR@docmidrule [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns.

```
5199 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
5200 {\LWR@subcmidrule{#1}{#2}{#3}}
```

59.15 Multicolumns

59.15.1 Parsing multicolumns

```
5201 \newcounter{LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
5202 \newcounter{LWR@tablemulticolspos}
```

`\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.

```
5203 \newcommand*{\LWR@printmccoltype}[1]{%
5204 \LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
5205 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
5206 \IfStrEq{\LWR@strresult}{l}{l}{}%
5207 \IfStrEq{\LWR@strresult}{c}{c}{}%
5208 \IfStrEq{\LWR@strresult}{r}{r}{}%
5209 \IfStrEq{\LWR@strresult}{p}{p}{}%
5210 \IfStrEq{\LWR@strresult}{m}{m}{}%
5211 \IfStrEq{\LWR@strresult}{b}{b}{}%
5212 \IfStrEq{\LWR@strresult}{P}{P}{}%
5213 \IfStrEq{\LWR@strresult}{M}{M}{}%
5214 \IfStrEq{\LWR@strresult}{B}{B}{}%
5215 \IfStrEq{\LWR@strresult}{S}{r}{}%
5216 \IfStrEq{\LWR@strresult}{X}{p}{}%
5217 \LWR@traceinfo{lwr@printmccoltype done}%
5218 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
5219 \newcommand*{\LWR@multicolpartext}{%
5220 \LWR@startpars%
5221 \LWR@multicoltext%
5222 \LWR@stoppars%
5223 }
```

`\LWR@multicolother` `{<colspec>}` For @, !, >, <, print the next token without paragraph tags:

```
5224 \newcommand*{\LWR@multicolother}[1]{%
5225 \addtocounter{LWR@tablemulticolspos}{1}%
5226 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
5227 \LWR@strresult%
```

A valid column data type was found:

```
5228 \booltrue{LWR@validtablecol}%
5229 }
```

\LWR@multicolskip Nothing to print for this column type.

```
5230 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
5231 \booltrue{LWR@validtablecol}%
5232 }
```

\LWR@printmccoldata $\langle\{colspec}\rangle$ Print the data for any valid column type found.

```
5233 \newcommand*{\LWR@printmccoldata}[1]{%
5234 \LWR@traceinfo{lw@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5235 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
5236 \StrChar{#1}{\theLWR@tablemulticolspos}[LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5237 \IfStrEq{LWR@strresult}{l}{LWR@multicoltext}{}%
5238 \IfStrEq{LWR@strresult}{c}{LWR@multicoltext}{}%
5239 \IfStrEq{LWR@strresult}{r}{LWR@multicoltext}{}%
5240 \IfStrEq{LWR@strresult}{D}{}%
5241 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
5242 \LWR@multicoltext%
5243 }{}%
5244 \IfStrEq{LWR@strresult}{p}{LWR@multicolparttext}{}%
5245 \IfStrEq{LWR@strresult}{m}{LWR@multicolparttext}{}%
5246 \IfStrEq{LWR@strresult}{b}{LWR@multicolparttext}{}%
5247 \IfStrEq{LWR@strresult}{P}{LWR@multicolparttext}{}%
5248 \IfStrEq{LWR@strresult}{M}{LWR@multicolparttext}{}%
5249 \IfStrEq{LWR@strresult}{B}{LWR@multicolparttext}{}%
5250 \IfStrEq{LWR@strresult}{S}{LWR@multicolparttext}{}%
5251 \IfStrEq{LWR@strresult}{X}{LWR@multicolparttext}{}%
5252 \IfStrEq{LWR@strresult}{|}{LWR@multicolskip}{}%
5253 \IfStrEq{LWR@strresult}{\detokenize{@}}{LWR@multicolother{#1}}{}%
```

```

5254 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
5255 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
5256 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%

```

If an invalid column type:

```
5257 \ifbool{LWR@validtablecol}{\LWR@multicoltext}{}%
```

Tracing:

```

5258 \LWR@traceinfo{lwr@printmccoldata done}%
5259 }

```

`\parsemulticolumnalignment` $\langle 1: \text{colspec} \rangle \langle 2: \text{printresults} \rangle$

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.

```

5260 \newcommand*\LWR@parsemulticolumnalignment}[2]{}%
5261 \setcounter{LWR@tablemulticolspos}{1}%
5262 \StrLen{#1}[\LWR@strresult]%
5263 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

5264 \whiledo{%
5265 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
5266 }%
5267 {}%

```

Execute the assigned print function for each token in the column spec:

```
5268 #2{#1}%
```

Move to the next token in the column spec:

```

5269 \addtocounter{LWR@tablemulticolspos}{1}%
5270 }%
5271 }

```

59.15.2 High-level multicolumn interface

```
5272 \newcommand{\LWR@multicoltext}{}%
```

```
\LWR@domulticolumn {<1: numcols>} {<2: colspec>} {<3: text>}
```

```
5273 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
5274 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5275 \renewcommand{\LWR@multicoltext}{%
5276 #3%
5277 \booltrue{LWR@validtablecol}%
5278 }%
```

Row processing:

```
5279 \LWR@maybe newtable row%
```

Begin the opening table data tag:

```
5280 \LWR@htmltag{td colspan="#1"
5281 class="td%
```

Print the column type:

```
5282 \LWR@parsemulticolumnalignment{#2}{\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”.

```
5283 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
5284 "%
5285 }% end of the opening table data tag
5286 \global\boolfalse{LWR@intabularmetadata}%
5287 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
5288 }
```

59.15.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
5289 \newbool{LWR@starredlongtable}
5290 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5291 \providecommand*\LTcapttype{table}
```

```
\LWR@longtabledatacaptiontag * [⟨toc entry⟩] {⟨caption⟩}
```

```
5292 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5293 {%
```

Remember the latest name for `\nameref`:

```
5294 \IfValueTF{#2}{% optional given?
5295   \ifthenelse{\equal{#2}{}}{% optional empty?
5296     {\LWR@setlatestname{#3}}% empty
5297     {\LWR@setlatestname{#2}}% given and non-empty
5298 }% optional given
5299 {\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:

```
5300 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5301 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{% \LWR@domulticolumn
5302 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5303 {% yes star
5304   \LWR@htmlblocktag{figcaption}%
5305   #3%
5306   \LWR@htmlblocktag{/figcaption}%
5307 }%
5308 {% No star:
```

Not the star version:

Don't step the counter if `\caption[] {A caption.}`

```
5309   \ifbool{LWR@starredlongtable}%
5310   {%
5311   \ifthenelse{\equal{#2}{}}% TOC entry
5312   {%
5313   {%
```

```

5314     \refstepcounter{\LTcaption}%
5315     \protected@edef\@currentlabel{%
5316     \csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5317     }%
5318     }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

5319     \LWR@htmlblocktag{figcaption}%
5320     \csuse{fnum@\LTcaption}\CaptionSeparator#3%
5321     \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

5322     \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

5323     {}%

```

If the optional caption was given, but might only be []:

```

5324     {% TOC entry not empty
5325     \IfNoValueTF{#2}% No TOC entry?

```

The optional caption is []:

```

5326     {% No TOC entry
5327     \addcontentsline%
5328     {\csuse{ext@\LTcaption}}%
5329     {\LTcaption}%
5330     {%
5331     \protect\numberline%
5332     {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5333     {\ignorespaces #3\protect\relax}%
5334     }%
5335     }% end of No TOC entry

```

The optional caption has text enclosed:

```

5336     {% yes TOC entry
5337     \addcontentsline%
5338     {\csuse{ext@\LTcaption}}%
5339     {\LTcaption}%
5340     {%
5341     \protect\numberline%
5342     {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5343     {\ignorespaces #2\protect\relax}%

```

```

5344         }%
5345     }% end of yes TOC entry
5346 }% end of TOC entry not empty
5347 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

5348 \booltrue{LWR@skipatbang}%
5349 }% end of \LWR@domulticolumn
5350
5351 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5352 \addtocounter{LWR@tablecolspos}{-1}
5353
5354 }

```

59.15.4 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.

```

5355 \newcounter{LWR@tabhtmlcolindex}
5356 \newcounter{LWR@tabhtmlcolend}
5357 \newcounter{LWR@tabhtmlcoltotal}

```

```
\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`.

```
5358 \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:

```

5359 \setcounter{LWR@tabhtmlcolindex}{#1}%
5360 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5361 \setcounter{LWR@tabhtmlcolend}{#1}%

```

```
5362 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the \LaTeX columns looking for @ and ! columns:

```
5363 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5364 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{\theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5365 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5366 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{\theLWR@tabhtmlcolindex}}%
```

```
5367 \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next \LaTeX column:

```
5368 \addtocounter{LWR@tabhtmlcolindex}{1}%
```

```
5369 }%
```

If at the left-most column, also skip the leftmost @ and ! cells:

```
5370 \ifthenelse{\value{LWR@tablecolspos}=1}{%
```

```
5371   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{leftedge}}%
```

```
5372   \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

```
5373   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{leftedge}}%
```

```
5374   \ifdefempty{\LWR@atbangspec}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

```
5375 }{}%
```

```
5376 }
```

59.15.5 \tabledatamulticolumntag

```
\LWR@tabledatamulticolumntag {<numcols>} {<alignment>} {<text>}
```

```
5377 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
```

```
5378 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5379 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5380 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next \LaTeX column:

```
5381 \addtocounter{LWR@tablecolspos}{#1}%
```

```
5382 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5383 \booltrue{LWR@skipatbang}%
```

```
5384 }
```

59.16 Multirow

Pkg multirow

```
\LWR@tabledatamultirowtag {<numrows>} [<bigstruts>] {<width>} [<fixup>] {<text>}
```

```
5385 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
```

```
5386 {%
```

```
5387 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
5388 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
5389 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
5390 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

```
5391 "{}}
```

While printing the text, redefine `\` to generate a new line

```
5392 \begingroup \LetLtxMacro{\}\{\LWR@endoffline} #5 \endgroup
```

```
5393 \LWR@stoppars%
```

```
5394 \global\boolfalse{LWR@intabularmetadata}%
```

```
5395 }%
```

59.17 Utility macros inside a table

```
5396 \newcommand*\LWR@donothing{}
5397 \newcommand*\LWR@domidrule{\booltrue\LWR@doinghline}}
5398 \newcommand*\LWR@dotbrule{\booltrue\LWR@doingtbrule}}
```

59.18 Checking for a new table cell

`\LWR@tabledatacolumnstag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
5399 \newbool\LWR@exitingtabular}
5400
5401 \newcommand*\LWR@tabledatacolumnstag}%
5402 {%
5403 \LWR@traceinfo\LWR@tabledatacolumnstag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
5404 \let\mynext\LWR@tabledatasinglecolumnstag%
```

If exiting the tabular:

```
5405 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}%
5406 {\booltrue\LWR@exitingtabular}}{}}%
```

`longtable` can have a caption in a cell

```
5407 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}%
5408 {\let\mynext\LWR@donothing}}{}}%
```

Look for other things which would not start a table cell:

```
5409 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}%
5410 {\let\mynext\LWR@donothing}}{}}%
5411 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}%
5412 {\let\mynext\LWR@donothing}}{}}%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
5413 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}%
5414 {\let\mynext\LWR@donothing}}{}}%
5415 %
5416 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}%
```

```

5417 {\let\mynext\LWR@donothing}{}%
5418 %
5419 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}%
5420 {\let\mynext\LWR@donothing}{}%
5421 %
5422 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}%
5423 {\let\mynext\LWR@donothing}{}%
5424 %
5425 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}%
5426 {\let\mynext\LWR@donothing}{}%
5427 %
5428 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\specialrule}}%
5429 {\let\mynext\LWR@donothing}{}%
5430 %
5431 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}%
5432 {\let\mynext\LWR@donothing}{}%
5433 %
5434 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}%
5435 {\let\mynext\LWR@donothing}{}%
5436 %
5437 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}%
5438 {\let\mynext\LWR@donothing}{}%
5439 %
5440 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}%
5441 {\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:

```

5442 \mynext%
5443 }

5444 \end{warpHTML}

```

59.19 `\mrowcell`

`\mrowcell`  The user must insert `\mrowcell` into any multirow cells which must be skipped. This command has no action during print output.

for HTML & PRINT:

```

5445 \begin{warpall}
5446 \newcommand*{\mrowcell}{}
5447 \end{warpall}

```

59.20 New \tabular definition

for HTML output: 5448 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to used, but must exist as placeholders.

```
5449 \newcommand*\LWR@origtoprule}[1]{\hline}
5450 \newcommand*\LWR@origmidrule}[1]{\hline}
5451 \LetLtxMacro\LWR@origcmidrule\cline
5452 \newcommand*\LWR@origbottomrule}[1]{\hline}
5453 \newcommand*\LWR@origaddlinespace}[1][1]{ }
5454 \newcommand*\LWR@origmorecmidrules[1]{ }
5455 \newcommand*\LWR@origspecialrule}[3]{\hline}
```

Env LWR@tabular [*(vertical position)*] {*(colspecs)*}

The new tabular environment will be \let in \LWR@LwarpStart, since siunitx might redefine tabular in the user's document.

```
5456 \StartDefiningTabulars
5457
5458 \newenvironment*LWR@tabular}[2][ ]
5459 {%
5460 \LWR@traceinfo{LWR@tabular started}%
5461 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
5462 \global\boolfalse{LWR@startedrow}%
```

Not yet doing an hline:

```
5463 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5464 \global\boolfalse{LWR@doingtbrule}%
```

For babel-french:

```
5465 \LWR@FBcancel%
```

Have not yet found the end of tabular command:

```
5466 \boolfalse{LWR@exitingtabular}%
```

Create the table tag:

```
5467 \global\booltrue{LWR@intabularmetadata}%
5468 \LWR@forcenewpage
5469 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5470 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
5471 \LWR@stoppars%
```

Track column # for setting text-align:

```
5472 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5473 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5474 \LetLtxMacro{\\}{\LWR@tabularendoffline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

```
5475 \LWR@traceinfo{LWR@tabular: redefining macros}%
5476 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5477
5478 \RenewDocumentCommand{\cline}{m}%
5479 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5480
5481 \DeclareDocumentCommand{\toprule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5482
5483 \DeclareDocumentCommand{\midrule}{o}{\LWR@domidrule\LWR@getmynexttoken}%
5484
5485 \DeclareDocumentCommand{\cmidrule}{o d() m}%
5486 {\LWR@docmidrule{##1}{##2}{##3}\LWR@getmynexttoken}%
5487
5488 \DeclareDocumentCommand{\bottomrule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5489 \DeclareDocumentCommand{\addlinespace}{o}{ }
5490 \DeclareDocumentCommand{\morecmidrules}{ }{ }
5491 \DeclareDocumentCommand{\specialrule}{m m m}{\LWR@domidrule\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`.

```
5492 \LetLtxMacro{\multicolumn}{\LWR@tabledatacolumnstag}%
5493 \LetLtxMacro{\multirow}{\LWR@tabledatamultirowtag}%
5494 \renewcommand*\mrowcell{\global\booltrue{\LWR@skippingmrowcell}}%
5495 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%
```

Reset for new processing:

```
5496 \global\boolfalse{\LWR@tableparcell}%
5497 \global\boolfalse{\LWR@skippingmrowcell}%
5498 \global\boolfalse{\LWR@skipatbang}%
```

Set `&` for its special meaning inside the tabular:

```
5499 \StartDefiningTabulars%
5500 \protected\gdef&{\LWR@tabularampersand}%
```

Look ahead for a possible table data cell:

```
5501 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
5502 \LWR@getmynexttoken%
5503 }%
```

Ending the environment:

```
5504 {%
5505 \LWR@traceinfo{\LWR@tabular ending}%
5506 \LWR@closetabledatacell%
5507 \LWR@htmlblocktag{/tr}%
5508 \LWR@htmlblocktag{/table}%
5509 \global\boolfalse{\LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
5510 \addtocounter{\LWR@tabulardepth}{-1}%
```

Restore `&` to its usual meaning:

```
5511 \protected\gdef&{\LWR@origampmacro}%
5512 \EndDefiningTabulars%
5513 \LWR@traceinfo{\LWR@tabular finished ending}%
5514 }
5515
5516 \EndDefiningTabulars

5517 \end{warpHTML}
```

59.21 Array

Pkg array

array is also automatically loaded by siunitx.

60 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 8 shows the data structures related to cross-referencing.

for HTML output: 5518 `\begin{warpHTML}`

60.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

5519 `\newcommand*{\@currentlabelname}{}`

`\LWR@stripperperiod` `{\text}` [`<.>`]

Removes a trailing period.

5520 `\def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%`

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

5521 `\newcommand*\LWR@setlatestname}[1]{%`

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

5522 `\GetTitleStringExpand{#1}%`

Table 8: Cross-referencing data structures

| | |
|---|------------------|
| Original \LaTeX: | (print and HTML) |
| <p><code>\refstepcounter</code>: Steps the counter and sets <code>\@currentlabel</code>.</p> <p><code>\@currentlabel</code>: <code>\p@<ctr>\the<ctr></code> Updated by <code>\refstepcounter</code>.</p> <p><code>\label</code>: Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code></p> <p><code>\newlabel</code>: When the .aux file is read, sets <code>\r@<label></code>.</p> <p><code>\r@<label></code>: Set to: <code>{\@currentlabel}{\thepage}}</code></p> <p><code>\ref</code>: Returns the first part of <code>\r@<label></code>.</p> <p><code>\pageref</code>: Returns the second part of <code>\r@<label></code>.</p> | |
| Added by lwarp: | (HTML only) |
| <p><code>\label</code>: Adds HTML tags (section 60.3), plus <code>\splabel</code> data (section 60.2):</p> <p style="padding-left: 2em;">zLWR@name: The section name for this label.</p> <p style="padding-left: 2em;">zLWR@htmlfilenumber: The file number or name for this label.</p> <p style="padding-left: 2em;">zLWR@lateximagedepth: The <code>lateximagedepth</code> for this label.</p> <p style="padding-left: 2em;">zLWR@lateximagenumber: The <code>lateximagenumber</code> for this label.</p> <p><code>\nameref</code>: Emulated from <code>hyperref</code> for <code>lwarp</code>. See section 60.4.</p> <p><code>\ref</code> and <code>\nameref</code>: Adds HTML tags. See section 60.4.</p> | |
| Added by amsmath: | (print and HTML) |
| <p><code>\label</code>: Execution is delayed until the math environment is completed.</p> <p><code>\ltx@label</code>: \LaTeX <code>\label</code>, (HTML: patched by <code>lwarp</code>), later patched by <code>cleveref</code>.</p> | |
| Added by cleveref: | (print and HTML) |
| <p><code>\refstepcounter</code>: Added: sets <code>\cref@currentlabel</code>.</p> <p><code>\cref@currentlabel</code>: (<code><type>=<ctr></code> unless an alias is used): <code>[<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>}</code> Also see section 46.4 for use with footnotes.</p> <p><code>\label</code>: Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code></p> <p><code>\newlabel</code>: (Unchanged.) When the .aux file is read, sets <code>\r@<label>@cref</code>.</p> <p><code>\r@<label>@cref</code>: Set to: <code>{\cref@currentlabel}{\thepage}}</code></p> <p>Utility functions: See <code>\cref@getlabel</code>, <code>\cref@gettype</code>, <code>\cref@getcounter</code>, <code>\cref@getprefix</code>.</p> <p>Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.</p> | |
| Additionally patched by lwarp: | (HTML only) |
| <p><code>\cref</code>, etc.: Modified for <code>lwarp</code>. See section 70.</p> <p><code>\label</code> inside math: See section 64.4.1.</p> | |
| Footnotes : See <code>\noteentry</code> in section 46.4. | |

```

5523 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5524 \edef\@currentlabelname{%
5525 \expandafter\LWR@stripperperiod\@currentlabelname%
5526 \ltx@empty.\ltx@empty\@nil%
5527 }%
5528 }

```

60.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called special:

```
5529 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5530 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```

5531 \zref@newprop{zLWR@htmlfilenumber}{%
5532 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}%
5533 }%

```

Additional properties for lateximages:

```

5534 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5535 \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:

```

5536 \zref@addprop{special}{zLWR@name}
5537 \zref@addprop{special}{zLWR@htmlfilenumber}
5538 \zref@addprop{special}{zLWR@lateximagedepth}
5539 \zref@addprop{special}{zLWR@lateximagenumber}

```

Returns the selected field:

```

5540 \newcommand*\LWR@spre{[2]}{%
5541 \zref@extractdefault{#1}{#2}{??}}

```

`\LWR@nameref` `{<label>}` Returns the section name for this label:

```
5542 \newcommand*\LWR@nameref}[1]{%
5543 \LWR@spref{#1}{zLWR@name}%
5544 }
```

`\LWR@htmlfileref` `{<label>}` Returns the file number for this label:

```
5545 \newcommand*\LWR@htmlfileref}[1]{%
5546 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5547 \LWR@spref{#1}{zLWR@htmlfilenumber}%
5548 }
```

`\LWR@lateximagedepthref` `{<label>}` Returns the lateximagedepth for this label:

```
5549 \newcommand*\LWR@lateximagedepthref}[1]{%
5550 \LWR@spref{#1}{zLWR@lateximagedepth}%
5551 }
```

`\LWR@lateximagenumberref` `{<label>}` Returns the lateximagenumber for this label:

```
5552 \newcommand*\LWR@lateximagenumberref}[1]{%
5553 \LWR@spref{#1}{zLWR@lateximagenumber}%
5554 }
```

`\LWR@splabel` `{<label>}` Sanitize the name and then creates the label:

```
5555 \newcommand*\LWR@splabel}[1]{%
5556 \LWR@setlatestname{\@currentlabelname}%
5557 \zref@labelbylist{#1}{special}}
```

60.3 Labels

`\LWR@subsublabel` `{<label>}` Creates an HTML id tag.

```
5558 \newcommand*\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5559 \ifthenelse{\cntttest{\value{LWR@lateximagedepth}}{>}{0}}{%
5560 {}}%
5561 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

5562 \ifbool{LWR@doingstartpars}%
5563   {% pars allowed
5564     \ifbool{LWR@doingapar}
5565     {% par started
5566       \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5567     }% par started
5568     {% par not started
5569       \LWR@stoppars%
5570       \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5571       \LWR@startpars%
5572     }% par not started
5573   }% pars allowed
5574   {% pars not allowed
5575     \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5576   }% pars not allowed
5577 }% not lateximage
5578 }

```

`\LWR@newlabel` $\langle label \rangle$ [$\langle type \rangle$]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular \LaTeX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```

5579 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5580 \LWR@traceinfo{LWR@newlabel: starting}%
5581 \LWR@traceinfo{LWR@newlabel: !#1!}%
5582 % \@bsphack%

```

Create a traditional \LaTeX label, as modified by `cleveref`:

```

5583 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

5584 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5585 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5586 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
5587 \LWR@splabel{#1}%
5588 \LWR@subsublabel{#1}%
5589 % \@esphack%
5590 \LWR@traceinfo{LWR@newlabel: done}%
5591 }

```

60.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
5592 \newcommand*{\LWR@startref}[1]
5593 {%
5594 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5595 \LWR@traceinfo{\LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
5596 \LWR@htmltag{a href="%
5597 \LWR@traceinfo{\LWR@startref B}%
5598 \LWR@htmlrefsectionfilename{#1}}%
5599 \LWR@traceinfo{\LWR@startref C}%
5600 \#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
5601 \LWR@traceinfo{\LWR@startref D: !#1!}%
5602 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```
5603 {%
5604   \LWR@traceinfo{\LWR@startref D0: ??}%
5605   ??%
5606 }%
```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```
5607 {%
5608   \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
5609   \ifthenelse{\cinttest{\LWR@lidref}{>}{0}}%
5610   {%
5611     \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
5612     lateximage\LWR@lateximagenumberref{#1}%
5613   }%
5614   {%
5615     \LWR@traceinfo{\LWR@startref D3}%
5616     #1%
5617   }%
```

```
5618 }%
5619 \LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
5620 "{}}%
5621 \LWR@traceinfo{LWR@startref F}%
5622 }
```

`\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)”.

```
5623 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5624 \LWR@traceinfo{LWR@subnewref #1 #2}%
5625 \LWR@startref{#1}%
5626 \LWR@origref{#2}%
5627 \LWR@htmltag{/a}%
5628 }
```

`\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.

```
5629 \NewDocumentCommand{\LWR@newref}{s m}{%
5630 \LWR@traceinfo{LWR@newref #2}%
5631 \IfBooleanTF{#1}%
5632 {\LWR@origref{#2}}%
5633 {\LWR@subnewref{#2}{#2}}%
5634 }
```

`\pagerefPageFor` Text for starred page references.

```
5635 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` * `{\label}` Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
5636 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5637 \IfBooleanTF{#1}%
5638 {\pagerefPageFor\LWR@origref{#2}}%
5639 {\cpageref{#2}}%
5640 }
```

`\nameref` $\langle label \rangle$

```
5641 \newcommand*\nameref[1]{%
5642 \LWR@traceinfo{nameref A}%
5643 \LWR@startref{#1}%
5644 \LWR@traceinfo{nameref B}%
5645 \LWR@nameref{#1}%
5646 \LWR@traceinfo{nameref C}%
5647 \LWR@htmltag{/a}%
5648 \LWR@traceinfo{nameref D}%
5649 }
```

`\Nameref` $\langle label \rangle$ In print, adds the page number. In HTML, does not.

```
5650 \let\Nameref\nameref
```

60.5 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.

```
5651 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5652 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref` $\langle URL \rangle$ $\langle text \rangle$

```
5653 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5654 \LWR@htmltag{a href="#1" target="\_{}blank"\LWR@originewline}#2\LWR@htmltag{/a}%
5655 \LWR@ensuredoingapar%
5656 }
```

`\LWR@subhyperrefclass` $\langle URL \rangle$ $\langle text \rangle$ $\langle htmlclass \rangle$

```

5657 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5658 \LWR@htmltag{a href="{#1}"
5659 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5660 \LWR@ensuredoingapar%
5661 }

```

`\href` [*options*] {*URL*} {*text*}

Create a link with accompanying text:

```

5662 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5663 \LWR@subhyperref{#2}{#3}%
5664 \endgroup%
5665 \LWR@ensuredoingapar%
5666 }
5667
5668 \newcommand{\href}{%
5669 \LWR@ensuredoingapar%
5670 \begingroup%
5671 \catcode'\_ =12
5672 \LWR@hrefb%
5673 }

```

`\nolinkurl` {*URL*}

Print the name of the link without creating the link:

```

5674 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5675
5676 \newcommand{\nolinkurl}{%
5677 \LWR@ensuredoingapar%
5678 \begingroup\catcode'\_ =12
5679 \LWR@nolinkurlb%
5680 }

```

`\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`.

```

5681 \newcommand*{\LWR@urlb}[1]{%
5682 \href{#1}{#1}%
5683 \endgroup%
5684 \LWR@ensuredoingapar%
5685 }
5686
5687 \newcommand{\LWR@urla}{%
5688 \LWR@ensuredoingapar%

```

```
5689 \begingroup\catcode'\_ =12
5690 \LWR@urlb%
5691 }
5692
5693 \let\url\LWR@urla
```

`\LWR@subinlineimage` [*⟨alntag⟩*] {*⟨class⟩*} {*⟨filename⟩*} {*⟨extension⟩*} {*⟨style⟩*}

```
5694 \newcommand*\LWR@subinlineimage[5] [] {%
5695 \ifthenelse{\equal{#1}{}}{%
5696 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
5697 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
5698 }

5699 \end{warpHTML}
```

Table 9: 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 \TeX file identifier for the output file.

`LWR@have<type>`: A boolean remembering 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.

61 Floats

Floats are supported, although partially through emulation.

Table 9 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”.

61.1 Float captions

for HTML output: 5700 `\begin{warpHTML}`

`\LWR@floatbegin` $\langle type \rangle$ [$\langle placement \rangle$]

Begins a `\newfloat` environment.

```
5701 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5702 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5703
5704 === #1 begin
5705
5706 }{}}%
5707 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5708 \addtocounter{\LWR@thisfloat}{1}%
5709 \booltrue{\LWR@freezethisfloat}%
```

```
5710 \begingroup
```

Settings while inside the environment:

```
5711 \LWR@origraggedright
```

Open an HTML figure tag:

```
5712 \LWR@htmltag{figure id="autofloat-\arabic{\LWR@thisfloat}" class="#1"}
```

```
5713 \renewcommand*{\@capttype}{#1}
5714 \caption@settype{#1}
5715 \LWR@startpars
5716 }
```

`\@float` Support packages which create floats directly.

`\@dblfloat`

```
5717 \let\@float\LWR@floatbegin
5718 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
5719 \newcommand*{\LWR@floatend}{%
5720 \LWR@stoppars%
5721 \LWR@htmlend{figure}}%
```

```

5722 \endgroup%
5723 \boolfalse{LWR@freezethisfloat}%
5724 \LWR@startpars%
5725 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5726
5727 === end
5728
5729 }{}%
5730 }

```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```

5731 \let\end@float\LWR@floatend
5732 \let\end@dblfloat\LWR@floatend

```

Ctr `LWR@thisfloat` 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.

```

5733 \newcounter{LWR@thisfloat}

```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5734 \newbool{LWR@freezethisfloat}
5735 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5736 \newcommand*{\LWR@maybeinthisfloat}{%
5737 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5738 }

```

`\@capttype` Remembers which float type is in use.

```

5739 \newcommand*{\@capttype}{}

```

61.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5740 \AtBeginDocument{\providecommand*\CaptionSeparator}{:-}}

```

`\@makecaption` `{<name and num>}{<text>}`

Prints the float type and number, the caption separator, and the caption text.

```

5741 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}

```

61.1.2 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 `autofloat` 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 `autofloat <id>` at the start of the float itself in the HTML file. The `autopage` and `autofloat` 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.

Ctrl `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctrl `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
5742 \newcounter{LWR@nextautofloat}
5743 \newcounter{LWR@nextautopage}
```

```
\LWRsetnextfloat {<autopage>} {<autofloat>}
```

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the HTML links.

```
5744 \newcommand*\LWRsetnextfloat}[2]{%
5745 \setcounter{LWR@nextautopage}{#1}%
5746 \setcounter{LWR@nextautofloat}{#2}%
5747 }
```

Ctrl `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autofloat` by the modified `\addcontentsline` just before each float's entry.

```
5748 \newcounter{LWR@latestautopage}
5749 \setcounter{LWR@latestautopage}{1}

5750 \let\LWR@origcaption@begin\caption@begin
5751 \let\LWR@origcaption@end\caption@end
5752 \let\LWR@orig@@par\@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5753 \newcommand*\LWR@caption@begin{
5754 {
5755 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
5756 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5757 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@par}%
```

No need for a minipage or \parbox inside the caption:

```
5758 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
5759 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5760 \LWR@htmlblocktag{figcaption}%
5761 \LWR@origcaption@begin%
5762 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5763 \newcommand{\LWR@caption@end}
5764 {%
5765 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5766 \LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
5767 \LWR@htmlblocktag{/figcaption}%
5768 \endgroup%
5769 % \leavevmode% avoid bad space factor (0) error
5770 \LWR@traceinfo{\LWR@caption@end: done}%
5771 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.
`\caption@end`

```
5772 \AtBeginDocument{
5773 \let\caption@begin\LWR@caption@begin
5774 \let\caption@end\LWR@caption@end
5775 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
5776 \let\LWR@origcaptionlistentry\captionlistentry
5777
5778 \renewcommand*{\captionlistentry}{%
```

```

5779 \LWR@maybeinthisfloat%
5780 \LWR@ensuredoingapar%
5781 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5782 \LWR@origcaptionlistentry%
5783 }
5784
5785 \def\LWR@LTcaptionlistentry{%
5786 \LWR@ensuredoingapar%
5787 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5788 \bgroup
5789 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
5790 {\egroup\LWR@LTcaptionlistentry}}%
5791 \def\LWR@LTcaptionlistentry#1{%
5792 \caption@listentry\@firstoftwo[LTcapter]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autofloat 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.

```

5793 \let\LWR@origaddcontentsline\addcontentsline
5794
5795 \renewcommand*{\addcontentsline}[3]{%
5796 \ifthenelse{\equal{#1}{toc}}{ }{%
5797   \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
5798   \addtocontents{\@nameuse{ext@#2}}{%
5799     \protect\LWRsetnextfloat%
5800     {\arabic{LWR@latestautopage}}%
5801     {\arabic{LWR@thisfloat}}%
5802     }% addtocontents
5803 }% not toc
5804 \LWR@origaddcontentsline{#1}{#2}{#3}%
5805 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5806 \AtBeginDocument{
5807 \let\LWR@origcaptionof\captionof
5808
5809 \renewcommand*{\captionof}{%
5810 \LWR@maybeinthisfloat%
5811 \LWR@stoppars
5812 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5813 \LWR@origcaptionof%
5814 }
5815 }

5816 \end{warpHTML}

```

62 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, 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 \TeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 5817 `\begin{warpHTML}`

62.1 Reading and printing the TOC

`\LWR@myshorttoc` `{\{toc/lof/lot\}}`

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 \TeX , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideroc.

```
5818 \newcommand*\LWR@myshorttoc}[1]{
5819 \LWR@ensuredoingapar
```

Only if the file exists:

```
5820 \IfFileExists{\jobname.#1}{
```



Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.



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`.

```
5821 \begingroup
5822 % \ifxetexorluatex%
5823 % \else
5824 % \inputencoding{latin1}% currently disabled
5825 % \fi
5826 \makeatletter
```

Read in the TOC file:

```
5827 \@input{\jobname.#1}
5828 % \makeatother
5829 \endgroup
5830 }%
5831 {}%
5832 }
```

```
\LWR@subtableofcontents {<toc/lof/lot>} {<sectionstarname>}
```

Places a TOC/LOF/LOT at the current position.

```
5833 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
5834 \@ifundefined{chapter}
5835 {\LWR@closeprevious{\LWR@depthsection}}
5836 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
5837 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
5838 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
5839 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
5840 \LWR@myshorttoc{#1}
```

Close the nav:

```
5841 \LWR@htmlelementclassend{nav}{#1}
5842 }
```

Patch \@starttoc to encapsulate the TOC inside HTML tags:

```
5843 \let\LWR@orig@starttoc\@starttoc
5844
5845 \renewcommand{\@starttoc}[1]{
```

```

5846 \LWR@htmlclass{nav}{#1}
5847 \LWR@orig@starttoc{#1}
5848 \LWR@htmlclassend{nav}{#1}
5849 }

```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

5850 \let\LWR@origtableofcontents\tableofcontents
5851 \let\LWR@origlistoffigures\listoffigures
5852 \let\LWR@origlistoftables\listoftables
5853
5854 \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:

```

5855 \ifbool{FormatWordProcessor}{}{

```

Copy the `.toc` file to `.sidedoc` for printing the sideroc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

5856   \LWR@copyfile{\jobname.toc}{\jobname.sidedoc}%
5857   \LWR@printpendingfootnotes
5858   \LWR@origtableofcontents
5859 }
5860 }
5861 \renewcommand*{\listoffigures}{
5862 \ifbool{FormatWordProcessor}{}{
5863   \LWR@printpendingfootnotes
5864   \LWR@origlistoffigures
5865 }
5866 }
5867
5868 \renewcommand*{\listoftables}{
5869 \ifbool{FormatWordProcessor}{}{
5870   \LWR@printpendingfootnotes
5871   \LWR@origlistoftables
5872 }
5873 }

```

62.2 High-level TOC commands

```
\listof {<type>} {<title>}
```

Emulate the `\listof` command from the float package (section 115). Used to create lists of custom float types. Also used to redefine the standard \TeX `\listoffigures` and `\listoftables` commands.

```
5874 \NewDocumentCommand{\listof}{m +m}{%
5875 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5876 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5877 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5878     \jobname.\csuse{ext@#1}\relax
5879 }
```

62.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sitetocname`, 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.

```
5880 \end{warpHTML}
```

for HTML & PRINT: 5881 `\begin{warpall}`

Ctrl **SideTOCDepth** Controls how deep the side-TOC gets. Use a standard \TeX section level similar to `tocdepth`.

```
5882 \newcounter{SideTOCDepth}
5883 \setcounter{SideTOCDepth}{1}
```

`\sitetocname` Holds the default name for the sidetoc.

```
5884 \newcommand{\sitetocname}{Contents}
```

```
5885 \end{warpall}
```

for HTML output: 5886 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

```
5887 \newcommand*\LWR@sidetoc}{
5888 \LWR@forcenewpage
5889 \LWR@stoppars
5890 }
```

The entire sidetoc is placed into a nav of class sidetoc.

```
5891 \LWR@htmlclass{nav}{sidetoc}
5892
5893 \setcounter{tocdepth}{\value{SideTOCDepth}}
5894 }
```

The title is placed into a div of class sidetoc title, and may contain paragraphs.

```
5895 \begin{BlockClass}{sidetoc title}
5896 \sidetocname
5897 \end{BlockClass}
```

The table of contents is placed into a div of class sidetoc contents.

```
5898 \begin{BlockClass}{sidetoc contents}
5899 \LinkHome
5900
5901 \LWR@myshorttoc{sidetoc}
5902 \end{BlockClass}
5903 \LWR@htmlclassend{nav}{sidetoc}
5904 }
```

62.4 Low-level TOC line formatting

`\numberline` $\{ \langle number \rangle \}$

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
5905 \renewcommand*\numberline[1]{%
5906 \LWR@sectionnumber{#1}\quad%
5907 }
```

`\hypertoc` $\{ \langle 1: depth \rangle \} \{ \langle 2: type \rangle \} \{ \langle 3: name \rangle \} \{ \langle 4: page \rangle \}$

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

#1 is depth

#2 is section, subsection, etc.

#3 the text of the caption

#4 page number

```
5908 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5909 \ifthenelse{\cinttest{#1}{<=} {\value{tocdepth}}}{%
5910   \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
5911   \LWR@subhyperrefclass{%
5912   \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
5913   \LWR@stoppars%
5914 }
5915 {}
5916 }
```

Ctrl `lofdepth` TOC depth for figures.

```
5917 \newcounter{lofdepth}
5918 \setcounter{lofdepth}{1}
```

Ctrl `lotdepth` TOC depth for tables.

```
5919 \newcounter{lotdepth}
5920 \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

```
5921 \newcommand{\hypertocfloat}[5]{%
5922 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5923 \@ifundefined{c@#3depth}{%
5924 \newcounter{#3depth}%
5925 \setcounter{#3depth}{1}%
5926 }{}%
```

Respond to `lofdepth`, etc.:

```
5927 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5928 \ifthenelse{\cnttest{#1}<=}{\arabic{#3depth}}{}%
5929 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-`(float number), with text of the caption, of the given HTML class.

```
5930 \LWR@subhyperrefclass{%
5931 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5932 \#autofloat-\arabic{LWR@nextautofloat}}%
5933 {#4}{toc#2}%
5934 \LWR@stoppars%
5935 }{}%
5936 }
```

Automatically called by `\contentsline`:

```
5937 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5938 \DeclareDocumentCommand{\l@chapter}{m m}
5939   {\hypertoc{0}{chapter}{#1}{#2}}
5940 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5941 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5942 \renewcommand{\l@subsubsection}[2]
5943   {\hypertoc{3}{subsubsection}{#1}{#2}}
5944 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5945 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5946 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5947 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5948 \end{warpHTML}
```

63 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: 5949 `\begin{warpHTML}`

```
5950 \newcounter{LWR@autoindex}
5951 \setcounter{LWR@autoindex}{0}
5952
5953 \newcounter{LWR@autoglossary}
5954 \setcounter{LWR@autoglossary}{0}
```

`\printindex`

```
5955 \let\LWR@origprintindex\printindex
5956
5957 \renewcommand*\printindex{
5958 {
5959 \LWR@startpars
5960 \LWR@origprintindex
5961 }
```

Env `theindex`

```
5962 \@ifundefined{chapter}
5963 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
5964 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
5965
5966 \renewenvironment*{theindex}{%
5967 \LWR@indexsection{\indexname}%
5968 \let\item\LWR@indexitem%
5969 \let\subitem\LWR@indexsubitem%
5970 \let\subsubitem\LWR@indexsubsubitem%
5971 }{}
```

`\LWR@indexitem`

```
5972 \newcommand*\LWR@indexitem>{
5973
```

```
5974 \InlineClass{indexitem}{  
5975 }
```

\LWR@indexitem

```
5976 \newcommand{\LWR@indexsubitem}{  
5977  
5978 \InlineClass{indexsubitem}{  
5979 }
```

\LWR@indexitem

```
5980 \newcommand{\LWR@indexsubsubitem}{  
5981  
5982 \InlineClass{indexsubsubitem}{  
5983 }
```

\@wrindex *{\langle term \rangle}* Redefined to write the LWR@latestautopage counter instead of page

```
5984 \def \@wrindex#1{%  
5985 \addtocounter{LWR@autoindex}{1}%  
5986 \LWR@newlabel{LWRindex-\theLWR@autoindex}%  
5987 \protected@write \@indexfile{%  
5988 {\string\indexentry{#1}{\theLWR@autoindex}}%  
5989 \endgroup  
5990 \@esphack}
```

\@wrglossary *{\langle term \rangle}* Redefined to write the LWR@latestautopage counter instead of page

```
5991 \def \@wrglossary#1{%  
5992 \addtocounter{LWR@autoglossary}{1}%  
5993 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%  
5994 \protected@write \@glossaryfile{%  
5995 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%  
5996 \endgroup  
5997 \@esphack}
```

\hyperindexref *{\langle autosecnumber \rangle}*

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

```
5998 \newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}  
  
5999 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case hyperref was not used:

```
6000 \begin{warpprint}
6001 \newcommand{\hyperindexref}[1]{#1}
6002 \end{warpprint}
```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
6003 \begin{warpall}
6004 \providecommand{\glo@name}{}
6005 \end{warpall}
```

64 Math

△ 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.

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

SVG math option For `svg math`, math is rendered as usual by \TeX into the initial PDF file using the current font¹², 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 \TeX 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 \TeX code which generated the math, allowing copy/paste of the \TeX 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 files As currently implemented, each instance of math creates a new SVG file. 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 checksumming may be used to remove the need for duplicate files.

SVG inline Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.

¹²See section 216 regarding fonts and fractions.

PNG files Others 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 also larger as well.

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 8 regarding EPUB output with MathJax.

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

Prog MathJax

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 Limitations when using MathJax include:

Prog MathJax

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 `Tikz` 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.

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

 **siunitx inside an equation**

LaTeX macros

- MathJax does not automatically support custom LaTeX macros, but they may be set up by the user.

custom MathJax macros

As an example of using custom LaTeX macros with MathJax, place the following at the start of the document, after `\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}
```

for HTML output: 6006 `\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.

```
6007 \let\LWR@origtextdollar\$
6008
6009 \renewcommand*{\$}{%
6010 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}{%
6011 {\LWR@origtextdollar}%
6012 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
6013 }
```

`ctr LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
6014 \newcounter{LWR@externalfilecnt}
```

64.1 Inline and display math

```
6015 \LetLtxMacro\LWR@origdollar$
6016 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

6017 \LetLtxMacro\LWR@origopenparen\(  
6018 \LetLtxMacro\LWR@origcloseparen\  
6019 \LetLtxMacro\LWR@origopenbracket\[
```

```
6020 \LetLtxMacro\LWR@origclosebracket\]
```

\$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:
\$\$

```
6021 \begingroup
6022 \catcode'\$=\active%
6023 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
6024 \gdef\LWR@doubledollar$#1$${
6025 \ifbool{mathjax}%
```

For MathJax, print the math between \lbrack and \rbrack :

```
6026 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
6027 {% not mathjax
6028
6029 \begin{lateximage}%
6030 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
6031 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6032 \end{lateximage}
6033
6034 }
6035 }%
```

\LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
6036 \gdef\LWR@subsingledollar#1{%
6037 \ifbool{mathjax}%
```

For MathJax, print the math between \lrcorner and \rcom :

```
6038 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
6039 {% not mathjax
6040   \begin{lateximage}%
6041   [\textbackslash(\LWR@HTMLsanitize{#1} \textbackslash)]%
6042   \LWR@origdollar#1\LWR@origdollar%
6043   \end{lateximage}%
6044 }%
6045 }
6046
```

```
6047 \gdef\LWR@singledollar#1${%
6048 \LWR@subsingledollar{#1}%
6049 }
```

\(Redefine to the above dollar macros.
 \)

```
6050 \gdef\(#1\){$#1$}
6051 \gdef\[#1\]{$$#1$$}
6052
6053 \endgroup
6054
6055
6056
```

\@ensuredmath $\{expression\}$ Not yet adapted to lwarp.

```
6057 \let\LWR@orig@ensuredmath\@ensuredmath
6058
6059 \newcommand*\LWR@ensuredmath}[1]{%
6060 \LWR@traceinfo{@ensuredmath: page \thepage: \detokenize{#1}}%
6061 \LWR@subsingledollar{#1}%
6062 }
```

Remove the old math and displaymath environments:

```
6063 \let\math\relax
6064 \let\endmath\relax
6065 \let\displaymath\relax
6066 \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`.

```
6067 \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`.

```
6068 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

64.2 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```
6069 \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 .

```
6070 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6071 \ifcsdef{thechapter}{
6072 \InlineClass{hidden}{
6073 \textbackslash(
6074 \textbackslash{}seteqsection \{\thechapter\}
6075 \textbackslash)
6076 }
6077 }
6078 }{}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6079 \ifthenelse{\cnttest{\value{equation}}>0}
6080 {
```

Tell MathJax that the next set of equations begins with the current \LaTeX equation number, plus one.

```
6081 \setcounter{LWR@nextequation}{\value{equation}}
6082 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\ (“ and “\)” characters, to be printed to HTML, not interpreted by \LaTeX .

```
6083 \InlineClass{hidden}{
6084 \textbackslash(
6085 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
6086 \textbackslash)
6087 }
6088 }{}% not eq > 0
6089 }
```

`\LWR@restoreorigrelsize` Patched by lwarp-relsize if loaded.

```
6090 \newcommand*{\LWR@restoreorigrelsize}{}
```

`\LWR@restoreorigformatting` While producing SVG math or a lateximage, use regular \LaTeX formatting instead of HTML tags.

```

6091 \newcommand*{\LWR@restoreorigformatting}{%
6092 \LWR@traceinfo{\LWR@restoreorigformatting}%
6093 \LetLtxMacro{\hspace}{\LWR@origspace}%
6094 \LetLtxMacro\rule{\LWR@origrule}%
6095 \let\,\LWR@origcomma% disable HTML short unbreakable space
6096 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6097 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6098 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6099 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6100 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
6101 \LetLtxMacro{\textit}{\LWR@origtextit}%
6102 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
6103 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6104 \LetLtxMacro{\textup}{\LWR@origtextup}%
6105 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6106 \LetLtxMacro{\emph}{\LWR@origemph}%
6107 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6108 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6109 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6110 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6111 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6112 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6113 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6114 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6115 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6116 \LetLtxMacro{\em}{\LWR@origem}%
6117 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6118 \let\sp\LWR@origsp%
6119 \let\sb\LWR@origsb%
6120 \let\textsuperscript\LWR@origtextsuperscript%
6121 \let\textsubscript\LWR@origtextsubscript%
6122 \let~\LWR@origtilde%
6123 \let\enskip\LWR@origenskip%
6124 \let\quad\LWR@origquad%
6125 \let\qquad\LWR@origqquad%
6126 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6127 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6128 \LetLtxMacro\toprule{\LWR@origtoprule}%
6129 \LetLtxMacro\midrule{\LWR@origmidrule}%
6130 \LetLtxMacro\cmidrule{\LWR@origcmidrule}%
6131 \LetLtxMacro\bottomrule{\LWR@origbottomrule}%
6132 \LetLtxMacro\addlinespace{\LWR@origaddlinespace}%
6133 \LetLtxMacro\morecmidrules{\LWR@origmorecmidrules}%
6134 \LetLtxMacro\specialrule{\LWR@origspecialrule}%
6135 \let\newline\LWR@orignewline%
6136 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6137 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%

```

```

6138 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6139 \let\reflectbox\LWR@origreflectbox%
6140 \let\framebox\LWR@origframebox%
6141 \let\makebox\LWR@origmakebox%
6142 \let\fbbox\LWRprint@fbbox%
6143 \let\fbboxBlock\LWRprint@fbbox%
6144 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6145 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6146 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6147 \let\endminipage\LWR@endminipage%
6148 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6149 \LWR@restoreorigxcolor%
6150 \LWR@restoremoreorigxcolor%
6151 %
6152 \LWR@restoreorigtextcomp%
6153 \LWR@restoreorigaccents%
6154 \LWR@restoreorigrelsize%
6155 %
6156 \LWR@FBcancel%
6157 }

```

`\LWR@hidelatexequation` $\{ \langle environment \rangle \} \{ \langle contents \rangle \}$

Creates the \LaTeX version of the equation inside an HTML comment.

```
6158 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```

6159 \LWR@stoppars
6160 \LWR@htmlopencomment
6161

```

Start the \LaTeX math environment inside the HTML comment:

```

6162 \begingroup
6163 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their \LaTeX meanings.

```
6164 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section [64.4.1](#).

Print the contents of the equation:

```
6165 #2
```

End the \LaTeX math environment inside the HTML comment:

```
6166 \csuse{LWR@origend#1}
6167 \endgroup
6168
```

Close the HTML comment and resume HTML paragraph handling:

```
6169 \LWR@htmlclosecomment
6170 \LWR@startpars
6171 }
```

```
\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 .

```
6172 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

```
6173 \LWR@origtilde\LWR@orignewline
6174 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
6175 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
6176 \textbackslash{}end\{#1\}
6177 \LWR@orignewline
6178 }
```

64.3 Equation environment

Remember existing equation environment:

```
6179 \let\LWR@origequation\equation
6180 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
6181 \let\equation\relax
6182 \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 \TeX 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 `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
6183 \NewEnviron{equation}{%
6184
6185 \ifbool{mathjax}
```

MathJax output:

```
6186 {
```

Print commands to synchronize MathJax's equation number and format to the current \TeX chapter/section and equation number:

```
6187 \LWR@syncmathjax
```

Print the \TeX math inside an HTML comment:

```
6188 \LWR@hidelatexequation{equation}{\BODY}
6189 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the \TeX equation environment commands, and the contents of the environment's `\BODY`.

```
6190 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
6191 \begin{lateximage}[{\theequation} \textbackslash{begin\{equation\}}}%
6192 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
6193 \textbackslash{end\{equation\}}]% alt tag
```

Create the actual \TeX -formatted equation inside the `lateximage` using the contents of the environment.

```
6194 \LWR@origequation
6195 \BODY% contents collected by NewEnviron
6196 \LWR@origendequation
6197 \end{lateximage}%
```

```
6198 }
6199
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6200 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

64.4 AMS Math environments

64.4.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for multiline-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a multiline environment and not an equation.

```
6201 \newbool{LWR@amsmultiline}
6202 \boolfalse{LWR@amsmultiline}
```

`\LWR@htmlmathlabel` $\langle label \rangle$

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \TeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \TeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
6203 \newcommand*{\LWR@htmlmathlabel}[1]{%
6204 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
6205 \ifbool{mathjax}{%
```

The combined \TeX & HTML label is printed in a `\text` field:

```
6206 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6207 \ifbool{LWR@amsmultiline}{\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.)

```
6208 \LWR@htmlclosecomment%
6209 \LWR@origltx@label{#1}%
6210 \LWR@htmlopencomment%
6211 }% text
6212 }% mathjax
6213 {%
6214 \LWR@origltx@label{#1}%
6215 }%
6216 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
6217 \newcommand*{\LWR@beginhideamsmath}{
6218 \LWR@stoppars
6219 \LWR@origtilde\LWR@orignewline
6220 \LWR@htmlopencomment
6221
6222 \begingroup
6223 \LWR@restoreorigformatting
6224 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```
6225 \newcommand*{\LWR@endhideamsmath}{
6226 \endgroup
6227
6228 \LWR@htmlclosecomment
6229 \LWR@orignewline
6230 \LWR@startpars
6231 }
```

64.4.2 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 environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument 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 environment 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`.

```
6232 \let\LWR@origeqnarray\eqnarray
6233 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6234 \newbool{LWR@numbereqnarray}
6235 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6236 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on MathJax vs SVG math:

```
6237 \ifbool{mathjax}
6238 {
```

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:

```
6239   \LWR@syncmathjax
6240   \boolfalse{LWR@amsmultline}
6241   \ifbool{LWR@numbereqnarray}
6242   {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
6243     \LWR@beginhideamsmath
6244     \LWR@origeqnarray
6245     \BODY
6246     \LWR@origendeqnarray
6247     \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6248     \LWR@addmathjax{eqnarray}{\BODY}
6249   }%
6250   {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6251 \LWR@addmathjax{eqnarray*}{\BODY}
6252 }% LWR@numbreqnarray
6253 }% mathjax
6254 {% not mathjax
6255 \ifbool{LWR@numbreqnarray}
6256 {
```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```
6257 \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
```

Then create the image contents using an actual eqnarray:

```
6258 \LWR@origeqnarray
6259 \BODY
6260 \LWR@origendeqnarray
6261 \end{lateximage}
6262 }%
6263 {% not LWR@numbreqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```
6264 \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6265 \LWR@origeqnarray
6266 \BODY
6267 \nonumber
6268 \LWR@origendeqnarray
6269 \end{lateximage}
6270 }% LWR@numbreqnarray
6271 }% not mathjax
```

Default to number equations in the future:

```
6272 \booltrue{LWR@numbreqnarray}
6273 }
```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```
6274 \RenewEnviron{eqnarray}
6275 {%
6276
6277 \LWR@eqnarrayfactor
6278
6279 }
```

The starred version is patched to turn off the numbering:

```
6280 \csgpreto{eqnarray*}{\boolfalse{LWR@numbreqnarray}}
```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env `multline`

```
6281 \BeforeBeginEnvironment{multline}{
6282
6283 \ifbool{mathjax}
6284 {
6285     \LWR@syncmathjax
6286     \booltrue{LWR@amsmultline}
6287     \LWR@beginhideamsmath
6288 }
6289 {
6290     \lateximage[LWRAMSMATHBODY]
6291 }
6292 }
6293
6294 \AfterEndEnvironment{multline}{
6295
6296 \ifbool{mathjax}
6297 {
6298     \LWR@endhideamsmath
6299     \boolfalse{LWR@amsmultline}
6300     \LWR@addmathjax{multline}{\the\@envbody}
6301 }
6302 {\endlateximage}
6303
6304 }
```

Env `multline*`

```
6305 \BeforeBeginEnvironment{multline*}{
6306
6307 \ifbool{mathjax}
6308 {
6309     \LWR@syncmathjax
6310     \booltrue{LWR@amsmultline}
6311     \LWR@beginhideamsmath
6312 }
6313 {
6314     \lateximage[LWRAMSMATHBODY]
6315 }
6316 }
6317
```

```
6318 \AfterEndEnvironment{multline*}{
6319
6320 \ifbool{mathjax}
6321 {
6322   \LWR@endhideamsmath
6323   \boolfalse{LWR@amsmultline}
6324   \LWR@addmathjax{multline*}{\the\@envbody}
6325 }
6326 {\endlateximage}
6327
6328 }
6329
```

Env **gather**

```
6330 \BeforeBeginEnvironment{gather}{
6331
6332 \ifbool{mathjax}
6333 {
6334   \LWR@syncmathjax
6335   \boolfalse{LWR@amsmultline}
6336   \LWR@beginhideamsmath
6337 }
6338 {
6339   \lateximage[LWRAMSMATHBODY]
6340 }
6341 }
6342
6343 \AfterEndEnvironment{gather}{
6344
6345 \ifbool{mathjax}
6346 {
6347   \LWR@endhideamsmath
6348   \LWR@addmathjax{gather}{\the\@envbody}
6349 }
6350 {\endlateximage}
6351
6352 }
```

Env **gather***

```
6353 \BeforeBeginEnvironment{gather*}{
6354
6355 \ifbool{mathjax}
6356 {
6357   \LWR@syncmathjax
6358   \boolfalse{LWR@amsmultline}
6359   \LWR@beginhideamsmath
```

```
6360 }
6361 {
6362   \lateximage[LWRAMSMATHBODY]
6363 }
6364 }
6365
6366 \AfterEndEnvironment{gather*}{
6367
6368 \ifbool{mathjax}
6369 {
6370   \LWR@endhideamsmath
6371   \LWR@addmathjax{gather*}{\the\@envbody}
6372 }
6373 {\endlateximage}
6374
6375 }
```

Env `align`

```
6376 \BeforeBeginEnvironment{align}{
6377
6378 \ifbool{mathjax}
6379 {
6380   \LWR@syncmathjax
6381   \boolfalse[LWR@amsmultline]
6382   \LWR@beginhideamsmath
6383 }
6384 {
6385   \lateximage[LWRAMSMATHBODY]
6386 }
6387 }
6388
6389 \AfterEndEnvironment{align}{
6390
6391 \ifbool{mathjax}
6392 {
6393   \LWR@endhideamsmath
6394   \LWR@addmathjax{align}{\the\@envbody}
6395 }
6396 {\endlateximage}
6397
6398 }
```

Env `align*`

```
6399 \BeforeBeginEnvironment{align*}{
6400
6401 \ifbool{mathjax}
```

```
6402 {
6403   \LWR@syncmathjax
6404   \boolfalse{LWR@amsmultline}
6405   \LWR@beginhideamsmath
6406 }
6407 {
6408   \lateximage[LWRAMSMATHBODY]
6409 }
6410 }
6411
6412 \AfterEndEnvironment{align*}{
6413
6414 \ifbool{mathjax}
6415 {
6416   \LWR@endhideamsmath
6417   \LWR@addmathjax{align*}{\the\@envbody}
6418 }
6419 {\endlateximage}
6420
6421 }
```

Env `flalign`

```
6422 \BeforeBeginEnvironment{flalign}{
6423
6424 \ifbool{mathjax}
6425 {
6426   \LWR@syncmathjax
6427   \boolfalse{LWR@amsmultline}
6428   \LWR@beginhideamsmath
6429 }
6430 {
6431   \lateximage[LWRAMSMATHBODY]
6432 }
6433 }
6434
6435 \AfterEndEnvironment{flalign}{
6436
6437 \ifbool{mathjax}
6438 {
6439   \LWR@endhideamsmath
6440   \LWR@addmathjax{flalign}{\the\@envbody}
6441 }
6442 {\endlateximage}
6443
6444 }
```

Env `flalign*`

```

6445 \BeforeBeginEnvironment{flalign*}{
6446
6447 \ifbool{mathjax}
6448 {
6449   \LWR@syncmathjax
6450   \boolfalse{LWR@amsmultline}
6451   \LWR@beginhideamsmath
6452 }
6453 {
6454   \lateximage[LWRAMSMATHBODY]
6455 }
6456 }
6457
6458 \AfterEndEnvironment{flalign*}{
6459
6460 \ifbool{mathjax}
6461 {
6462   \LWR@endhideamsmath
6463   \LWR@addmathjax{flalign*}{\the\envbody}
6464 }
6465 {\endlateximage}
6466
6467 }

6468 \end{warpHTML}

```

65 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding 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 \TeX 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}`

for HTML output: 6469 `\begin{warpHTML}`

Ctrl LWR@lateximagenumber Sequence the images.

```
6470 \newcounter{LWR@lateximagenumber}
6471 \setcounter{LWR@lateximagenumber}{0}
```

Ctrl LWR@lateximagedepth Do not create `\lateximage` inside of `\lateximage`.

```
6472 \newcounter{LWR@lateximagedepth}
6473 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
6474 \edef\LWR@hashmark{\string#} % for use in \write
6475 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl LWR@LIpage Used to reference the PDF page number of a `lateximage` to be written into `lateximages.txt`.

```
6476 \newcounter{LWR@LIpage}
6477 \end{warpHTML}
```

for HTML & PRINT: 6478 `\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.

Do not include the leading backslash in the name.

```
6479 \newcommand*{\LateximageFontSizeName}{large}
6480 \end{warpall}
```

for HTML output: 6481 `\begin{warpHTML}`

`\LWR@HTMLsanitize` `{\langle text \rangle}`

Math expressions are converted to `lateximages`, 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...

```

6482 \newcommand{\LWR@HTMLsanitize}[1]{%
6483 \begingroup%
6484 \LWR@FBcancel%
6485 \protect\StrSubstitute{\detokenize{#1}}%
6486 {\detokenize{&}}%
6487 {\detokenize{&}}[\LWR@strresult]%
6488 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6489 {\detokenize{<}}%
6490 {\detokenize{&lt;}}%
6491 [\LWR@strresult]%
6492 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6493 {\detokenize{>}}%
6494 {\detokenize{&gt;}}%
6495 [\LWR@strresult]%
6496 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6497 {\detokenize{##}}%
6498 {\#}%
6499 [\LWR@strresult]%
6500 \LWR@strresult%
6501 \endgroup%
6502 }

```

`\LWR@HTMLsanitizeexpand` $\{ \langle \textit>text \rangle \}$

This version expands the argument before sanitizing it.

```

6503 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6504 \begingroup%
6505 \LWR@FBcancel%
6506 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
6507 {\detokenize{&}}%
6508 {\detokenize{&}}[\LWR@strresult]%
6509 [\LWR@strresult]%
6510 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6511 {\detokenize{<}}%
6512 {\detokenize{&lt;}}%
6513 [\LWR@strresult]%
6514 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6515 {\detokenize{>}}%
6516 {\detokenize{&gt;}}%
6517 [\LWR@strresult]%
6518 \LWR@strresult%
6519 \endgroup%
6520 }

```

Env `lateximage` $[\langle \textit>alttag \rangle]$

```

6521 \catcode'\$=\active%

```

```

6522
6523 \NewDocumentEnvironment{lateximage}{0{image}}
6524 {%
6525 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
6526 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

6527 {%
6528   \addtocounter{LWR@lateximagedepth}{1}%
6529 }%

```

Otherwise, this is the outer-most lateximage:

```

6530 {% start of outer-most lateximage

```

Starting a new lateximage:

```

6531   \addtocounter{LWR@lateximagenumber}{1}%
6532   \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a lateximage, locally do not use mathjax:

```

6533   \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

6534   \LWR@ensuredoingapar%

```

Next file:

```

6535   \addtocounter{LWR@externalfilecnt}{1}%
6536   \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

6537   \setcounterpageref{LWR@LIpage}{LWRlateximage\theLWR@lateximagenumber}%
6538   \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:

```

6539   \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
6540   class="lateximagesource"{}} \LWR@orignewline

```

Write instructions to the lateximages.txt file:

```
6541 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
6542 \immediate\write\LWR@lateximagesfile{|\theLWR@LPage|\theLWR@externalfilecnt|}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```
6543 \LWR@traceinfo{lateximage: about to create open comment}%
6544 \LWR@htmlopencomment%
```

One level deeper:

```
6545 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
6546 \LWR@traceinfo{lateximage: about to create new page}%
6547 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6548 \LWR@traceinfo{lateximage: about to create minipage}%
6549 \LWR@origminipage{6in}%
6550 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a lateximage.

```
6551 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6552 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
6553 \def\@mpfn{footnote}%
6554 \def\thempfn{\thefootnote}%
6555 \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
6556 \LWR@traceinfo{lateximage: about to create label}%
6557 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
6558 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
6559 \LetLtxMacro$\LWR@origdollar%
6560 \catcode'\$=3% math shift
```

```

6561 \LetLtxMacro\(\LWR@origopenparen%
6562 \LetLtxMacro\)\LWR@origcloseparen%
6563 \LetLtxMacro\[\LWR@origopenbracket%
6564 \LetLtxMacro\]\LWR@origclosebracket%
6565 \let\@ensuredmath\LWR@orig@ensuredmath%
6566 }% end of outer-most lateximage
6567 \LWR@traceinfo{lateximage: finished start of environment}%
6568 }% end of \begin{lateximage}

```

When the environment closes:

```

6569 {% start of \end{lateximage}
6570 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

6571 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

6572 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

6573 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

6574 \LWR@origendminipage%
6575 \LWR@orignewpage%
6576 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

6577 \LWR@htmlclosecomment{\LWR@orignewline%
6578 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}}%

```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```

6579 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
6580 {%
6581 \LWR@subinlineimage[%
6582 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6583 ]%

```

```

6584     {lateximage}%
6585     {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}%
6586     {svg}%
6587     {}%
6588     }%
6589     {%
6590     \LWR@subinlineimage[#1]{lateximage}%
6591     {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}{svg}{}%
6592     }%
6593 % \LWR@orignewline% Removed to prevent extra space.

```

Be sure that are doing a paragraph:

```
6594     \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```

6595     \LWR@htmltag{/span}%
6596     \ifbool{HTMLDebugComments}{%
6597         \LWR@htmlcomment{End of lateximage}%
6598     }{}%
6599 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

6600     \addtocounter{LWR@lateximagedepth}{-1}%
6601 }% end of outer-most lateximage
6602 \LWR@traceinfo{lateximage: done}%
6603 }%
6604 \catcode'\$=3% math shift
6605 \end{warpHTML}

```

```

for PRINT output: 6606 \begin{warpprint}
6607 % \newenvironment{lateximage}[1] [] {\minipage{\linewidth}}{\endminipage}
6608 \newenvironment{lateximage}[1] [] {}{}
6609 \end{warpprint}

```

66 center, flushleft, flushright

```
for HTML output: 6610 \begin{warpHTML}
```

Env center Replace center functionality with css tags:

```
6611 \renewenvironment*{center}
```

```

6612 {
6613 \LWR@forcenewpage
6614 \BlockClass{center}
6615 }
6616 {\endBlockClass}

```

Env **flushright**

```

6617 \renewenvironment*{flushright}
6618 {
6619 \LWR@forcenewpage
6620 \BlockClass{flushright}
6621 }
6622 {\endBlockClass}

```

Env **flushleft**

```

6623 \renewenvironment*{flushleft}
6624 {
6625 \LWR@forcenewpage
6626 \BlockClass{flushleft}
6627 }
6628 {\endBlockClass}

6629 \end{warpHTML}

```

67 Siunitx

Pkg **siunitx**



per-mode

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

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

for HTML output: 6630 `\begin{warpHTML}`

Options for `siunitx`:

```

6631 \PassOptionsToPackage{
6632   detect-mode=true,
6633   per-mode=symbol,% fraction is not seen by pdftotext
6634 %   text-celsius = {\protect\LWRsiunitx@degree}{C},
6635 %   text-degree = {\protect\LWRsiunitx@degree},
6636 }{siunitx}
6637

```

6638 \end{warpHTML}

68 Graphics

Pkg `graphics`

Pkg `graphicx`

⚠ **graphics vs. graphicx** 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.

⚠ **viewports**

⚠ **\graphicspath** `\graphicspath` only works for a single directory; all graphics must be in this directory.

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`. Using 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.

⚠ **image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for HTML output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.

\rotatebox `\rotatebox` accepts the optional `origin` key.

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

for HTML output: 6639 \begin{warpHTML}

68.1 `\graphicspath`

`\graphicspath` $\{\langle path \rangle\}$

```
6640 \newcommand*\thisgraphicspath{{}}
6641 \renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{#1}}
```

`\DeclareGraphicsExtensions` $\{\langle list \rangle\}$

`\DeclareGraphicsRule` $\{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\}$

```
6642 \renewcommand*\DeclareGraphicsExtensions[1]{}
6643 \renewcommand*\DeclareGraphicsRule[4]{}

```

68.2 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.

```
6644 \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.

```
6645 \newlength{\LWR@igwidth}
6646 \newlength{\LWR@igheight}
6647 \newcommand*\LWR@igwidthstyle{}
6648 \newcommand*\LWR@igheightstyle{}
6649 \newcommand*\LWR@igorigin{}
6650 \newcommand*\LWR@igangle{}
6651 \newcommand*\LWR@igxscale{1}
6652 \newcommand*\LWR@igyscale{1}
6653 \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:

```
6654 \define@key{igraph}{width}{%
6655 \setlength{\LWR@igwidth}{#1}%
6656 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
6657 {%
```

Default to use the converted fixed length given:

```
6658 \uselengthunit{PT}%
6659 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
6660 \IfEndWith{#1}{ex}%
6661 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
6662 {}% not ex
6663 \IfEndWith{#1}{em}%
6664 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6665 {}% not em
6666 \IfEndWith{#1}{\}%
6667 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6668 {}% not percent
6669 \IfEndWith{#1}{px}%
6670 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6671 {}% not px
6672 }{}% end of length > Opt
6673 }
```

If an optional height was given, set an HTML style:

```
6674 \define@key{igraph}{height}{%
6675 \setlength{\LWR@igheight}{#1}%
6676 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
6677 {%
```

Default to use the converted fixed length given:

```
6678 \uselengthunit{PT}%
6679 \renewcommand*{\LWR@igheightstyle}{%
6680 height:\rndprintlength{\LWR@igheight} %
6681 }%
```

If ex or em dimensions were given, use those instead:

```
6682 \IfEndWith{#1}{ex}%
6683 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6684 {}% not ex
6685 \IfEndWith{#1}{em}%
6686 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6687 {}% not em
6688 \IfEndWith{#1}{\}%
6689 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6690 {}% not percent
6691 \IfEndWith{#1}{px}%
6692 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
```

```

6693   {}% not px
6694 }{}% end of length > Opt
6695 }

```

Handle origin key:

```

6696 \define@key{igraph}{origin}{%
6697 \renewcommand*\LWR@igorigin}{#1}%
6698 }

```

Handle angle key:

```

6699 \define@key{igraph}{angle}{\renewcommand*\LWR@igangle}{#1}}

```

Handle class key:

```

6700 \define@key{igraph}{class}{\renewcommand*\LWR@igclass}{#1}}
6701

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

6702 \define@key{igraph}{scale}{%
6703 \renewcommand*\LWR@igxscale}{#1}%
6704 \renewcommand*\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

6705 \define@key{igraph}{bb}{}
6706 \define@key{igraph}{bbllx}{}
6707 \define@key{igraph}{bbllx}{}
6708 \define@key{igraph}{bburx}{}
6709 \define@key{igraph}{bbury}{}
6710 \define@key{igraph}{natwidth}{}
6711 \define@key{igraph}{natheight}{}
6712 \define@key{igraph}{hiresbb}{}
6713 \define@key{igraph}{viewport}{}
6714 \define@key{igraph}{trim}{}
6715 \define@key{igraph}{totalheight}{}
6716 \define@key{igraph}{keepaspectratio}{}
6717 \define@key{igraph}{clip}{}
6718 \define@key{igraph}{draft}{}
6719 \define@key{igraph}{type}{}
6720 \define@key{igraph}{ext}{}
6721 \define@key{igraph}{read}{}
6722 \define@key{igraph}{command}{}

```

```
\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.

```
6723 \newcommand*\LWR@rotstyle}[2]{%
6724 #1transform:rotate(-#2deg);
6725 }
```

`\LWR@scalestyle` `{\langle prefix \rangle}{\langle xscale \rangle}{\langle yscale \rangle}`

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.

```
6726 \newcommand*\LWR@scalestyle}[3]{%
6727 #1transform:scale(#2,#3);
6728 }
```

68.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
6729 \newbool{LWR@infloatrow}
6730 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
6731 \def\LWR@opacity{1}
```

```
6732 \newcommand*\LWR@imageextension{}
6733 \newcommand*\LWR@expgraphicsfilename{}
```

`\LWR@includegraphicsb` * `[\langle 2: options \rangle] [\langle 3: options \rangle] {\langle 4: filename \rangle}`

graphics syntax is `\includegraphics * [\langle llx, lly \rangle] [\langle urx, ury \rangle] {\langle file \rangle}`

graphicx syntax is `\includegraphics [\langle key values \rangle] {\langle file \rangle}`

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
6734 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
6735 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6736 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6737 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
6738 {%
6739   \ifbool{LWR@infloatrow}%
6740   {}
6741   {% not in a minipage or a floatrow:
6742     \setlength{\linewidth}{6in}%
6743     \setlength{\textwidth}{6in}%
6744     \setlength{\textheight}{9in}%
6745   }%
6746 }{}%
```

See if can find the image by adding an extension:

Preference is `svgz`, then `svg`, `gif`, `png`, and `jpg`.

`\detokenize\expandafter` allows underscore characters in filenames.

```
6747 \edef\LWR@expgraphicsfilename{#4}
6748 \renewcommand*\LWR@imageextension{}%
6749 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6750 {\renewcommand*\LWR@imageextension}{.jpg}}{}%
6751 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6752 {\renewcommand*\LWR@imageextension}{.JPG}}{}%
6753 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
6754 {\renewcommand*\LWR@imageextension}{.png}}{}%
6755 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6756 {\renewcommand*\LWR@imageextension}{.PNG}}{}%
6757 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
6758 {\renewcommand*\LWR@imageextension}{.gif}}{}%
6759 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6760 {\renewcommand*\LWR@imageextension}{.GIF}}{}%
6761 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
6762 {\renewcommand*\LWR@imageextension}{.svg}}{}%
6763 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6764 {\renewcommand*\LWR@imageextension}{.SVG}}{}%
6765 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6766 {\renewcommand*\LWR@imageextension}{.svgz}}{}%
6767 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6768 {\renewcommand*\LWR@imageextension}{.SVGZ}}{}%
```

Convert a PDF extension to `svg`, leave the result in `\LWR@strresult`:

Must also `\detokenize .pdf` and `.svg` comparison strings.

```
6769 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6770 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6771 %
6772 \StrSubstitute{\LWR@strresult}%
6773 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%
```

For correct em sizing during the width and height conversions:

```
6774 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
6775 \setlength{\LWR@igwidth}{0pt}%
6776 \setlength{\LWR@igheight}{0pt}%
6777 \renewcommand*{\LWR@igwidthstyle}{}%
6778 \renewcommand*{\LWR@igheightstyle}{}%
6779 \renewcommand*{\LWR@igorigin}{}%
6780 \renewcommand*{\LWR@igangle}{}%
6781 \renewcommand*{\LWR@igxscale}{1}%
6782 \renewcommand*{\LWR@igyyscale}{1}%
6783 \renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```
6784 \IfValueTF{#3}{}{}%
6785 \IfValueTF{#2}{}{}%
6786 {\setkeys{igraph}{#2}}{}%
6787 {\setkeys{igraph}{}{}%
6788 }%
```

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.

```
6789 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
6790 {% start of href
6791 \LWR@htmltag{% start of image tags
6792 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6793 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline
```

Only include a style tag if a width, height, angle, or scale was given:

```
6794 \ifthenelse{
6795   \NOT\equal{\LWR@igwidthstyle}{} \OR
6796   \NOT\equal{\LWR@igheightstyle}{} \OR
```

```

6797 \NOT\equal{\LWR@igorigin}{ } \OR
6798 \NOT\equal{\LWR@igangle}{ } \OR
6799 \NOT\equal{\LWR@igxscale}{1} \OR
6800 \NOT\equal{\LWR@igyscale}{1}
6801 }%
6802 {\LWR@origtilde{ } style="%
6803 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
6804 {\LWR@igwidthstyle; }}%
6805 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
6806 {\LWR@igheightstyle; }}%
6807 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
6808 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{ }%
6809 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
6810 { %
6811 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6812 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6813 \LWR@rotstyle}{\LWR@igangle%
6814 } }%
6815 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
6816 \NOT\equal{\LWR@igyscale}{1}}%
6817 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
6818 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6819 \LWR@scalestyle}{\LWR@igxscale}{\LWR@igyscale}}}%
6820 %
6821 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
6822 {opacity:\LWR@opacity; }%
6823 { }%
6824 %
6825 " \LWR@orignewline}{ }%

```

Set the class:

```

6826 \LWR@origtilde{ } class="\LWR@igclass" \LWR@orignewline%
6827 }% end of image tags
6828 }% end of href
6829 \endgroup

```

Return to small-sized output:

```

6830 \LWR@origscriptsize
6831 }

```

`\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.

```
6832 \renewcommand*\includegraphics{
6833 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
6834 \LWR@ensuredoingapar%
6835 \begingroup%
6836 \LWR@includegraphicsb%
6837 }
```

```
6838 \end{warpHTML}
```

for PRINT output: For print output, accept and then discard the new `class` key:

```
6839 \begin{warpprint}
6840 \define@key{Gin}{class}{}
6841 \end{warpprint}
```

68.4 `\rotatebox`, `\scalebox`, `\reflectbox`

for HTML output: 6842 `\begin{warpHTML}`

`\LWR@rotboxorigin` Holds the origin key letters.

```
6843 \newcommand*\LWR@rotboxorigin{}
```

`\LWR@originname` `{\langle letter \rangle}`

Given one \TeX origin key value, translate into an HTML origin word:

```
6844 \newcommand*\LWR@originname}[1]{%
6845 \ifthenelse{\equal{#1}{t}}{top}{}%
6846 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6847 \ifthenelse{\equal{#1}{c}}{center}{}%
6848 \ifthenelse{\equal{#1}{l}}{left}{}%
6849 \ifthenelse{\equal{#1}{r}}{right}{}%
6850 }
```

`\LWR@originnames` `{\langle letters \rangle}`

Given one- or two-letter \TeX origin key values, translate into HTML origin words:

```

6851 \newcommand*\LWR@originnames}[1]{%
6852 \StrChar{#1}{1}[\LWR@strresult]%
6853 \LWR@originname{\LWR@strresult}
6854 \StrChar{#1}{2}[\LWR@strresult]%
6855 \LWR@originname{\LWR@strresult}
6856 }

```

Handle the origin key for `\rotatebox`:

```

6857 \define@key{krotbox}{origin}{%
6858 \renewcommand*\LWR@rotboxorigin{#1}%
6859 }

```

These keys are ignored:

```

6860 \define@key{krotbox}{x}{}
6861 \define@key{krotbox}{y}{}
6862 \define@key{krotbox}{units}{}

```

`\rotatebox` [*⟨keyval list⟩*] [*⟨angle⟩*] [*⟨text⟩*]

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```

6863 \LetLtxMacro\LWR@origrotatebox\rotatebox
6864
6865 \NewDocumentCommand{\LWR@rotatebox}{0{} m +m}{%

```

Reset the origin to “none-given”:

```

6866 \renewcommand*\LWR@rotboxorigin{}

```

Process the optional keys, which may set `\LWR@rotboxorigin`:

```

6867 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```

6868 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```

6869 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
6870 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```

6871 \LWR@rotstyle{-ms-}{#2} %
6872 \LWR@rotstyle{-webkit-}{#2} %
6873 \LWR@rotstyle{}{#2} %
6874 "{}\LWR@orignewline%

```

Print the text to be rotated:

```

6875 \begin{LWR@nestspan}%
6876 #3%

```

Close the span:

```

6877 \LWR@htmltagc{/span}%
6878 \end{LWR@nestspan}%
6879 }

```

`\scalebox` $\langle h\text{-scale} \rangle$ $[\langle v\text{-scale} \rangle]$ $\langle text \rangle$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```

6880 \LetLtxMacro\LWR@origscalebox\scalebox
6881
6882 \NewDocumentCommand{\LWR@scalebox}{m o m}{%

```

Select inline-block so that HTML will transform this span:

```

6883 \LWR@htmltagc{span style="display: inline-block; %

```

Print the scaling information:

```

6884 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6885 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6886 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6887 "{}}%

```

Print the text to be scaled:

```

6888 \begin{LWR@nestspan}%
6889 #3%

```

Close the span:

```

6890 \LWR@htmltagc{/span}%
6891 \end{LWR@nestspan}%
6892 }

```

`\reflectbox` $\langle text \rangle$

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
6893 \let\LWR@origreflectbox\reflectbox
6894
6895 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}

6896 \end{warpHTML}
```

68.5 Null functions

These functions are not supported by lwarp's HTML conversion.

for HTML output: 6897 `\begin{warpHTML}`

```
\resizebox <{<h-length>} <{<v-length>} <{<text>}
```

Simply prints its text argument.

```
6898 \renewcommand{\resizebox}[3]{#3}

6899 \end{warpHTML}
```

69 Xcolor defaults

The following are used when xcolor is not loaded:

69.1 Defaults without xcolor

for HTML & PRINT: 6900 `\begin{warpall}`

The following is used if xcolor is not loaded, and is replaced when lwarp-xcolor is loaded.

If xcolor is not loaded, there is no xcolor formatting to restore:

```
6901 \newcommand*\LWR@restoreorigxcolor{}
6902 \newcommand*\LWR@restoremorigxcolor{}

6903 \end{warpall}
```

69.2 xcolor print-mode additions

Print-mode versions of new xcolor definitions. 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 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: 6904 `\begin{warpall}`

After `xparse` may have been loaded ...

6905 `\AtBeginDocument{`

... and *only* if xcolor was loaded:

6906 `\@ifpackageloaded{xcolor}{`
6907 `\LWR@traceinfo{patching xcolor}`

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

6908 `\LetLtxMacro\colorboxBlock\colorbox`

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

6909 `\warpprintonly{`
6910 `\LetLtxMacro\LWRprint@colorboxBlock\colorbox`
6911 `\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox`
6912 `\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox`
6913 `}`

`\fcolorbox` [`\framecolor`] [`\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`.)

6914 `\newsavebox{\LWR@colorminipagebox}`
6915
6916 `\DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%`
6917 `\LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%`

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6918 \begin{lrbox}{\LWR@colorminipagebox}%
6919 #5%
6920 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```
6921 \ifthenelse{\equal{#4}{none}}%
6922 {% #4 none
6923   \LWR@traceinfo{background is none}%
6924   {% scope the \colorlet
6925     \colorlet{\LWR@currentcolor}{.}%
6926     \color{#2}%
6927     \fbox{%
6928       \color{\LWR@currentcolor}%
6929       \usebox{\LWR@colorminipagebox}%
6930     }% fbox
6931   }% colorlet
6932 }% #4 none
6933 {% #4 not none
6934 \LWR@traceinfo{background not none}%
6935 \IfValueTF{#1}%
6936 {%
6937   \IfValueTF{#3}%
6938   {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6939   {\LWRorigprint@fcolorbox[#1]{#2}[#4]{\usebox{\LWR@colorminipagebox}}}%
6940 }%
6941 {% no value #1
6942   \IfValueTF{#3}%
6943   {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6944   {\LWRorigprint@fcolorbox{#2}[#4]{\usebox{\LWR@colorminipagebox}}}%
6945 }% no value #1
6946 }% #4 not none
6947 \LWR@traceinfo{\LWRprint@fcolorbox done}%
6948 }
```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
6949 \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:

```
6950 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
```

```
6951 {%
6952 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6953 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
6954 \IfValueTF{#7}%
6955 {\begin{minipage}[#5][#6][#7]{#8}}%
6956 {\begin{minipage}[#5][#6][#5]{#8}}%
6957 }%
6958 {%
6959 \end{minipage}%
6960 \end{lrbox}%
6961 \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`.

```
6962 \ifthenelse{\equal{#4}{none}}%
6963 {% #4 none
6964   {% scope the \colorlet
6965     \colorlet{LWR@currentcolor}{.}%
6966     \color{#2}%
6967     \fbox{%
6968       \color{LWR@currentcolor}%
6969       \usebox{\LWR@colorminipagebox}%
6970     }% fbox
6971   }% colorlet
6972 }% #4 none
6973 {% #4 not none
6974   \IfValueTF{#1}%
6975   {%
6976     \IfValueTF{#3}%
6977     {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6978     {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6979     }%
6980     {% no value #1
6981     \IfValueTF{#3}%
6982     {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6983     {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6984     }% no value #1
6985 }% #4 not none
6986 \LWR@traceinfo{*** finished end fcolorminipage}%
6987 }
```

`\LWR@restoremorigxcolor` 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`.

```
6988 \renewcommand*{\LWR@restoremorigxcolor}{%
6989 \LWR@traceinfo{\LWR@restoremorigxcolorformatting}%
6990 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
6991 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
6992 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
6993 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
6994 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
6995 }
```

If print mode, immediately activate the print-mode enhancements for xcolor:

```
6996 \warpprintonly{\LWR@restoremorigxcolor}
6997
6998 \LWR@traceinfo{xcolor patches done}
6999 }{}% xcolor
7000 }% AtBeginDocument

7001 \end{warpall}
```

70 Cleveref

`Pkg cleveref` cleveref package is used as-is with minor patches.

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 8 on 293 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: `7002 \begin{warpHTML}`

`\AtEndPreamble` forces cleveref to be loaded last:

```
7003 \AtEndPreamble{
7004 \RequirePackage{cleveref}
```

7005 }

The following patches are applied after cleveref has loaded, and after `\AtBeginDocument`:

```
7006 \AfterEndPreamble{
7007 \LWR@traceinfo{Patching cleveref.}
```

```
\@@setcref {<kindofref>} {<label>}
```

```
7008 \renewcommand*\@@setcref}[2]{#1{\ref{#2}}-{}{}}
```

```
\@@setcrefrange {<text>} {<label>} {<label>}
```

```
7009 \renewcommand{\@@setcrefrange}[3]{%
7010 #1{\ref{#2}}-{\ref{#3}}-{}{}}{}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
7011 \newcommand*\cpagerefFor{for}
```

```
\@@setcpageref {<typeofref>} {<label>}, where typeofref is “page” or “pages”
```

```
7012 \renewcommand*\@@setcpageref}[2]{%
7013 #1{\cpagerefFor\ \cref{#2}}-{}{}}%
7014 }
```

```
7015 \renewcommand{\@@setcpagerefrange}[3]{%
7016 #1{\cpagerefFor\ \cref{#2}}-{\cref{#3}}-{}{}}{}}
7017 }% AfterEndPreamble
```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

```
7018 \let\LWR@origlabel\label
7019 \let\label\LWR@newlabel
7020 \let\LWR@origref\ref
7021 \let\ref\LWR@newref% \end{ syntax highlighting
7022 \let\LWR@origpageref\pageref
7023 \let\pageref\LWR@newpageref
7024
7025
7026
7027 \end{warpHTML}
```

71 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: `7028 \begin{warpHTML}`

Env `picture`

```
7029 \BeforeBeginEnvironment{picture}{\lateximage}
7030
7031 \AfterEndEnvironment{picture}{\endlateximage}

7032 \end{warpHTML}
```

72 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.

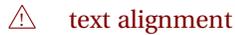
side-by-side Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the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

in a span There is limited support for minipages inside an HTML ``. An HTML `<div>` 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: 7033 `\begin{warpHTML}`

72.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
7034 \newcounter{LWR@minipagedepth}
7035 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
7036 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
7037 \newlength{\LWR@minipageheight}
```

72.2 Footnote handling

Also see section 46 for other forms of footnotes. Minipage footnotes are gathered in section 46.5, and then placed into the document in section 72.3.

72.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the \LaTeX definition and modified to create a `mpfootnotes` div class:

```
7038 \def\LWR@endminipage{%
7039   \par
7040   \unskip
7041   \ifvoid\@mpfootins\else
7042     \vskip\skip\@mpfootins
7043     \normalcolor
7044     \LWR@htmldivclass{mpfootnotes}
7045     \LWR@origmedskip
7046     \unvbox\@mpfootins
7047     \LWR@htmldivclassend{mpfootnotes}
7048   \fi
```

```

7049   \@minipagefalse
7050   \color@endgroup
7051   \egroup
7052   \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}

```

`\LWR@subminipage` Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```

7053 \newcommand*\LWR@subminipage}{%
7054 \LWR@stoppars
7055 \LWR@origminipage{6in}

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

7056 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

7057 \LWR@startpars%
7058 }

```

`\LWR@endsubminipage` Closes the subminipage.

```

7059 \newcommand*\LWR@endsubminipage}{%
7060 \LWR@stoppars%
7061 \LWR@endminipage% The following empty line is required:
7062
7063 }

```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```

7064 \newbool{LWR@minipagefullwidth}
7065 \boolfalse{LWR@minipagefullwidth}

```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output: `7066 \newcommand*\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`
`7067 \end{warpHTML}`

for PRINT output: `7068 \begin{warpprint}`
`7069 \newcommand*\minipagefullwidth}{}`
`7070 \end{warpprint}`

for HTML output: `7071 \begin{warpHTML}`

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
7072 \newbool{LWR@minipagethispar}
7073 \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.

```
7074 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
7075 {%
7076 \uselengthunit{in}%
7077 \setlength{\LWR@minipagewidth}{#4}%
7078 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}0}{%
7079   \addtolength{\LWR@minipagewidth}{3em}% room for frames
7080   \setlength{\linewidth}{6in}%
7081   \setlength{\textwidth}{6in}%
7082   \setlength{\textheight}{9in}%
7083 }{}%
7084 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
7085 \addtocounter{LWR@minipagedepth}{1}%
7086 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
7087 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%
```

ℒ_{TEX} wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
7088 \LWR@stoppars%
```

Create the `<div>` tag with optional alignment style:

```
7089 \LWR@traceinfo{minipage: creating div class}%
7090 \LWR@orignewpage%
7091 \LWR@htmltag{div class="minipage" style="%
7092 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
7093 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
7094 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
7095 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
7096 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
7097 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
7098 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```

7099 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
7100 \uselengthunit{PT}%
7101 \ifbool{LWR@minipagefullwidth}%
7102 {\boolfalse{LWR@minipagefullwidth}}%
7103 {%
7104   \ifthenelse{\lengthtest{#4}=\linewidth}%
7105   {%}
7106   {width:\rndprintlength{\LWR@minipagewidth} ; }%
7107 }%
7108 \LWR@traceinfo{minipage: about to print the height}%
7109 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
7110 "{}%

```

Finish with an empty line to start \LaTeX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

7111
7112 \LWR@origminipage{6in}% The preceding empty line is required.

```

Set the user-accessible minipage and text width and height values inside the minipage. These do not affect the actual size of the large minipage created by `\LWR@origminipage` above, but are used by any reference to `\linewidth`, etc. inside the PDF minipage being created here.

```

7113 \setlength{\linewidth}{#4}% the original width
7114 \setlength{\textwidth}{6in}%
7115 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

7116 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

7117 \LWR@startpars%
7118 \LWR@traceinfo{minipage: finished starting the minipage}%
7119 }

```

End the environment with \LaTeX processing and closing tag:

```

7120 {%
7121 \LWR@stoppars%
7122 \LWR@endminipage% The following empty line is required:
7123
7124 \LWR@htmldivclassend{minipage}%
7125 \LWR@origvspace{1\baselineskip}% required for subcaption

```

```
7126 \addtocounter{LWR@minipagedepth}{-1}%
7127 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
7128 \global\booltrue{LWR@minipagethispar}%
7129 }
```

72.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output: `\parbox` [*⟨pos⟩*] [*⟨height⟩*] [*⟨inner-pos⟩*] {*⟨width⟩*} {*⟨text⟩*}

A parbox uses the minipage code:

```
7130 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
7131 {
7132 \LWR@traceinfo{parbox of width #4}%
7133 \begin{minipage}[#1][#2][#3]{#4}
7134 #5
7135 \end{minipage}
7136 }
```

`\makebox` (<⟨⟩posn) [*⟨width⟩*] [*⟨pos⟩*] {*⟨text⟩*}

```
7137 \LetLtxMacro\LWR@origmakebox\makebox
7138
7139 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
7140 \IfValueTF{#2}%
7141 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
7142   {% scope
7143     \def\LWR@align{center}%
7144     \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
7145     \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
7146     \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
7147 \setlength{\LWR@tempwidth}{#2}%
```

inline-block allows width and text-alignment to be used in a .

```
7148 \uselengthunit{PT}%
7149 \InlineClass[%
7150     display:inline-block ; %
7151     text-align:\LWR@align\ ; %
7152     width:\rndprintlength{\LWR@tempwidth}%
7153 ]%
7154 {makebox}%
```

Without a width argument, the text is simply used inline:

```
7155 {\mbox{#4}}%
7156 }% scope
7157 }%
7158 {\mbox{#4}}%
7159 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```
7160 \LetLtxMacro\LWR@origframebox\framebox
7161
7162 \RenewDocumentCommand{\framebox}{o o m}{%
7163 \fbox{\makebox[#1][#2]{#3}}%
7164 }
```

`\LWR@forceminwidth` {*<legth>*}

Sets `\LWR@tempwidth` to be at least 1pt.

```
7165 \newlength{\LWR@atleastonept}
7166
7167 \newcommand*{\LWR@forceminwidth}[1]{%
7168 \setlength{\LWR@atleastonept}{#1}%
7169 \ifthenelse{%
7170     \lengthtest{\LWR@atleastonept>0pt}\AND%
7171     \lengthtest{\LWR@atleastonept<1pt}%
7172 }%
7173 {\setlength{\LWR@atleastonept}{1pt}}%
7174 {}%
7175 }
```

`\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.

```

7176 \newcommand*\LWR@blackborderpadding}{%
7177 \uselengthunit{PT}%
7178 border:\rndprintlength{\LWR@atleastonept} solid black ; %
7179 padding:\rndprintlength{\fboxsep}%
7180 }

```

`\fbox` $\{ \langle text \rangle \}$

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7181 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```

7182 \newcommand{\LWRhtml@fbox}[1]{%
7183 \LWR@traceinfo{HTML fbox}
7184 \LWR@forceminwidth{\fboxrule}%
7185 \InlineClass[%
7186 \LWR@blackborderpadding%
7187 ]{fbox}{#1}
7188 }

```

xcolor \lets 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:

```
7189 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` $\{ \langle text \rangle \}$ Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```

7190 \newcommand{\fboxBlock}[1]{%
7191 \LWR@forceminwidth{\fboxrule}%
7192 \begin{BlockClass}%
7193 \LWR@blackborderpadding%
7194 ]{fboxBlock}
7195 #1
7196 \end{BlockClass}
7197 }

```

Env `fminipage` $[\langle align \rangle] [\langle height \rangle] [\langle align \rangle] \{ \langle width \rangle \}$

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```

7198 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
7199 {%
7200 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7201 \LWR@forceminwidth{\fboxrule}%
7202 \setlength{\LWR@tempwidth}{#4}%
7203 \IfValueTF{#2}{\setlength{\LWR@tempheight}{#2}}{}%
7204 \begin{BlockClass}[%
7205 \LWR@blackborderpadding ; %
7206 \uselengthunit{PT}]%
7207 \IfValueTF{#2}{height:\rndprintlength{\LWR@tempheight} ; }{}%
7208 width:\rndprintlength{\LWR@tempwidth}%
7209 ]{fminipage}%
7210 }
7211 {%
7212 \end{BlockClass}%
7213 \LWR@traceinfo{fminipage done}%
7214 }

```

`\raisebox` $\langle raiselen \rangle$ $[\langle height \rangle]$ $[\langle depth \rangle]$ $\langle text \rangle$

```

7215 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7216
7217 \RenewDocumentCommand{\raisebox}{m o o m}{%
7218 #4%
7219 }

7220 \end{warpHTML}

```

for HTML & PRINT: 7221 `\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` $[\langle 1:align \rangle]$ $[\langle 2:height \rangle]$ $[\langle 3:inner-align \rangle]$ $\langle 4:width \rangle$

Creates a frame around its contents.

```

7222 \newsavebox{\LWR@fminipagebox}
7223
7224 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
7225 {%

```

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:

```
7226 \IfValueTF{#3}%
7227 {\def\LWR@thisalign{#3}}
7228 {\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.

```
7229 \IfValueTF{#2}%
7230 {\minipage[#1][#2+2\fbboxsep+2\fbboxrule][\LWR@thisalign]{#4+2\fbboxsep+2\fbboxrule}}%
7231 {\minipage[#1]{#4+2\fbboxsep+2\fbboxrule}}%
```

Capture the contents of the environment:

```
7232 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7233 \IfValueTF{#2}%
7234 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7235 {\minipage[#1]{#4}}%
7236 }
7237 {%
```

Close the inner minipage and the LR box with the contents:

```
7238 \endminipage%
7239 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7240 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7241 \endminipage%
7242 }
```

```
7243 \end{warppall}
```

for PRINT output: 7244 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock {<text>}
```

Creates a framed HTML <div> around the text.

```
7245 \let\fboxBlock\fbox
```

Env `fminipage` [*align*] [*height*] [*align*] {*width*}

Creates a frame around its contents.

```
7246 \LetLtxMacro{fminipage}{\LWRprint@fminipage}
7247 \LetLtxMacro{endfminipage}{\endLWRprint@fminipage}

7248 \end{warpprint}
```

73 Direct formatting

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

For high-level block and inline custom CSS classes, see section [40.8](#).

for HTML output: 7249 `\begin{warppHTML}`

`\emph` {*text*}

```
7250 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
7251 \DeclareRobustCommand{\LWR@nulllemp}[1]{#1}
7252 \LetLtxMacro{emph}{\LWR@HTMLemph}
```

`\textmd` {*text*}

```
7253 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspanclass{textmd}{#1}}
7254 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
7255 \LetLtxMacro{textmd}{\LWR@HTMLtextmd}
```

`\textbf` {*text*}

```
7256 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7257 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7258 \LetLtxMacro{textbf}{\LWR@HTMLtextbf}
```

`\textrm` {*text*}

```
7259 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspanclass{textrm}{#1}}
7260 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7261 \LetLtxMacro{textrm}{\LWR@HTMLtextrm}
```

`\textsf` {*text*}

```
7262 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspanclass{textsf}{#1}}
7263 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7264 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` $\{ \langle text \rangle \}$

```
7265 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7266 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7267 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` $\{ \langle text \rangle \}$

```
7268 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspanclass{textup}{#1}}
7269 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7270 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\{ \langle text \rangle \}$

```
7271 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7272 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7273 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\{ \langle text \rangle \}$

```
7274 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspanclass{textsc}{#1}}
7275 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7276 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\{ \langle text \rangle \}$

```
7277 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspanclass{testsl}{#1}}
7278 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7279 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

`\textnormal` $\{ \langle text \rangle \}$

```
7280 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7281 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7282 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
7283 \DeclareRobustCommand{\LWR@nullrmfamily}{-}
7284 \DeclareRobustCommand{\LWR@nullsfamily}{-}
7285 \DeclareRobustCommand{\LWR@nullttfamily}{-}
7286 \DeclareRobustCommand{\LWR@nullbfseries}{-}
7287 \DeclareRobustCommand{\LWR@nullmdseries}{-}
```

```

7288 \DeclareRobustCommand{\LWR@nullupshape}{}
7289 \DeclareRobustCommand{\LWR@nullslshape}{}
7290 \DeclareRobustCommand{\LWR@nullscshape}{}
7291 \DeclareRobustCommand{\LWR@nullitshape}{}
7292 \DeclareRobustCommand{\LWR@nullem}[1]{}
7293 \DeclareRobustCommand{\LWR@nullnormalfont}{}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```

7294 \newcommand*{\LWR@nullfonts}{}%
7295 \LetLtxMacro{\emph}{\LWR@nullemph}%
7296 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7297 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7298 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7299 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7300 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7301 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7302 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7303 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7304 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7305 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7306 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7307 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7308 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7309 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7310 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7311 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7312 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7313 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7314 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7315 \LetLtxMacro{\em}{\LWR@nullem}%
7316 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7317 \renewcommand*{\HTMLUnicode}[1]{}%
7318 \renewcommand*{\HTMLentity}[1]{}%

```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```

7319 \renewcommand*{\&}{and}%
7320 \renewcommand{\textsuperscript}[1]{##1}%
7321 \renewcommand{\textsubscript}[1]{##1}%
7322 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7323 \DeclareExpandableDocumentCommand{\InlineClass}{o m +m}{##3}%
7324 }

```

`\mdseries`

```

7325 \renewcommand*{\mdseries}{}

```

`\bfseries`

7326 `\renewcommand*{\bfseries}{}`

`\rmfamily`

7327 `\renewcommand*{\rmfamily}{}`

`\sffamily`

7328 `\renewcommand*{\sffamily}{}`

`\ttfamily`

7329 `\renewcommand*{\ttfamily}{}`

`\upshape`

7330 `\renewcommand*{\upshape}{}`

`\itshape`

7331 `\renewcommand*{\itshape}{}`

`\scshape`

7332 `\renewcommand*{\scshape}{}`

`\normalfont`

7333 `\renewcommand*{\normalfont}{}`

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

7334 `\renewcommand{\sp}[1]{\text{^{#1}}}`

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

7335 `\renewcommand{\sb}[1]{\text{_{#1}}}`

`\textsuperscript` $\langle text \rangle$

7336 `\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}`

`\textsubscript` $\langle text \rangle$

7337 `\renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}`

`\up` $\langle text \rangle$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

7338 `\AtBeginDocument{\let\up\textsuperscript}`

`\fup` $\langle text \rangle$ Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

7339 `\AtBeginDocument{\let\fup\textsuperscript}`

`\hfill`

7340 `\renewcommand*\hfill{\quad}`

`\hrulefill`

7341 `\renewcommand*\hrulefill{\rule{1in}{1pt}}`

`\dotfill`

7342 `\renewcommand*\dotfill{\dots}`

7343 `\end{warpHTML}`

74 Skips, spaces, font sizes

for HTML output: 7344 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
7345 \let\LWR@origcomma\,
7346 \let\LWR@origtilde~
7347 \let\LWR@origenskip\enskip
7348 \let\LWR@origquad\quad
7349 \let\LWR@origqqquad\qqquad
7350 \let\LWR@orighspace\hspace
7351 \let\LWR@origvspace\vspace
7352 \let\LWR@origrule\rule
7353 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
7354 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space

7355
7356 \renewcommand*{~}{\HTMLentity{nbsp}}
7357
7358 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7359 \let\LWR@orignormalsize\normalsize
7360 \let\LWR@origsmall\small
7361 \let\LWR@origfootnotesize\footnotesize
7362 \let\LWR@origscriptsize\scriptsize
7363 \let\LWR@origtiny\tiny
7364 \let\LWR@origlarge\large
7365 \let\LWR@origLarge\Large
7366 \let\LWR@origLARGE\LARGE
7367 \let\LWR@orighuge\huge
7368 \let\LWR@origHuge\Huge
7369 \renewcommand*{\normalsize}{}
7370 \renewcommand*{\small}{}
7371 \renewcommand*{\footnotesize}{}
7372 \renewcommand*{\scriptsize}{}
7373 \renewcommand*{\tiny}{}
7374 \renewcommand*{\large}{}
7375 \renewcommand*{\Large}{}
7376 \renewcommand*{\LARGE}{}
7377 \renewcommand*{\huge}{}
7378 \renewcommand*{\Huge}{}
7379
7380 \renewcommand*{\onecolumn}{}
7381
7382 \renewcommand{\twocolumn}[1][{}]{
7383
7384 #1
```

```
7385
7386 }
```

`\newline` Uses the HTML `
` element.

```
7387 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7388 \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`.

```
7389 \LetLtxMacro\LWR@origendofline\\
7390 \NewDocumentCommand{\LWR@endofline}{s o}
7391 {%
7392 \newline%
7393 }
```

`\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 `<div>` 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 suppresses 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 suppressed 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 suppress 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.

```
7394 \newcommand*{\LWR@minipagestartpars}{%
7395 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
7396 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qqquad`'s HTML output.

```
7397 \newcommand*{\LWR@minipagestoppars}{%
7398 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
7399 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
7400 \renewcommand*{\quad}{%
7401 \LWR@minipagestoppars%
7402 \HTMLUnicode{2001}%
7403 \LWR@minipagestartpars%
7404 }
```

`\qqquad` Handles special minipage & horizontal space interactions.

```
7405 \renewcommand*{\qqquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
7406 \renewcommand*{\enskip}{%
7407 \LWR@minipagestoppars%
7408 \HTMLUnicode{2000}%
7409 \LWR@minipagestartpars%
7410 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

Len `\WR@tempheight`

Len `\WR@tempraise`

```
7411 \newlength{\WR@tempwidth}
7412 \newlength{\WR@tempheight}
7413 \newlength{\WR@tempraise}
```

`\LWR@hspace` * $\langle length \rangle$

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 `\qqquad`.

```
7414 \NewDocumentCommand{\LWR@hspace}{s m}{%
7415 \setlength{\WR@tempwidth}{#2}%
```

If `\fill`, change to `\qqquad`:

```
7416 \ifnum\gluestretchorder\LWR@tempwidth>0%
```

```
7417 \setlength{\LWR@tempwidth}{2em}%
7418 \fi%
```

Only if the width is not zero:

```
7419 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7420 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7421 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
7422 {%
7423 \HTMLUnicode{2009}}% thin breakable space
7424 }%
```

Print the span with the converted width. Not rounded.

```
7425 {%
7426 \uselengthunit{PT}%
7427 \LWR@htmltagc{%
7428 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
7429 }%
7430 \LWR@htmltagc{/span}%
7431 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7432 \LWR@minipagestartpars%
7433 }% width not 0
7434 }
```

```
\LWR@nohspace * {<length>}
```

Used to disable \hspace while creating description \items.

```
7435 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

```
\hspace * {<length>}
```

Handles special minipage & horizontal space interactions.

```
7436 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * $\langle length \rangle$ Nullified vspace.

```
7437 \NewDocumentCommand{\LWR@vspace}{s m}{}
```

`\vspace` * $\langle length \rangle$ Nullified.

```
7438 \let\vspace\LWR@vspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML br tag.

```
7439 \renewcommand*\linebreak[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
7440 \renewcommand*\nolinebreak[1] [] {}
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
7441 \renewcommand*\pagebreak[1] [] {
7442
7443 }
```

`\nopagebreak` [$\langle num \rangle$]

```
7444 \renewcommand*\nopagebreak[1] [] {}
```

`\enlargethispage` * $\langle len \rangle$

```
7445 \RenewDocumentCommand{\enlargethispage}{s m}{}
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
7446 \newcommand*\LWR@currenttextcolor}{black}
```

`\LWR@rule` [$\langle raise \rangle$] $\langle width \rangle$ $\langle height \rangle$

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.

```
7447 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
7448 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
7449 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
7450 {}% zero- width
7451 {% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
7452     \ifthenelse{%
7453         \lengthtest{\LWR@tempwidth>0pt}\AND%
7454         \lengthtest{\LWR@tempwidth<1pt}}%
7455     }%
7456     {\setlength{\LWR@tempwidth}{1pt}}{}}%
```

Likewise with height:

```
7457     \setlength{\LWR@tempheight}{#3}%
7458     \ifthenelse{%
7459         \lengthtest{\LWR@tempheight>0pt}\AND%
7460         \lengthtest{\LWR@tempheight<1pt}}%
7461     }%
7462     {\setlength{\LWR@tempheight}{1pt}}{}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7463     \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 \TeX code.

```
7464     \uselengthunit{PT}%
7465     \LWR@htmltagc{%
7466     span
7467     style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
7468     background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
7469 width:\printlength{\LWR@tempwidth}; %
7470 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 \TeX length allows a typical \TeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \TeX length limits the allowable syntax. To do: A superior method would compute a ratio of \TeX ex height, then print that to HTML with an ex unit.

```
7471 \IfValueTF{#1}%
7472 {%
7473   \setlength{\LWR@tempraise}{Opt-#1}%
7474   \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7475   \LWR@orignewline%
7476   -ms-transform: translate(Opt,\printlength{\LWR@tempraise}); %
7477   \LWR@orignewline%
7478   -webkit-transform: translate(Opt,\printlength{\LWR@tempraise}); %
7479   \LWR@orignewline%
7480   transform: translate(Opt,\printlength{\LWR@tempraise}); %
7481   \LWR@orignewline%
7482 }{}%
```

Display inline-block to place the span inline with the text:

```
7483 display:inline-block;"%
7484 }%
7485 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7486 \LWR@minipagestartpars%
7487 }% non-zero width
7488 }
```

`\rule` [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

```
7489 \renewcommand{\rule}{\LWR@rule}
```

```
7490 \end{warpHTML}
```

75 `\phantomsection`

for HTML output: 7491 `\begin{warpHTML}`

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
7492 \newcommand*\phantomsection{%
7493 \section*{}%
7494 }

7495 \end{warpHTML}
```

76 `\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>

76.1 HTML logos

for HTML output: 7496 `\begin{warpHTML}`

`\TeX` \TeX

`latexlogo` is a css class used to properly typeset the E and A in \TeX and friends.

`latexlogofont` is a css class used to select the font for the rest of the logo in \TeX , $\text{Lua}\TeX$, $\text{Con}\TeX$ t, etc.

```
7497 \newcommand*\LWR@TeX}
7498 {\InlineClass{latexlogofont}%
7499 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` \TeX , $\text{TeX}2\epsilon$
`\LaTeXe`

```

7500 \newcommand*{\LWR@LaTeX}
7501 {\InlineClass{latexlogofont}%
7502 {\InlineClass{latexlogo}%
7503 {\L\textsuperscript{a}T\textsubscript{e}X}}
7504
7505 \renewcommand*{\LaTeXe}
7506 {\LaTeX\InlineClass{latexlogofont}%
7507 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}

```

`\LuaTeX` Lua \TeX , Lua $\mathbb{T}\TeX$
`\LuaLaTeX`

```

7508 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}\Lua\TeX}
7509 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}\Lua\LaTeX}

```

`\XeTeX` X $\mathbb{T}\TeX$, X $\mathbb{E}\mathbb{T}\TeX$
`\XeLaTeX`

xetexlogo is a css class which aligns the backwards E in X $\mathbb{T}\TeX$ and spaces $\mathbb{T}\TeX$ appropriately.

xelatexlogo is a css class which aligns the backwards E in X $\mathbb{E}\mathbb{T}\TeX$ and spaces $\mathbb{E}\mathbb{T}\TeX$ appropriately.

```

7510 \newcommand*{\Xe}
7511 {X\textsubscript{\HTMLUnicode{18e}}}
7512 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}\Xe\TeX}
7513 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}\Xe\LaTeX}

```

`\ConTeXt` Con $\mathbb{T}\mathbb{E}\mathbb{X}t$

```

7514 \newcommand*{\LWR@ConTeXt}
7515 {\InlineClass{latexlogofont}\Con\TeX{}}
7516 \InlineClass{latexlogofont}{t}}

```

`\BibTeX` BiB $\mathbb{T}\mathbb{E}\mathbb{X}$, *MakeIndex*
`\MakeIndex`

```

7517 \providecommand*{\BibTeX}
7518 {\InlineClass{latexlogofont}\B\textsc{ib}\TeX}
7519
7520 \newcommand*{\MakeIndex}
7521 {\InlineClass{latexlogofont}\textit{MakeIndex}}

```

`\AmS` $\mathcal{A}\mathcal{M}\mathcal{S}$

amslogo is a css class used for the $\mathcal{A}\mathcal{M}\mathcal{S}$ logo.

```

7522 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
7523 {\InlineClass{amslogo}\textit{A\textsubscript{M}S}}}

```

`\MiKTeX` `MiKTeX`

```
7524 \newcommand*\MiKTeX{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the LyX logo.

```
7525 \newcommand*\LyX{\InlineClass{lyxlogo}{LyX}}
```

```
7526 \end{warpHTML}
```

76.2 Print logos

for PRINT output:

```
7527 \begin{warpprint}
7528 \newcommand*\XeTeXrevE}
7529   {\hspace{- .1667em}\raisebox{- .5ex}{\reflectbox{E}}\hspace{- .125em}}
7530 \providecommand*\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
7531 \providecommand*\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
7532 \providecommand*\AmS}{%
7533 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
7534 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
7535 \newcommand*\LyX}{\textsf{LyX}}
7536 \providecommand*\LuaTeX}{\mbox{Lua\TeX}}
7537 \providecommand*\LuaLaTeX}{\mbox{Lua\LaTeX}}
7538 \providecommand*\BibTeX}{\mbox{B\textsc{ib}\TeX}}
7539 \providecommand*\MakeIndex}{\mbox{\textit{MakeIndex}}}
7540 \providecommand*\ConTeXt}{\mbox{Con\TeXt}}
7541 \providecommand*\MiKTeX}{\mbox{MiK\TeX}}
7542 \end{warpprint}
```

77 \AtBeginDocument, \AtEndDocument

for HTML output: `7543 \begin{warpHTML}`

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
7544 \AfterEndPreamble{\LWR@LwarpStart}
```

```
7545 \AtEndDocument{\LWR@LwarpEnd}
```

```
7546 \end{warpHTML}
```

78 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.

Package 2

lwarp-abstract.sty

79 Abstract

(Based on original 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: Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitlekip}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs

```

```
29     \else
30     \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32     \endparpenalty\M
33     \end\absnamepos%
34 %%     \vspace{\abstitlekip}%
35     \fi
36     \fi
37     \vspace{\abstitlekip}%
38     \fi
39     \put@bsintoc%
40     \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41     {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42     }
43 \else
44 \renewenvironment{abstract}{%
45     \if@bsrunin
46     \else
47     \if@bsstyle
48     \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49     \else
50     \ifnumber@bs
51     \num@bs
52     \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%     \vspace{\abstitlekip}%
57     \fi
58     \fi
59     \vspace{\abstitlekip}%
60     \fi
61     \put@bsintoc%
62     \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63     {\par\end{@bstr@ctlist}}
64 \fi
65
```

Package 3

lwarp-afterpage.sty

80 Afterpage

Pkg afterpage Emulated.

for HTML output: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

Package 4

lwarp-algorithmicx.sty

81 Algorithmicx

Pkg algorithmicx algorithmicx is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithmicx}

Inside the algorithmic environment, level indenting is converted to a `` of the required length, and comments are placed inside a `` which is floated right.

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 209.1.

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*\ALG@doentity{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand\Comment[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }

20 \end{warpHTML}
```

Package 5

lwarp-alltt.sty

82 Alltt

Pkg alltt alltt is patched for use by lwarp.

```
for HTML output: 1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

Package 6

lwarp-amsthm.sty

83 AMSthm

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class amsthmbody<theoremstyle>
Theorem Name: of class amsthmname<theoremstyle>
Theorem Number: of class amsthmnumber<theoremstyle>
Theorem Note: of class amsthmnote<theoremstyle>
Proof: <div> of class amsthmproof
Proof Name: of class amsthmproofname
 where <theoremstyle> is plain, definition, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \@ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```

13 \def\xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let@tempa\relax
16   \exp@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef@tempa##1{%
20         \gdef\exp@nx\csname#2\endcsname{%
21           \@nx@thm{\@xp@nx\csname th@the\thm@style\endcsname}%
22             {##1}}}%
23     \else % numbered theorem, need to check for optional arg
24       \def@tempa{\@oparg{\ynthm{#2}} []}%
25     \fi
26     \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27   }%
28   \@tempa
29 }

```

Patched to enclose with css:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for css:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47 \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48 \deferred@thm@head{
49   \the\thm@headfont \thm@indent
50   \@ifempty{#1}{\let\thmname@gobble}{\LWR@haveamsthmname}% new
51   \@ifempty{#2}{\let\thmnumber@gobble}{\LWR@haveamsthmnumber}% new
52   \@ifempty{#3}{\let\thmnote@gobble}{\LWR@haveamsthmnote}% new
53   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54   \the\thm@headpunct~
55   \thmheadnl % possibly a newline.

```

```
56 \hskip\thm@headsep
57 }%
58 \ignorespaces}
```

Patched for CSS:

```
59 \def\endtheorem{\endBlockClass\endtrivlist\@endpefalse }
```

Proof QED symbol:

```
60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }
```

Patched for CSS:

```
65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67 \BlockClass{amsthmproof}% new
68 \pushQED{\qed}%
69 \normalfont \topsep6\p@\@plus6\p@\relax
70 \trivlist
71 \item[\hskip\labelsep
72 \InlineClass{amsthmproofname}{#1\@addpunct{.}}\ignorespaces% changes
73 }{%
74 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75 \endBlockClass% new
76 \@endpefalse
77 }
```

Package 7

lwarp-appendix.sty

84 Appendix

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:

```
1 \LWR@ProvidesPackagePass{appendix}
2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }
15
```

Package 8

lwarp-arabicfront.sty

85 Arabicfront

Pkg arabicfront arabicfront is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{arabicfront}

Package 9

lwarp-balance.sty

86 Balance

Pkg balance Emulated.

for HTML output: Discard all options for lwarp-balance:

1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}

3 \newcommand*{\nobalance}{}

Package 10

lwarp-bookmark.sty

87 Bookmark

Pkg bookmark bookmark is emulated during HTML output, and the bookmark package is ignored.
 for HTML output: Discard all options for lwarp-bookmark:

```

1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*{\bookmarksetup}[1]{}
3 \newcommand*{\bookmarksetupnext}[1]{}
4 \newcommand*{\bookmark}[2] [] {}
5 \newcommand*{\bookmarkdefinestyle}[2]{}
6 \newcommand*{\bookmarkget}[1]{}
7 \newcommand{\BookmarkAtEnd}[1]{}

```

Package 11

lwarp-booktabs.sty

88 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

for HTML output: `1 \LWR@ProvidesPackagePass{booktabs}`

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML lateximages. The HTML versions temporarily overwrite these print versions when tabular is started.

```

2 \LetLtxMacro\LWR@origtoprule\toprule
3 \LetLtxMacro\LWR@origmidrule\midrule
4 \LetLtxMacro\LWR@origcmidrule\cmidrule
5 \LetLtxMacro\LWR@origbottomrule\bottomrule
6 \LetLtxMacro\LWR@origaddlinespace\addlinespace
7 \LetLtxMacro\LWR@origmorecmidrules\morecmidrules

```

8 \LetLtxMacro\LWR@origspecialrule\specialrule

Package 12

lwarp-boxedminipage.sty

89 Boxedminipage

Pkg boxedminipage boxedminipage is superceded by boxedminipage2e.
for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

Package 13

lwarp-boxedminipage2e.sty

90 Boxedminipage2e

Pkg boxedminipage2e boxedminipage2e is emulated.
for HTML output: Discard all options for lwarp-boxedminipage2e:
1 \LWR@ProvidesPackageDrop{boxedminipage2e}

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }

Package 14

lwarp-caption2.sty

91 Caption2

Pkg caption2 caption2 is not used. The user is recommended to use caption instead.

for HTML output: `1 \LWR@loadnever{caption2}{caption}`

Package 15

lwarp-ccaption.sty

92 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: `1 \LWR@loadnever{ccaption}{caption}`

Package 16

lwarp-changepage.sty

93 Changepage

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

Package 17

lwarp-chappg.sty

94 Chappg

Pkg chappg chappg is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{chappg}
2 \renewcommand{\pagenumbering}[2] [] {}
3 \providecommand{\chappgsep}{--}
```

Package 18

lwarp-color.sty

95 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:

```
1 \LWR@ProvidesPackagePass{color}
2 \RequirePackage{xcolor}
```

Package 19

lwarp-crop.sty

96 Crop

`Pkg crop` Emulated.

for HTML output: Discard all options for `lwarp-crop`:

```
1 \LWR@ProvidesPackageDrop{crop}

2 \newcommand*{\crop}[1] [] {}
3 \newcommand*{\cropdef}[6] [] {}
```

Package 20

lwarp-cutwin.sty

97 Cutwin

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```
1 \LWR@ProvidesPackageDrop{cutwin}

2 \newcommand*\opencutleft{}
3 \newcommand*\opencutright{}
4 \newcommand*\opencutcenter{}
5 \newcommand*\cutfuzz{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}
9 {}
10
11 \newcommand*\windowpagestuff{}
12
13 \newcommand*\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}
21 {}
22
23 \newcommand*\putstuffinpic{}
24
25 \newcommand*\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}
```

Package 21

lwarp-dcolumn.sty

98 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 22

lwarp-draftwatermark.sty

99 Draftwatermark

Pkg draftwatermark draftwatermark is emulated during HTML output, and the draftwatermark package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{draftwatermark}
2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

Package 23

lwarp-ellipsis.sty

100 Ellipsis

Pkg ellipsis ellipsis is emulated during HTML output, and the ellipsis package is ignored.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

Package 24

lwarp-emptypage.sty

101 Emptypage

Pkg emptypage emptypage is ignored.

for HTML output: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 25

lwarp-endnotes.sty

102 Endnotes

(Based on original 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: `1 \LWR@ProvidesPackagePass{endnotes}`

```
2 \def\enoteformat{%
3 % \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5 % \llap{
6 \makeenmark
7 % }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

Package 26

lwarp-enumerate.sty

103 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: `1 \LWR@ProvidesPackagePass{enumerate}`

Package 27

lwarp-enumitem.sty

104 Enumitem

Pkg enumitem enumitem is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{enumitem}

for HTML output: 2 \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.

```
3 \let\LWR@orignewlist\newlist
4
5 \renewcommand*{\newlist}[3]{%
6 \LWR@orignewlist{#1}{#2}{#3}%
7 \AtBeginEnvironment{#1}{\csuse{LWR@#2start}}%
8 \AtEndEnvironment{#1}{\csuse{LWR@#2end}}%
9 }

10 \end{warpHTML}
```

Package 28

lwarp-epigraph.sty

105 Epigraph

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}

```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{flushepinormal}{}{}
24 \newcommand{\textflush}[1]{flushepinormal}
25 \newcommand{\epigraphflush}[1]{flushright}
26 \newcommand{\sourceflush}[1]{flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}

```

```

31 \newcommand{\epigraphhead}[2][0]{#2}
32 \newcommand{\dropchapter}[1]{}
33 \newcommand*\undodrop{}
34 \newcommand{\cleartoevenpage}[1][{}]{

```

Package 29

lwarp-eso-pic.sty

106 Eso-pic

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*\LenToUnit{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*\ClearShipoutPictureBG{}
16 \newcommand*\ClearShipoutPicture{}
17 \newcommand*\ClearShipoutPictureFG{}
18 \newcommand{\gridSetup}[6][{}]{

```

Package 30

lwarp-everypage.sty

107 Everypage

Pkg everypage everypage is emulated during HTML output, and the everypage package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{everypage}
2 \newcommand*{\AddEverypageHook}[1]{}
3 \newcommand*{\AddThispageHook}[1]{}

```

Package 31

lwarp-everyshi.sty

108 Everyshi

Pkg everyshi Emulated.

for HTML output: Discard all options for lwarp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}
2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}

```

Package 32

lwarp-extramarks.sty

109 Extramarks

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for lwarp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}
2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}

```

Package 33

lwarp-fancybox.sty

110 Fancybox

(Based on original 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:

```
1 \begin{warpHTML}
2 \LWR@ProvidesPackagePass{fancybox}
3 \renewcommand*{\@shadowbox}[1]{\InlineClass{shadowbox}{#1}}
4
5 \renewcommand*{\@doublebox}[1]{\InlineClass{doublebox}{#1}}
6
7 \renewcommand*{\@ovalbox}[2]{%
8 \ifthenelse{\isequivalentto{#1}{\thinlines}}
9 {\InlineClass{ovalbox}{#2}}
10 {\InlineClass{Ovalbox}{#2}}
11 }
```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```
12 \let\LWR@origSbox\Sbox
13
14 \def\Sbox{\LWR@origSbox\LWR@nestspan}
15
16
17 \let\LWR@origendSbox\endSbox
18
19 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}
```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```
20 \RenewEnviron{Beqnarray}
```

```

21 {\LWR@eqnarrayfactor}
22
23 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

24 \renewcommand{\GenericCaption}[1]{%
25 \LWR@htmlblocktag{figcaption}%
26 #1%
27 \LWR@htmlblocktag{/figcaption}%
28 }

```

Btrivlist is enclosed in an HTML block:

```

29 \RenewDocumentEnvironment{Btrivlist}{m o}
30 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
31 {\endtabular\end{BlockClass}}

```

Btrivlist is also neutralized when used inside a span:

```

32 \AtBeginEnvironment{LWR@nestspan}{%
33 \RenewDocumentEnvironment{Btrivlist}{m o}{}}%
34 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
35 }

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

36 \let\LWRFB@origitemizeitem\LWR@itemizeitem
37 \let\LWRFB@origdescitem\LWR@descitem
38 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
39
40 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
41 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

42 \renewenvironment{Bitemize}[1][\begin{itemize}]{\end{itemize}}
43 \renewenvironment{Benumerate}[1][\begin{enumerate}]{\end{enumerate}}
44 \renewenvironment{Bdescription}[1][\begin{description}]{\end{description}}

```

\boxput simply prints one then the other argument, side-by-side instead of above and behind:

```

45 \RenewDocumentCommand{\boxput}{s d() m m}{%
46 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
47 }

```

Neutralized commands:

```

48 \RenewDocumentCommand{\fancyput}{s d() m}{}
49 \RenewDocumentCommand{\thisfancyput}{s d() m}{}
50
51 \RenewDocumentCommand{\fancypage}{m m}{}
52 \RenewDocumentCommand{\thisfancypage}{m m}{}
53
54 \def\LandScape#1{}
55 \def\endLandScape{}
56 \def\@Landscape#1#2#3{}
57 \def\endLandscape{}

```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```

58 \let\LWRFB@UseVerbatim\UseVerbatim
59 \renewcommand*{\UseVerbatim}[1]{%
60 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{- .5\baselineskip}%%
61 \LWRFB@UseVerbatim{#1}%
62 \LWR@afterendverbatim%
63 }
64
65 \let\LWRFB@LUseVerbatim\LUseVerbatim
66
67 \renewcommand*{\LUseVerbatim}[1]{%
68 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
69 \noindent%
70 \LWRFB@LUseVerbatim{#1}%
71 \LWR@afterendverbatim%
72 }
73
74 \def\@BUseVerbatim[#1]#2{%
75 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{- .5\baselineskip}%%
76 \LWRFB@UseVerbatim{#2}%
77 \LWR@afterendverbatim%
78 }

79 \end{warpHTML}

```

Package 34

lwarp-fancyhdr.sty

111 Fancyhdr

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2] [] {}
3 \newcommand*{\fancyfoot}[2] [] {}
4 \newcommand*{\fancyhf}[2] [] {}
5 \newcommand*{\fancyheadstyle}[2] {}
6 \newcommand*{\lhead}[2] [] {}
7 \newcommand*{\chead}[2] [] {}
8 \newcommand*{\rhead}[2] [] {}
9 \newcommand*{\lfoot}[2] [] {}
10 \newcommand*{\cfoot}[2] [] {}
11 \newcommand*{\rfoot}[2] [] {}
12 \newcommand*{\headrulewidth}{}
13 \newcommand*{\footrulewidth}{}
14 \newcommand*{\fancyheadoffset}[2] [] {}
15 \newcommand*{\fancyfootoffset}[2] [] {}
16 \newcommand*{\fancyhfoffset}[2] [] {}
17 \newcommand*{\iffloatpage}[2] {#2}
18 \newcommand*{\ifftopfloat}[2] {#2}
19 \newcommand*{\iffbotfloat}[2] {#2}

```

Package 35

lwarp-fancyvrb.sty

112 Fancyvrb

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg fancyvrb fancyvrb is supported with some patches.

for HTML output: 1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{fancyvrb}

for HTML output: 3 \begin{warpHTML}

Initial default patch for fancyvrb:

```
4 \fvset{frame=none}%
```

For \VerbatimFootnotes:

```
5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
8 {This may be improved some day.}
9 }
```

After the preamble is loaded, after any patches to Verbatim:

```
10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}
```

Remember the original defintion of Verbatim:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

Env Verbatim Patched to place the environment in a fancyvrb div, and the label in a fancyvrblabel div. Also corrects the left margin for line numbers. Also uses VerbatimHTMLWidth to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*{\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
17 \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.

```
18 \newcommand*\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@\LWR@afterendverbatim}%
26 }
27
28 \newcommand*\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@EndListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@EndListFrame@Lines%
49 \LWR@fvendnone%
50 }
```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```
51
52 \newcommand*\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{LWR@current@color}{.}%
54 \protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
```

```

55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@EndListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@EndListFrame\LWR@fvendsingle}
93
94 \def\FV@Frame@lines{%
95 \renewcommand*{\LWR@FVstyle}{%
96   \LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
97 }%
98 \let\FV@BeginListFrame\LWR@fvstartline%
99 \let\FV@LeftListFrame\relax%
100 \let\FV@RightListFrame\relax%
101 \let\FV@EndListFrame\LWR@fvendline}
102
103 \def\FV@Frame@topline{%
104 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%

```

```

105 \let\FV@BeginListFrame\LWR@fvstartline%
106 \let\FV@LeftListFrame\relax%
107 \let\FV@RightListFrame\relax%
108 \let\FV@EndListFrame\LWR@fvendnone}
109
110 \def\FV@Frame@bottomline{%
111 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}}%
112 \let\FV@BeginListFrame\LWR@fvstartnone%
113 \let\FV@LeftListFrame\relax%
114 \let\FV@RightListFrame\relax%
115 \let\FV@EndListFrame\LWR@fvendline}
116
117 \def\FV@Frame@leftline{%
118 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}}%
119 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
120 \ifx\FancyVerbFillColor\relax%
121 \let\FV@FrameFillLine\relax%
122 \else%
123 \@tempdima\FV@FrameRule\relax%
124 \multiply\@tempdima-\tw@%
125 \edef\FV@FrameFillLine{%
126 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
127 \kern-\number\@tempdima sp}}%
128 \fi%
129 \let\FV@BeginListFrame\LWR@fvstartnone%
130 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
131 \let\FV@RightListFrame\relax%
132 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

133 \def\FV@SingleFrameLine#1{%
134 \hbox to\z@{%
135 % \kern\leftmargin
136 \ifnum#1=\z@\relax
137 \let\FV@Label\FV@LabelBegin
138 \else
139 \let\FV@Label\FV@LabelEnd
140 \fi
141 \ifx\FV@Label\relax
142 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
143 \else
144 \ifnum#1=\z@
145 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
146 \ifx\FV@LabelPositionTopLine\relax
147 \else
148 \LWR@FVfindbordercolor
149 \LWR@htmltagc{div class="fancyvrblabel" style="color:\#\LWR@tempcolor"}

```

```

150     \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
151     \LWR@htmltagc{/div}
152     \fi
153   \else
154 %     \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
155     \ifx\FV@LabelPositionBottomLine\relax
156     \else
157     \LWR@FVfindbordercolor
158     \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
159     \LWR@origtextrm{\FV@LabelEnd}
160     \LWR@htmltagc{/div}
161     \fi
162   \fi
163
164 \fi
165 \hss
166 }
167 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

168 \def\FV@ListProcessLine#1{%
169   \hbox to \hsize{%
170 %     \kern\leftmargin
171     \hbox to \VerbatimHTMLWidth {%
172       \ifcvoid{FV@LeftListNumber}{}{\kern 2.5em}%
173       \FV@LeftListNumber%
174 %     \FV@LeftListFrame
175     \FancyVerbFormatLine{#1}%
176     \hss%
177 %     \FV@RightListFrame
178     \FV@RightListNumber%
179   }%
180   \hss% required to avoid underfull hboxes
181 }
182 }

```

Env **BVerbatim**

```

183 \AtBeginEnvironment{BVerbatim}
184 {
185 \LWR@forcenewpage
186 \LWR@atbeginverbatim{bverbatim}
187
188 }
189
190 \AfterEndEnvironment{BVerbatim}
191 {

```

```
192 \leavevmode\par\LWR@origvspace{-\baselineskip}  
193 \LWR@afterendverbatim  
194 }
```

End of the modifications to make at the end of the preamble:

```
195 } % \AfterEndPreamble
```

```
196 \end{warpHTML}
```

Package 36

lwarp-figcaps.sty

113 Figcaps

Pkg figcaps Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```
1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}
```

Package 37

lwarp-fix2col.sty

114 Fix2col

Pkg fix2col fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

Package 38

lwarp-float.sty

115 Float and \newfloat

`\pkg float` float is emulated during HTML output, and the float package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{float}[2016/03/04]`

See section [62.2](#) for the `\listof` command.

`\newfloat` $\langle 1: type \rangle \langle 2: placement \rangle \langle 3: ext \rangle [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
5 {\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.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

`\floatname` $\langle type \rangle \langle name \rangle$

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement` $\langle type \rangle \langle placement \rangle$

Float placement is ignored.

```
12 \newcommand*{\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
```

```
14 }
```

```
\floatstyle {<style>}
```

Float styles are ignored.

```
15 \newcommand{\floatstyle}[1]{%  
16 }
```

```
\restylefloat * {<style>}
```

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%  
18 }
```

Package 39

lwarp-floatflt.sty

116 Floatflt

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16 {\begin{KFLTfloatflt@marginfloat}{figure}}
17 {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {\begin{KFLTfloatflt@marginfloat}{table}#2}
21 {\end{KFLTfloatflt@marginfloat}}
```

Package 40**lwarp-floatpag.sty****117 Floatpag**

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]{}

```

Package 41

lwarp-floatrow.sty

118 Floatrow

`\pkg floatrow` floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{floatrow}`

 **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:

```
2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }
```

```
\floatbox [1 preamble] [2 capttype] [3 width] [4 height] [5 vert pos]
{6 caption} {7 object}
```

Only parameters for `capttype`, `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.

```

9 \NewDocumentCommand{\floatbox}{o m o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For subcaption:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

subfigure and subtable environments take width as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

figure and table environments do not take a width argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For subfig:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% 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.

```

31 \beginngroup

```

```

32 \let\caption\@firstofone
33 \subfloat[#6]{#7}
34 \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

figure and table are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*\nocapbeside{}
46 \newcommand*\capbeside{}
47 \newcommand*\captop{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*\useFCwidth{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*\killfloatstyle{}

```

Preamble and default width are ignored.

```

57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }

```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }

```

```

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside][

```

```
68 \newfloatcommand{ttabbox}{table}[\captop] [\FBwidth]
```

```
69 \newfloatcommand{fcapside}{figure}[\capbeside] []
```

The row of floats is placed into a <div> of class floatrow.

```
70 \newenvironment*{floatrow}[1] [2]
71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
74 \booltrue{LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{LWR@infloatrow}
79 \endBlockClass
80 }
```

Keys for \DeclareNewFloatType:

```
81 \newcommand*{\LWR@frowkeyplacement}{}
82 \newcommand*{\LWR@frowkeyname}{}
83 \newcommand*{\LWR@frowkeyfileext}{}
84 \newcommand*{\LWR@frowkeywithin}{}
85 \newcommand*{\LWR@frowkeycapstyle}{}
86
87 \define@key{frowkeys}{placement}{}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use \listof{type}{Title} to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2] {%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}
94 \renewcommand*{\LWR@frowkeyname}{}
95 \renewcommand*{\LWR@frowkeyfileext}{}
96 \renewcommand*{\LWR@frowkeywithin}{}
97 \renewcommand*{\LWR@frowkeycapstyle}{}

```

Read new key values:

```

98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}
102 {
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
104     \LWR@frowkeyfileext}%
105   \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
109     \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{#1}{\LWR@frowkeyplacement}%
111   {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112   \LWR@traceinfo{finished newfloat #1}
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}
120 \newcommand*{\TopFloatBoxes}{}
121 \newcommand*{\BottomFloatBoxes}{}
122 \newcommand*{\PlainFloatBoxes}{}
123
124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS <div> of class floatfoot.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{}
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}

```

```
132 }
```

Used to compute `\linewidth`.

```
133 \newbool{LWR@infloatrow}
134 \boolfalse{LWR@infloatrow}
```

```
135 \newenvironment*{subfloatrow}[1][2]
136 {
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
137 \LWR@forcenewpage
138 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@infloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
139 \begingroup
140 \booltrue{LWR@infloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@infloatrow}
146 }
```

Package 42

lwarp-flushend.sty

119 Flushend

Pkg flushend Emulated.

for HTML output: Discard all options for lwarp-flushend:

```
1 \LWR@ProvidesPackageDrop{flushend}
2%   \end{ma-crocode}
3%
4%   \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```

Package 43

lwarp-fncychap.sty

120 Fncychap

Pkg fncychap fncychap is ignored.

for HTML output: Discard all options for lwarp-fncychap:

```

1 \LWR@ProvidesPackageDrop{fncychap}

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}
9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}
13 \newcommand{\TheAlphaChapter}{}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}

```

Package 44

lwarp-fontenc.sty

121 Fontenc

Pkg fontenc Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}

2 \LWR@loadbefore{fontenc}

Package 45

lwarp-fontspec.sty

122 Fontspec

Pkg fontspec Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontspec}

2 \LWR@loadbefore{fontspec}

Package 46

lwarp-footmisc.sty

123 Footmisc

(Based on original code by ROBIN FAIRBAIRNS.)

Pkg footmisc footmisc is emulated during HTML output, and the footmisc package is ignored.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[1]{}
4 \NewDocumentCommand{\DefineFNsymbols}{s m o m}{}
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker}{3sp}
17 \providecommand*\multfootsep}{,}
```

Using cleveref:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar [%
21     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
}
```

```
27 }
28 \def\xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

Package 47

lwarp-footnote.sty

124 Footnote

Pkg footnote footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```

2 \def\fn@startnote{%
3   \@parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5   \color@begingroup% *** conflicts with lwarp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*\fn@endnote}{%
10 \LWR@htmltagc{/\LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }
```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15   \fn@startnote%
16   \fn@prefntext%
17   \ignorespaces%
18 }
```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/\LWR@tagregularparagraph}%
21   \LWR@orignewline%
22   \fn@postfntext%
23 \egroup%
24 \begingroup%
25   \let\@makefntext\@empty%
26   \let\@finalstrut\@gobble%
27   \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28   \@footnotetext{\unvbox\z@}%
29 \endgroup%
```

```
30 }
```

These have been redefined, so re-\let them again:

```
31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote
```

Package 48

lwarp-footnotehyper.sty

125 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:

```
1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

Package 49

lwarp-framed.sty

126 Framed

(Based on original code by DONALD ARSENEAU.)

Pkg framed framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{framed}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{framed}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage

```

```

34   \BlockClass{framedleftbar}
35   \def\FrameCommand{}%
36   \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{-}{Opt}{Opt}{Opt}{Opt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{-}{% not empty
67   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68   \textcolor{TFTitleColor}{\textbf{#1}}%
69   \end{BlockClass}
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{-}{% not empty
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77   \textcolor{TFTitleColor}{\textbf{#2}}%
78   \end{BlockClass}

```

```
79 }% not empty
80 \end{BlockClass}
81 }

\TitleBarFrame [marker] {title} {contents}

82 \renewcommand\TitleBarFrame[3][]{
83 \CustomFBox
84   {#2}{}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {settings}

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame {frame cmd no split} {frame cmd split}

95 \renewcommand*\fb@put@frame[2]{%
96 \relax%
97 \@tempboxa%
98 }
```

Package 50

lwarp-ftnright.sty

127 Ftnright

Pkg ftnright ftnright is ignored.
for HTML output: Discard all options for lwarp-ftnright:
`1 \LWR@ProvidesPackageDrop{ftnright}`

Package 51

lwarp-fullpage.sty

128 Fullpage

Pkg fullpage Emulated.
for HTML output: Discard all options for lwarp-fullpage:
`1 \LWR@ProvidesPackageDrop{fullpage}`

Package 52

lwarp-geometry.sty

129 Geometry

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:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*\geometry[1]{}
3 \renewcommand*\newgeometry[1]{}
4 \renewcommand*\restoregeometry{}
5 \renewcommand*\savegeometry[1]{}
6 \renewcommand*\loadgeometry[1]{}

```

Package 53

lwarp-glossaries.sty

130 Glossaries

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

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
5 \renewcommand*{\@p@glossarysection}[2]{%
6   \glsclearpage
7   \phantomsection
8   \ifdefempty\@glossarysecstar
9   {%
10    \csname\@glossarysec\endcsname{#2}%
11  }%
12  {%
```

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:

```
13    \csname\@glossarysec\endcsname*{#2}%
14    \gls@toc{#1}{\@glossarysec}% Moved after the previous line.
15  }%
16  \@glossaryseclabel
17 }
```

Package 54**lwarp-graphics.sty****131 Graphics**

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

Package 55**lwarp-graphicx.sty****132 Graphicx**

Pkg graphicx graphicx is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 56

lwarp-grffile.sty

133 Grffile

Pkg grffile grffile is partially emulated. 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 version for HTML. No attempt is made to convert input or filename encodings.

⚠ matching PDF and SVG
⚠ encodings

for HTML output:

```
1 \LWR@ProvidesPackageDrop{grffile}

2 \define@key{igraph}{multidot}{}
3 \define@key{igraph}{babel}{}
4 \define@key{igraph}{extendedchars}{}
5 \define@key{igraph}{encoding}{}
6 \define@key{igraph}{inputencoding}{}
7 \define@key{igraph}{filenameencoding}{}
8 \define@key{igraph}{space}{}
9
10 \newcommand*{\grffilesetup}[1]{}

```

Package 57

lwarp-hyperref.sty

134 Hyperref

Pkg hyperref hyperref is emulated during HTML output, and the hyperref package is ignored.

```

for HTML output: 1% \LWR@ProvidesPackageDrop{hyperref}
2\typeout{Using the lwarp html version of package 'hyperref' -- discarding options.}
3\typeout{   Are not using ProvidesPackage, so that other packages}
4\typeout{   do not attempt to patch lwarp's version of 'hyperref'.}
5% \ProvidesPackage{lwarp-#1-#2}
6\DeclareOption*{}
7\ProcessOptions\relax

8\newcommand*\hypersetup}[1]{}
9\newcommand*\hyperbaseurl}[1]{}

```

Insert an image with alt text:

```

10\NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11\LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{}%}
12\endgroup%
13\LWR@ensuredoingapar%
14}
15
16\newcommand{\hyperimage}{%
17\LWR@ensuredoingapar%
18\begingroup\catcode'\_ =12
19\LWR@hyperimageb%
20}

```

Creates an HTML anchor to category.name with the given text.

```

21\NewDocumentCommand{\hyperdef}{m m +m}{%
22\LWR@ensuredoingapar%
23\LWR@subsublabel{#1.#2}%
24#3%
25}

```

Creates an HTML link to URL#category.name with the given text.

```

26\NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27\LWR@htmltag{a href="#1\LWR@hashmark#2.#3"%}

```

```

28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }

```

Creates text as an HTML link to the \LaTeX label.

```

32 \NewDocumentCommand{\LWR@hyperrefc}{0{label} +m}{%
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

```

```

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }

```

Creates an anchor to name with the given text.

```

43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }

```

Creates a link to the anchor created by `hypertarget`, with the given link text.

```

47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref[#{1}]{#2}%
49 }

```

For HTML, `\cleveref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleveref` is used instead.

```

53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

```

```

56 \newcommand{\pdfstringdef}[2]{}

```

```
57 \newcommand{\pdfbookmark}[3] [] {}
58 \newcommand{\currentpdfbookmark}[2] {}
59 \newcommand{\subpdfbookmark}[2] {}
60 \newcommand{\belowpdfbookmark}[2] {}
61 \newcommand{\texorpdfstring}[2] {#2}
```

From hyperref.

```
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%

65 \newcommand{\Acrobatmenu}[2] {}

66 \newcommand*{\TextField}[2] [] {}

67 \newcommand*{\CheckBox}[2] [] {}

68 \newcommand{\ChoiceMenu}[3] [] {}

69 \newcommand*{\PushButton}[2] [] {}

70 \newcommand*{\Submit}[2] [] {}

71 \newcommand*{\Reset}[2] [] {}

72 \newcommand*{\LayoutTextField}[2] {}

73 \newcommand*{\LayoutChoiceField}[2] {}

74 \newcommand*{\LayoutCheckField}[2] {}

75 \newcommand*{\MakeRadioField}[2] {}

76 \newcommand*{\MakeCheckField}[2] {}

77 \newcommand*{\MakeTextField}[2] {}

78 \newcommand*{\MakeChoiceField}[2] {}

79 \newcommand{\MakeFieldButton}[1] {}
```

Package 58**lwarp-hyperxmp.sty****135 Hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

Package 59

lwarp-idxlayout.sty

136 Idxlayout

Pkg idxlayout Emulated.

for HTML output: Discard all options for lwarp-idxlayout:

```
1 \LWR@ProvidesPackageDrop{idxlayout}

2 \newcommand{\LWR@indexprenote}{}
3
4 \renewcommand*{\printindex}
5 {
6 \LWR@startpars
7
8 \LWR@indexprenote
9
10 \LWR@origprintindex
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{} }
15
16 \newcommand{\idxlayout}[1]{}
17 \newcommand*{\indexfont}{}
18 \newcommand*{\indexjustific}{}
19 \newcommand*{\indexsubsdelim}{}
20 \newcommand*{\indexstheadcase}{}

```

Package 60

lwarp-indentfirst.sty

137 Indentfirst

Pkg indentfirst indentfirst is ignored.
Discard all options for lwarp-indentfirst:
for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

Package 61

lwarp-inputenc.sty

138 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}

Package 62

lwarp-keyfloat.sty

139 Keyfloat

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```

2 \AtBeginDocument{
3 \let\KFLT@boxinner\relax
4 \let\endKFLT@boxinner\relax
5
6 \NewEnviron{KFLT@boxinner}
7 {%
8 \LWR@traceinfo{kflt@boxinner}%
9 \LWR@stoppars%
10 \KFLT@frame{\BODY}%
11 \LWR@startpars%
12 \LWR@traceinfo{ended kflt@boxinner}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29
30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}

```

```
33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }

42 }% AtBeginDocument
```

Package 63

lwarp-layout.sty

140 Layout

Pkg layout layout is ignored.
for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

Package 64

lwarp-letterspace.sty

141 Letterspace

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:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][1]{}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

Package 65

lwarp-lettrine.sty

142 Lettrine

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.
 for HTML output: Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `` of class `lettrine`, and the following text is in a `` of class `lettrinetext`. `\lettrine [keys] {letter} {additional text}`

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
9 \newcommand*\DefaultOptionsFile{\relax}
10 \newcommand*\DefaultLoversize{0}
11 \newcommand*\DefaultLraise{0}
12 \newcommand*\DefaultLhang{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*\LettrineTextFont{\scshape}
26
27 \newcommand*\LettrineFontHook{}
28
29 \newcommand*\LettrineFont[1]{\InlineClass{lettrine}{#1}}
30 \newcommand*\LettrineFontEPS[1]{\includegraphics [height=1.5ex]{#1}}
```

Package 66

lwarp-lips.sty

143 Lips

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```
1% \LWR@ProvidesPackageDrop{lips}
2\PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
3\ProvidesPackage{lwarp-lips}
4
5\NewDocumentCommand{\Lips}{}{\textellipsis}
6
7\NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}
8
9\let\lips\Lips
10\let\olips\lips
11
12\DeclareOption*{}
13\DeclareOption{mla}{
14\let\lips\BracketedLips
15}
16\ProcessOptions\relax
17
18\newcommand \LPNobreakList {}
```

Package 67

lwarp-listings.sty

144 Listings

(Based on original 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: `1 \begin{warpHTML}`

```
2 \LWR@ProvidesPackagePass{listings}
```

Patches to embed listings inside pre tags:

```
3 \let\LWR@origlst@Init\lst@Init
4 \let\LWR@origlst@DeInit\lst@DeInit
5
6 \let\LWR@origlsthkEveryPar\lsthk@EveryPar
7
8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{l@l}{#1}{#2}}
```

Done at the start of a listing.

```
9 \renewcommand{\lst@Init}[1]{%
```

First, perform the listings initialization:

```
10 \LWR@traceinfo{\lst@Init}%
11 \renewcommand*{\@capttype}{\lstlisting}%
12 \LWR@origlst@Init{#1}%
13 \LWR@traceinfo{finished origlst@Init}%
14 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```
15 \LWR@traceinfo{About to create verbatim.}%
16 \let\lsthk@EveryPar\relax%
17 \LWR@forcenewpage
18 \LWR@atbeginverbatim{programlisting}%
19
20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
```

```
21 \else%
```

Inline, so open a :

```
22 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }
```

```
25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```
27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so create the closing :

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #1t%
41 \ifx\lst@caption\@empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 % \ifx\lst@label\@empty\else
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
```

```

49     \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50     \lst@ifnolol\else
51         \ifx\lst@@caption\@empty
52             \ifx\lst@caption\@empty
53                 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54                 \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %             \addcontentsline{lol}{\lstlisting}{\lst@name}
57             \fi\fi
58         \fi
59     \else

```

This would have to be modified for lwarp:

```

60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61             \addcontentsline{lol}{\lstlisting}%
62 {\protect\numberline{\thelstlisting}}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64         \fi
65     \fi
66 \fi
67 \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{\lst@caption not empty-}%
69     \lst@ifsubstring #1\lst@captionpos
70     {\begingroup
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 %             \let\@@vskip\vskip
73 %             \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
74 %             \def\lst@vskip{\nobreak\@@vskip\@tempkipa\nobreak}%
75 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %             \ifx #1t\allowbreak \fi
77             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
79         \else

```

New lwarp code to create a title:

```

80 %             \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%

```

```

82 \begin{BlockClass}{\lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp
85         \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87         \ifx\lst@label\empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 %             \ifx #1b\allowbreak \fi
95             \endgroup}{}%
96     \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98 \else
99 \LWR@traceinfo{INLINE}%
100 \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}{%
104     \let\lst@PlaceNumber\empty
105     \lstKV@SwitchCases{#1}%
106     {none&\\%
107     left&\def\lst@PlaceNumber{%
108 % \llap{
109 \LWR@orignormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115         \kern\VerbatimHTMLWidth \kern\lst@numbersep
116         \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
118 \end{warpHTML}

```

Package 68

lwarp-longtable.sty

145 Longtable

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \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 relevant 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` `* [⟨horizontal⟩] {⟨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.

```
2 \newenvironment{longtable*}[2][ ]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
```

```

8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][\LWR@floatbegin{table}]%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{LTcaption}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{LTcaption}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and /*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{LTleft}
34 \newlength{LTright}
35 \newlength{LTpre}
36 \newlength{LTpost}
37 \newlength{LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

Package 69**lwarp-lscape.sty****146 Lscape**

Pkg lscape lscape is nullified.
for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

Package 70**lwarp-ltcaption.sty****147 Ltcaption**

Pkg ltcaption ltcaption is emulated during HTML output, and the ltcaption package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by lwarp.

longtable* is already defined by lwarp-longtable.

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginfalse}{}

```

Package 71

lwarp-marginfit.sty

148 Marginfit

Pkg marginfit Emulated.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

Package 72

lwarp-marginfix.sty

149 Marginfix

Pkg marginfix Emulated.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1] [] {}
10 \newcommand*{\unblockmargin}[1] [] {}
11 \newcommand*{\marginphantom}[2] [] {}
```

Package 73

lwarp-marginnote.sty

150 Marginnote

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```

1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

Package 74

lwarp-mcaption.sty

151 Mcaption

Pkg mcaption mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```

Package 75

lwarp-mdframed.sty

152 Mdfamed

`Pkg mdframed` mdframed is loaded with options forced to `framemethod=none`.

for HTML output:

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackageDrop{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.

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 `` of class `mdframedtitle`. Subtitles are in a `` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

amsthm must be loaded before mdframed

```
3 \LWR@origRequirePackage{amsthm}
```

Do not require Tikz or pstricks:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
5 \mdfsetup{
6 startcode={\LWR@mdframedstart\LWR@origraggedright},
7 endcode={\LWR@mdframedend},
8 startinnercode={\LWR@startpars\LWR@origraggedright},
9 endinnercode={\LWR@stoppars},
10 }
```

Given the mdframed key, print the color.

```
11 \newcommand*\LWR@mdfprintcolor}[1]{%
12 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
13 \#\LWR@tempcolor
14 }
```

Given the mdframed key, print the length.

```
15 \newcommand*\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired class.

```
18 \newcommand*\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
19 \LWR@stoppars%
```

Below, print HTML pt units:

```
20 \uselengthunit{PT}%
```

Open a `<div>` and with custom class and custom style:

```
21 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline
```

Convert and print the background color:

```
23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
24 border: \LWR@mdfprintlength{linewidth} solid
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
27 \ifbool{mdf@shadow}{%
28   box-shadow:
29     \LWR@mdfprintlength{shadowsize}
30     \LWR@mdfprintlength{shadowsize}
31     \LWR@mdfprintlength{shadowsize}
32     \LWR@mdfprintcolor{shadowcolor} ;
33 }
34 {box-shadow: none ;}
35 \LWR@orignewline

36 "}
37 % \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:

```
38 \LetLtxMacro{\hspace}{\LWR@origspace}%
39 \LetLtxMacro\rule\LWR@origrule%
40 \LetLtxMacro\makebox\LWR@origmakebox%
41 }
```

Actions after an mdframe ends.

After closing the <div>, globally restore to the default environment type:

```
42 \newcommand*{\LWR@mdframedend}{
```

Close the custom <div>:

```
43 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
44 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
45 \LWR@startpars%
46 }
```

Encapsulation of the original which places the title inside a `` of class `mdframedtitle`:

```
47 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd\mdfframedtitleenv[1]{%
52 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
53 \uselengthunit{PT}%
```

Open a `` with a custom class and custom style:

```
54 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline
```

Convert and print the title background color:

```
56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline
```

Convert and print the title rule:

```
59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
61   \LWR@mdfprintlength{frametitlerulewidth}
62   solid
63   \LWR@mdfprintcolor{frametitlerulecolor}
64   ; \LWR@orignewline
65 }{}%
```

The title's top border radius is adjusted for the line width:

```
66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
```

```

69 \rndprintlength{\LWR@titleroundcorner}
70 \rndprintlength{\LWR@titleroundcorner}
71 Opt Opt
72 \LWR@orignewline

```

Finish the custom style and the opening span tag:

```

73 " \LWR@orignewline
74 }% span

```

Restrict paragraph tags inside a span:

```

75 \begin{LWR@nestspan}%

```

Print the title inside the span:

```

76 #1%

```

Close the span and unnest the paragraph tag restriction:

```

77 \LWR@htmltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }

```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `` of class `mdframedsubtitle`:

```

81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {% the following empty line is required
83

```

Special handling for `mdframed`: Subtitles have `\pars` around them, so temporarily disable them here.

```

84 \let\par\LWR@origpar%

```

Open a `` with a custom class and custom style:

```

85 \LWR@htmltagc{span class="mdframed#1title"
86 style=" \LWR@orignewline

```

Convert and print the background color:

```

87 background:
88 \LWR@mdfprintcolor{#1titlebackgroundcolor}
89 ; \LWR@orignewline

```

Convert and print the above line:

```

90 \ifbool{mdf@#1titleaboveline}{%
91   border-top:
92   \LWR@mdfprintlength{#1titleabovelinewidth}
93   solid
94   \LWR@mdfprintcolor{#1titleabovelinecolor}
95   ; \LWR@orignewline
96 }{}%
```

Convert and print the below line:

```

97 \ifbool{mdf@#1titlebelowline}{%
98   border-bottom:
99   \LWR@mdfprintlength{#1titlebelowlinewidth}
100  solid
101  \LWR@mdfprintcolor{#1titlebelowlinecolor}
102  ; \LWR@orignewline
103 }{}%
```

Finish the custom style and the opening span tag:

```
104 "}% span
```

Restrict paragraph tags inside a span:

```
105 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

106 \IfNoValueTF{#2}
107 {\csuse{LWR@origmdf#1title}{#3}}%
108 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

109 \LWR@htmltagc{/span}% the following empty line is required
110 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
111
112 }
```

```

113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle
```

```

119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

Stores the environment of the frame about to be created:

```
125 \newcommand*\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

126 \renewrobustcmd*\newmdenv[2] [] {%
127 \newenvironment{#2}%
128 {%
129 \mdfsetup{#1}%
130 \renewcommand*\LWR@mdthisenv}{md#2}%
131 \begin{mdframed}%
132 }
133 {\end{mdframed}}%
134 }

```

Modified from the original to remember the environment.

```

135 \renewrobustcmd*\surroundwithmdframed[2] [] {%
136 \BeforeBeginEnvironment{#2}{%
137 \renewcommand*\LWR@mdthisenv}{md#2}%
138 \begin{mdframed}[#1]}%
139 \AfterEndEnvironment{#2}{\end{mdframed}}%
140 }

```

[*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

141 \DeclareDocumentCommand{\mdtheorem}{ 0{} m o m o }%
142 {\ifcsdef{#2}%
143   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144   {%
145     \IfNoValueTF {#3}%
146     {%#3 not given -- number relationship
147       \IfNoValueTF {#5}%
148       {%#3+#5 not given
149         \@definecounter{#2}%
150         \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151         \newenvironment{#2}[1] [] {%
152           \refstepcounter{#2}%
153           \ifstrempy{##1}%

```

```

154     {\let\@temptitle\relax}%
155     {%
156         \def\@temptitle{\mdf@theoremseparator%
157             \mdf@theoremspace%
158             \mdf@theoremtitlefont%
159             ##1}%
160         \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
161     }%
162     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
163         \@temptitle}]]%
164     {\end{mdframed}}%
165     \newenvironment{#2*}[1] [] {%
166         \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
167         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
168         {\end{mdframed}}%
169     }%
170     {%#5 given -- reset counter
171     \@definecounter{#2}\@newctr{#2}[#5]%
172     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
173     \expandafter\xdef\csname the#2\endcsname{%
174         \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
175         \@thmcounter{#2}}%
176     \newenvironment{#2}[1] [] {%
177         \refstepcounter{#2}%
178         \ifstrempy{##1}%
179         {\let\@temptitle\relax}%
180         {%
181             \def\@temptitle{\mdf@theoremseparator%
182                 \mdf@theoremspace%
183                 \mdf@theoremtitlefont%
184                 ##1}%
185             \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
186         }
187         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
188             \@temptitle}]]%
189         {\end{mdframed}}%
190     \newenvironment{#2*}[1] [] {%
191         \ifstrempy{##1}%
192         {\let\@temptitle\relax}%
193         {%
194             \def\@temptitle{\mdf@theoremseparator%
195                 \mdf@theoremspace%
196                 \mdf@theoremtitlefont%
197                 ##1}%
198             \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
199         }%
200         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
201         {\end{mdframed}}%
202     }%
203 }%

```

```

204   {%#3 given -- number relationship
205   \global\@namedef{the#2}{\@nameuse{the#3}}%
206   \newenvironment{#2}[1][]{%
207     \refstepcounter{#3}%
208     \ifstrempy{##1}%
209     {\let\@temptitle\relax}%
210     {%
211       \def\@temptitle{\mdf@theoremseparator%
212         \mdf@theoremspace%
213         \mdf@theoremtitlefont%
214         ##1}%
215       \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
216     }
217   \begin{mdframed}[#1,frametitle={\strut#4 \csname the#2\endcsname%
218     \@temptitle}]]%
219     {\end{mdframed}}%
220   \newenvironment{#2*}[1][]{%
221     \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
222     \begin{mdframed}[#1,frametitle={\strut#4@temptitle}]]%
223     {\end{mdframed}}%
224   }%
225   \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
226   \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
227 }%
228 }

```

[*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
230   \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
231     {\newtheorem{#2}{#4}}{%
232       \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
233         \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
234           }%
235       }%
236     \renewcommand*\LWR@mdthisenv}{md#2}}%
237   \begin{mdframed}[#1]}%
238   \AfterEndEnvironment{#2}{%
239     \end{mdframed}}%
240 }

```

Package 76

lwarp-metalogo.sty

153 Metalogo

Pkg metalogo metalogo is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{metalogo}

2 \newcommand\setlogokern[2]{}
3 \newcommand\setlogodrop[2][XeTeX]{}
4 \newcommand\setLaTeXa[1]{}
5 \newcommand\setLaTeXee[1]{}
6 \newcommand\seteverylogo[1]{}
7 \newcommand\everylogo[1]{}

```

Package 77

lwarp-microtype.sty

154 Microtype

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:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

Package 78

lwarp-moreverb.sty

155 Moreverb

Pkg moreverb moreverb is supported with some patches.

```

for HTML output: 1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatimtab}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatimtab}{%
8 \LWR@afterendverbatim%
9 }
10
11
12 \LetLtxMacro\LWRMV@orig@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2][ ]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#1]{#2}%
18 \LWR@afterendverbatim%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%

```

```
37% \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }
40
41 \renewcommand{\listinglabel}[1]{\InlineClass{listinglabel}{\the#1}}
42
43 \def\thelisting@line{%
44 \@tempcnta=\listing@line%
45 \divide\@tempcnta\listing@step \multiply\@tempcnta\listing@step%
46 \ifnum\listing@line=\@ne%
47 \listinglabel\listing@line%
48 \else%
49 \ifnum\@tempcnta=\listing@line%
50 \listinglabel\listing@line%
51 \else%
52 \InlineClass{listinglabel}{}%
53 \fi%
54 \fi}
55
56 \LetLtxMacro\LWRMV@@listinginput\@listinginput
57
58 \renewcommand{\@listinginput}[3][]{
59 \LWR@forcenewpage
60 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
61 \LWRMV@@listinginput[#1]{#2}{#3}
62 \LWR@afterendverbatim%
63 }
64
65
66 \renewenvironment*{boxedverbatim}
67 {
68 \LWR@forcenewpage
69 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
70 \verbatim%
71 }
72 {
73 \endverbatim%
74 \LWR@afterendverbatim%
75 }
76
77
78 \end{warpHTML}
```

Package 79**lwarp-mparhack.sty****156 Mparhack**

Pkg mparhack Emulated.

for HTML output: Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

Package 80

lwarp-multicol.sty

157 Multicol

Pkg multicol multicol is emulated during HTML output, and the multicol package is ignored.

for HTML output: 1 \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.

```
2 \begin{warpHTML}
```

```
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML div class to contain everything:

```
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}
```

Optional HTML div class for the heading:

```
7 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}
```

When done with the environment, close the div:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}
10 \newcommand*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
```

```
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}
```

Package 81

lwarp-multirow.sty

158 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{multirow}`

Package 82

lwarp-nameref.sty

159 Nameref

Pkg nameref nameref is emulated by lwarp.

for HTML output: Discard all options for lwarp-nameref:

```
1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{   Are not using ProvidesPackage, so that other packages}
3 \typeout{   do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

Package 83

lwarp-needspace.sty

160 Needspace

Pkg needspace needspace is not used during HTML conversion.

for HTML output: Discard all options for lwarp-needspace:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}

```

Package 84

lwarp-newclude.sty

161 Newclude

Pkg newclude Error if newclude is loaded after lwarp.

Discard all options for lwarp-newclude:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{newclude}
2 \LWR@loadbefore{newclude}

```

Package 85

lwarp-newunicodechar.sty

162 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.
Discard all options for lwarp-newunicodechar:
for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}
2 \LWR@loadbefore{newunicodechar}

Package 86

lwarp-nextpage.sty

163 Nextpage

Pkg nextpage nextpage is nullified.
for HTML output: Discard all options for lwarp-nextpage.
1 \LWR@ProvidesPackageDrop{nextpage}
2 \newcommand{\cleartoevenpage}[1] [] {}
3 \newcommand{\movetoevenpage}[1] [] {}
4 \newcommand{\cleartooddpage}[1] [] {}
5 \newcommand{\movetooddpage}[1] [] {}

Package 87**lwarp-nonumonpart.sty****164 Nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

Package 88**lwarp-nopageno.sty****165 Nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

Package 89

lwarp-nowidow.sty

166 Nowidow

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}
2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}
4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

Package 90

lwarp-ntheorem.sty

167 Ntheorem

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

`\pkg ntheorem` ntheorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

 **Font control** This conversion is not total. Font control is via CSS, and the custom \TeX 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.

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:

```
1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{
```

```

8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{theorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \@ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
28     \theorem@style{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30     }%
31   {
32     \theorem@style{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}% new
34   }
35 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

36
37 \gdef\@xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40     \expandafter\@ifundefined{c@#1}%
41       {\@definecounter{#1}}{}%
42     \@newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45       {\noexpand\noexpand\the\theoremnumbering\endcsname{#1}}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname

```

```

47     {\csname setparms@#1\endcsname
48       \@thm{#1}{#1}{#2}
49       }%
50   \global\@namedef{end#1}{\@endtheorem}
51   \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \@thm{#1}{#1}{#2}
65         }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
68     \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72   \@ifundefined{c@#2}{\@nocounterr{#2}}%
73   {\ifthm@tempif
74     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75     \global\@namedef{the#1}{\@nameuse{the#2}}%
76     \expandafter\protected@xdef\csname num@addtheoremeline#1\endcsname{%
77       \noexpand\@num@addtheoremeline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremeline#1\endcsname{%
79       \noexpand\@nonum@addtheoremeline{#1}{#3}}%
80     \theoremkeyword{#3}%
81     \expandafter\protected@xdef\csname #1Keyword\endcsname
82       {\the\theoremkeyword}%
83     \expandafter\gdef\csname mkheader@#1\endcsname
84       {\csname setparms@#1\endcsname
85         \@thm{#1}{#2}{#3}
86         }%
87     \global\@namedef{end#1}{\@endtheorem}
88     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
89     \fi}
90 }

```

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%

```

```

92 \addtocounter{LWR@thisfloat}{1}%
93 \LWR@stoppars%
94 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
95 \LWR@startpars%
96 }

```

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>.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}%
99     {\expandafter\gdef\csname th@#1\endcsname{%
100       \def\@begintheorem####1####2{%
101 \LWR@forcenewpage% new
102 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
103 \LWR@inctheorem% new
104 #2}%
105       \def\@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108 \LWR@inctheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hskip\labelsep \theorem@headerfont
117     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 % \rlap{\vbox{\hbox{
124   \hskip\labelsep \theorem@headerfont
125   \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 %   }\hbox{\strut}}
127   ]}%
128   {\item[
129 % \rlap{\vbox{\hbox{
130   \hskip\labelsep \theorem@headerfont
131   \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline

```

```

132% } \hbox{\strut}}
133   ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hskip\labelsep
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139   {\item[\hskip\labelsep
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145%   \rlap{\vbox{\hbox{
146         \hskip\labelsep \theorem@headerfont
147         \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148%   } \hbox{\strut}}
149   ]}%
150   {\item[
151%   \rlap{\vbox{\hbox{
152         \hskip\labelsep \theorem@headerfont
153         \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154%   } \hbox{\strut}}
155   ]}
156
157 \renewtheoremstyle{margin}%
158   {\item[\hskip\labelsep\theorem@headerfont
159     \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160   ]}%
161   {\item[\hskip\labelsep\theorem@headerfont
162     \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163   ]}
164
165 \renewtheoremstyle{marginbreak}%
166   {\item[\hskip\labelsep\theorem@headerfont
167     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168   ]}%
169   {\item[\hskip\labelsep\theorem@headerfont
170     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171   ]}
172
173 \renewtheoremstyle{nonumberplain}%
174   {\item[\theorem@headerfont\hskip\labelsep
175     \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
176   {\item[\theorem@headerfont\hskip\labelsep
177     \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}
178
179 \renewtheoremstyle{nonumberbreak}%
180   {\item[
181%   \rlap{\vbox{\hbox{

```

```

182 \hskip\labelsep \theorem@headerfont
183 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184% } \hbox{\strut}}
185 ]}%
186 {\item[
187% \rlap{\vbox{\hbox{
188 \hskip\labelsep \theorem@headerfont
189 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190% } \hbox{\strut}}
191 ]}
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196 \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%
200 {\item[\theorem@headerfont \hskip\labelsep\relax
201 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@theoremamsthm}{}{%
203% upright text via CSS
204 \newtheoremstyle{plainupright}%
205 {\item[\hskip\labelsep \theorem@headerfont
206 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
207 {\item[\hskip\labelsep \theorem@headerfont
208 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
209
210% upright text and small caps header via CSS
211 \newtheoremstyle{nonumberplainuprightsc}%
212 {\item[\theorem@headerfont\hskip\labelsep
213 \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
214 {\item[\theorem@headerfont\hskip \labelsep
215 \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new CSS:

```

216 \theoremstyle{plainupright}
217 \theorembodyfont{\upshape}
218 \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
219 \renewtheorem{Example}{Example}
220 \renewtheorem{example}{Example}
221 \renewtheorem{Beispiel}{Beispiel}
222 \renewtheorem{beispiel}{Beispiel}
223 \renewtheorem{Bemerkung}{Bemerkung}

```

```

224 \renewtheorem{bemerkung}{Bemerkung}
225 \renewtheorem{Anmerkung}{Anmerkung}
226 \renewtheorem{anmerkung}{Anmerkung}
227 \renewtheorem{Remark}{Remark}
228 \renewtheorem{remark}{Remark}
229 \renewtheorem{Definition}{Definition}
230 \renewtheorem{definition}{Definition}
231
232 \theoremstyle{nonumberplainuprightsc}
233 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
234 \renewtheorem{Proof}{Proof}
235 \renewtheorem{proof}{Proof}
236 \renewtheorem{Beweis}{Beweis}
237 \renewtheorem{beweis}{Beweis}
238 \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
239
240 \theoremsymbol{}
241 }% not amsthm

```

Only if the amsthm option was given:

```

242 \ifbool{LWR@theoremamsthm}{
243
244 \gdef\th@plain{%
245 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246 \def\@begintheorem##1##2{%
247 \LWR@forcenewpage% new
248 \BlockClass{theorembodyplain}% new
249 \LWR@inctheorem% new
250 \item[\hskip\labelsep
251 % \theorem@headerfont
252 \InlineClass{theoremheaderplain}{##1\ ##2.}
253 ]}%
254 \def\@opargbegintheorem##1##2##3{%
255 \LWR@forcenewpage% new
256 \BlockClass{theorembodyplain}% new
257 \LWR@inctheorem% new
258 \item[\hskip\labelsep
259 % \theorem@headerfont
260 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
261 ]}}
262
263 \gdef\th@nonumberplain{%
264 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265 \def\@begintheorem##1##2{%
266 \LWR@forcenewpage% new
267 \BlockClass{theorembodyplain}% new
268 \LWR@inctheorem% new
269 \item[\hskip\labelsep
270 % \theorem@headerfont

```

```

271 \InlineClass{theoremheaderplain}{##1.}
272   }%
273 \def\@opargbegintheorem##1##2##3{%
274 \LWR@forcenewpage% new
275     \BlockClass{theorembodyplain}% new
276     \LWR@inctheorem% new
277     \item[\hskip\labelsep
278 % \theorem@headerfont
279 \InlineClass{theoremheaderplain}{##1\ (##3).}
280   ]}}
281
282 \gdef\th@definition{%
283 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284 \def\@begintheorem##1##2{%
285 \LWR@forcenewpage% new
286     \BlockClass{theorembodydefinition}% new
287     \LWR@inctheorem% new
288     \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291   ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294     \BlockClass{theorembodydefinition}% new
295     \LWR@inctheorem% new
296     \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299   ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305     \BlockClass{theorembodydefinition}% new
306     \LWR@inctheorem% new
307     \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310   ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313     \BlockClass{theorembodydefinition}% new
314     \LWR@inctheorem% new
315     \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318   ]}}
319
320 \gdef\th@remark{%

```

```

321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324     \BlockClass{theorembodyremark}% new
325     \LWR@inctheorem% new
326     \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329     ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332     \BlockClass{theorembodyremark}% new
333     \LWR@inctheorem% new
334     \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337     ]}}
338
339 \gdef\th@nonumberremark{%
340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343     \BlockClass{theorembodyremark}% new
344     \LWR@inctheorem% new
345     \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348     ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351     \BlockClass{theorembodyremark}% new
352     \LWR@inctheorem% new
353     \item[\hskip\labelsep
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (##3).}
356     ]}}
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362     \BlockClass{theorembodyproof}% new
363     \LWR@inctheorem% new
364     \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367     ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370     \BlockClass{theorembodyroof}% new

```

```

371     \LWR@inctheorem% new
372     \item[\hskip\labelsep
373 % \theorem@headerfont
374 \InlineClass{theoremheaderproof}{##1\ (##3).}
375     ]}}
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381     \newcounter{currproofctr}%
382     \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{
390     \th@proof
391     \def\theorem@headerfont{\itshape}%
392     \normalfont
393     \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
394     \@thm{proof}{proof}{#1}
395 }%
396 {\@endtheorem}
397
398 }{}% amsthm option

```

Patched for CSS:

```

399 \let\LWR@origendtheorem\@endtheorem
400 \renewcommand{\@endtheorem}{%
401 \ifbool{LWR@theoremmarks}{%
402     \ifsetendmark%
403     \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404     \setendmarkfalse%
405     \fi%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```
\thm@thmline {<1: printed type>} {<2: #>} {<3: optional>} {<4: page>}
```

```
412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }
```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```
419 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```
420 \def\listtheorems#1{
421 \LWR@htmlclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@htmlclassend{nav}{lothm}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\quad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
435 }
```

```
\thref {<label>}
```

```
436 \newcommand*{\thref}[1]{\cref{#1}}
```

Package 91

lwarp-pagenote.sty

168 Pagenote

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

for HTML output:

```
1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}
```

Package 92

lwarp-paralist.sty

169 Paralist

Pkg paralist paralist is supported with minor changes.

for HTML output:

```
1 \LWR@ProvidesPackagePass{paralist}

2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtEndEnvironment{compactitem}{\LWR@itemizeend}
4 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
5 \AtEndEnvironment{compactenum}{\LWR@enumerateend}
6 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
7 \AtEndEnvironment{compactdesc}{\LWR@descriptionend}
8 \def\paradescriptionlabel#1{\normalfont\textbf{#1}}
```

Package 93**lwarp-parskip.sty****170 Parskip**

Pkg parskip parskip is ignored.
for HTML output: Discard all options for lwarp-parskip.
`1 \LWR@ProvidesPackageDrop{parskip}`

Package 94**lwarp-pdfscape.sty****171 Pdfscape**

Pkg pdfscape Emulated.
for HTML output: Discard all options for lwarp-pdfscape:
`1 \LWR@ProvidesPackageDrop{pdfscape}`

Package 95

lwarp-pdftsync.sty

172 Pdftsync

Pkg pdftsync Emulated.

for HTML output: Discard all options for lwarp-pdftsync:

```
1 \LWR@ProvidesPackageDrop{pdftsync}

2 \let\pdftsync\relax
3 \let\pdftsyncstart\relax
4 \let\pdftsyncstop\relax
```

Package 96

lwarp-placeins.sty

173 Placeins

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{placeins}

2 \newcommand*{\FloatBarrier}{}
```

Package 97**lwarp-prelim2e.sty****174 Prelim2e**

Pkg `prelim2e` Emulated.

for HTML output: Discard all options for `lwarp-prelim2e`:

```
1 \LWR@ProvidesPackageDrop{prelim2e}
```

```
2 \newcommand{\PrelimText}{}
```

```
3 \newcommand{\PrelimTextStyle}{}
```

```
4 \newcommand{\PrelimWords}{}
```

Package 98

lwarp-ragged2e.sty

175 Ragged2e

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

```

for HTML output: 1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{}
6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}

```

Package 99

lwarp-realscripts.sty

176 Realscripts

Pkg realscripts realscripts is emulated. See lwarp.css for the `` of class `supsubscript`.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\LWR@origtextsuperscript\textsuperscript
3 \let\LWR@origtextsubscript\textsubscript
4
5 \RenewDocumentCommand{\textsuperscript}{s}{\LWR@origtextsuperscript}
6 \RenewDocumentCommand{\textsubscript}{s}{\LWR@origtextsubscript}
7
8 \let\realsuperscript\textsuperscript
9 \let\realsubscript\textsubscript
10
11 \let\fakesuperscript\textsuperscript
12 \let\fakesubscript\textsubscript
13
14 \newlength{\subsupersep}
15
16 \newcommand*{\LWR@realscriptsalign}{}
17
18 \newcommand*{\LWR@setrealscriptsalign}[1]{%
19 \renewcommand*{\LWR@realscriptsalign}{}%
20 \ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
21 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
22 }
23
24 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
25 \LWR@setrealscriptsalign{#2}%
26 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
27 \textsuperscript{#4}\textsubscript{#3}%
28 }%
29 }
30
31 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
32 \LWR@setrealscriptsalign{#2}%
33 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
34 \textsubscript{#4}\textsuperscript{#3}%
35 }%
36 }

```

Package 100

lwarp-reysize.sty

177 Relsize

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 temporarily restored, and so should work as expected.

for HTML output:

```

1 \LWR@ProvidesPackagePass{relsize}

2 \let\LWR@origrelsize\relsize
3 \LetLtxMacro\LWR@origlarger\larger
4 \LetLtxMacro\LWR@origsmaller\smaller
5 \let\LWR@relscale\relscale
6 \LetLtxMacro\LWR@origtextlarger\textlarger
7 \LetLtxMacro\LWR@origtextsmaller\textsmaller
8 \let\LWR@textscale\textscale
9
10 \renewcommand*{\LWR@restoreorigrelsize}{%
11 \let\relsize\LWR@origrelsize%
12 \LetLtxMacro\larger\LWR@origlarger%
13 \LetLtxMacro\smaller\LWR@origsmaller%
14 \let\relscale\LWR@relscale%
15 \LetLtxMacro\textlarger\LWR@origtextlarger%
16 \LetLtxMacro\textsmaller\LWR@origtextsmaller%
17 \let\textscale\LWR@textscale%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{%
23 \renewcommand*{\larger}[1][1]{%
24 \renewcommand*{\smaller}[1][1]{%
25 \renewcommand*{\relscale}[1]{%
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{\textlarger}{#2}%

```

```
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsize-temp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsize-temp}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsize-temp}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relsize-temp}\%]{textscale}{#2}%
40 }
```

Package 101

lwarp-romanbar.sty

178 Romanbar

Pkg romanbar romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

```
for HTML output: 1 \LWR@ProvidesPackagePass{romanbar}

2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3 \InlineClass[%
4   text-decoration: overline underline ;
5 ]{romanbar}{#1}%
6 }
```

Package 102

lwarp-romanbarpagenumber.sty

179 Romanbarpagenumber

Pkg romanbarpagenumber romanbarpagenumber is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{romanbarpagenumber}
```

Package 103

lwarp-rotating.sty

180 Rotating

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{rotating}

2 \LetLtxMacro\sidewaystable\table
3 \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
```

Package 104

lwarp-rotfloat.sty

181 Rotfloat

Pkg rotfloat rotfloat is emulated during HTML output, and the rotfloat package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}

```

`\newfloat` $\langle 1: type \rangle \langle 2: placement \rangle \langle 3: ext \rangle [\langle 4: within \rangle]$

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.

```

15 \cslet{listof#1s}\relax
16 \cslet{listof#1es}\relax
17 }

```

Package 105

lwarp-savetrees.sty

182 Savetrees

Pkg savetrees Emulated.

for HTML output: Discard all options for lwarp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

Package 106

lwarp-scalefnt.sty

183 Scalefnt

Pkg scalefnt scalefnt is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scalefnt}

```
2 \DeclareRobustCommand\scalefont[1]{} 
```

Package 107

lwarp-setspace.sty

184 Setspace

Pkg setspace setspace is not used during HTML conversion.

Discard all options for lwarp-setspace:

```

for HTML output: 1 \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}
```

Package 108

lwarp-shadow.sty

185 Shadow

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```

Package 109

lwarp-showidx.sty

186 Showidx

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 110

lwarp-showkeys.sty

187 Showkeys

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for lwarp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```

Package 111

lwarp-sidecap.sty

188 Sidecap

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \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.

```
2 \NewDocumentEnvironment{SCTable}{soo}
3 {\IfValueTF{#3}{\table[#3]{}{\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCTable*} {\SCTable*}
8 \cs_new_eq:cN {endSCTable*} \endSCTable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]{}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}
```

Package 112

lwarp-sidenotes.sty

189 Sidenotes

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}
15    {\def\@sidenotes@sidecaption@tof{#2}}
16    \IfNoValueOrEmptyTF{#3}
17    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18    {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19  }
20  \LWR@startpars
21 }
```

Borrowed from the lwarp version of keyfloat:

```
22 \NewDocumentEnvironment{KFLT sidenotes@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
```

```
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36   {\begin{KFLTsidenotes@marginfloat}{figure}}
37   {\end{KFLTsidenotes@marginfloat}}
38
39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLTsidenotes@marginfloat}{table}}
41   {\end{KFLTsidenotes@marginfloat}}
```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the $\LaTeX 2_{\epsilon}$ `article.cls` source:

```
42 \renewenvironment{figure*}
43     {\@dblfloat{figure}}
44     {\end@dblfloat}
45
46 \renewenvironment{table*}
47     {\@dblfloat{table}}
48     {\end@dblfloat}
```

Package 113

lwarp-siunitx.sty

190 Siunitx

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.

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

for HTML output:

```
1 \LWR@ProvidesPackagePass{siunitx}

2 \AtBeginDocument{% in case textcomp was not loaded
3 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
4 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
5 \DeclareSIUnit\elementarycharge{\textit{e}}
6 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
7 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
8 }% AtBeginDocument
9
```

Package 114

lwarp-soul.sty

191 Soul

(Based on original code by MELCHIOR FRANZ.)

pkg soul Emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soul}
2 \RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
3 \newcommand*\LWR@soululcolor{}
4
5 \newcommand*\LWR@soulstcolor{}
6
7 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*\LWR@soulhlcolor}{LWR@soulhlcolordefault}
9 \newcommand*\LWR@soulhlcolor{}
```

Basic markup with CSS:

```
10 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
11 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Add colors if not empty:

```
12 \newcommand{\LWR@soulcolor}[4]{%
13 \ifcempty{#2}%
14 {%
15 \InlineClass{#3}{#1}}%
16 {%
17 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
18 \InlineClass[#4: \#\LWR@tempcolor]{#3}{#1}%
19 }%
20 }
21
22 \newcommand{\ul}[1]{%
23 \LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
24 }
25
26 \newcommand{\st}[1]{
```

```

27 \LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
28 }
29
30 \newcommand{\hl}[1]{
31 \LWR@soulcolor{#1}{LWR@soulhlcolor}{highlight}{background-color}%
32 }

```

Nullified:

```

33 \newcommand*\soulaccent}[1]{}
34 \newcommand*\soulregister}[2]{}
35 \newcommand*\sloppyword}[1]{#1}
36 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
37 \newcommand*\resetso{}
38 \newcommand*\capsdef}[5]{}
39 \newcommand*\capsreset{}
40 \newcommand*\capssave}[1]{}
41 \newcommand*\capselect}[1]{}
42 \newcommand*\setul}[2]{}
43 \newcommand*\resetul{}
44 \newcommand*\setuldepth}[1]{}
45 \newcommand*\setuloverlap}[1]{}

```

Set colors:

```

46 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
47 \newcommand*\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
48 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

49 \let\textso\so
50 \let\textul\ul
51 \let\texthl\hl
52 \let\textcaps\caps

```

Package 115

lwarp-subfig.sty

192 Subfig

(Based on original 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:

```
1 \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.

```
2 \long\def\sf@@@subfloat#1[#2] [#3] #4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \@ifundefined{FBsc@max}{-}{%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7     \@tempcnta=\@ne
8     \if@minipage
9       \@tempcnta=\z@
10    \else\ifdim \lastskip=\z@ \else
11      \@tempcnta=\tw@
12    \fi\fi
13    \ifmaincaptiontop
14      \sf@top=\sf@nearskip
15      \sf@bottom=\sf@farskip
16    \else
17      \sf@top=\sf@farskip
18      \sf@bottom=\sf@nearskip
19    \fi
20    \leavevmode
21    \setbox\@tempboxa \hbox{#4}%
```

```

22   \@tempdima=\wd\@tempboxa
23   \@ifundefined{FBsc@max}{}%
24       {\global\advance\Xhsize-\wd\@tempboxa
25         \dimen@=\ht\@tempboxa
26         \advance\dimen@\dp\@tempboxa
27         \ifdim\dimen@>\FBso@max
28           \global\FBso@max\dimen@
29         \fi}%
30   \vtop\bgroup
31   \vbox\bgroup
32   \ifcase\@tempcnta
33     \@minipagefalse
34   \or
35     \vskip\sf@top
36   \or
37     \ifdim \lastskip=\z@ \else
38       \@tempkipb\sf@top\relax\@xaddvskip
39     \fi
40   \fi
41   \sf@ifpositiontop{%
42     \ifx \@empty#3\relax \else
43       \sf@subcaption{#1}{#2}{#3}%
44       \vskip\sf@capskip
45       \vskip\sf@captopadj
46     \fi\egroup
47     \hrule widthOpt heightOpt depthOpt
48     \LWR@startpars% new
49% \box\@tempboxa
50     #4
51     \LWR@stoppars% new
52   }{%
53     \LWR@startpars% new
54     \@ifundefined{FBsc@max}{%
55       {
56% \box\@tempboxa
57       #4
58     }%
59     {\ifx\FBsuboheight\relax
60%       \box\@tempboxa
61       #4
62     \else
63%       \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64       #4
65     \fi}%
66   \LWR@stoppars% new
67   \egroup
68   \ifx \@empty#3\relax \else
69     \vskip\sf@capskip
70     \hrule widthOpt heightOpt depthOpt
71     \sf@subcaption{#1}{#2}{#3}%

```

```

72     \fi
73   }%
74   \vskip\sf@bottom
75 \egroup
76 \@ifundefined{FBsc@max}{}%
77   {\addtocounter{FRobj}{-1}}%
78   \ifnum\c@FRobj=0\else
79     \subfloatrowsep
80   \fi}%
81 \ifmaincaptiontop\else
82   \global\advance\@nameuse{c@\@capttype}\m@ne
83 \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

\sf@subcaption {<1 type>} {<2 lof entry>} {<3 caption>}

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91   \bgroup
92     \let\label=\@gobble
93     \let\protect=\string
94     \def\@subcaplabel{%
95       \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96     \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\the\value{#1}}%
97   \egroup
98 \fi
99 \bgroup
100 \ifx \relax#3\relax
101   \let\captionlabelsep=\relax
102 \fi
103 %   \setbox0\vbox{%
104 %     \hb@xt@\the\@tempdima{%
105 %
106 % %       \hss
107 % %       \parbox[t]{\the\@tempdima}{%
108 % %         \caption@make
109 % %           {\@nameuse{sub\@capttype name}}}%
110 % %           {\@nameuse{thesub\@capttype}}}%
111 % %         {#3}
112 % %   }%
113 % %     \hss
114 %   }
115 % }%
116 \@ifundefined{FBsc@max}%
117 %   {\box0}%
118 %   {

```

```

119% \parbox[t]{\the\@tempdima}{%
120\LWR@traceinfo{sfsubcap B1}% new
121     \LWR@htmlblocktag{figcaption}% new
122     \caption@make
123         {\@nameuse{sub\@capttype name}}}%
124         {\@nameuse{thesub\@capttype}}}%
125         {#3}
126     \LWR@htmlblocktag{/figcaption}% new
127\LWR@traceinfo{sfsubcap B2}% new
128% }%
129     }%
130     {\dimen@ht0%
131     \advance\dimen@dp0%
132     \ifdim\dimen@>\FBsc@max
133     \global\FBsc@max\dimen@
134     \fi
135     \FB@readaux{\let\FBsubcheight\relax}%
136     \ifx\FBsubcheight\relax
137     \def\next{
138% \parbox[t]{\the\@tempdima}
139%     }%
140     \else
141     \def\next{
142% \parbox[t][\FBsubcheight][t]{\the\@tempdima}
143%     }%
144     \fi
145     \vbox{%
146%         \hb@xt@\the\@tempdima{%
147
148%             \hss
149%             \next{%
150\LWR@traceinfo{sfsubcap C1}% new
151         \caption@make
152             {\@nameuse{sub\@capttype name}}}%
153             {\@nameuse{thesub\@capttype}}}%
154             {#3}
155\LWR@traceinfo{sfsubcap C1}% new
156%     }%
157%         \hss
158
159%     }
160     }
161     }%
162 \egroup
163\LWR@startpars% new
164 }

\caption@@@make {<caption label>} {<caption text>}

165 \renewcommand\caption@@@make [2] {%

```

```

166 \LWR@startpars% new
167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169 \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172 \let\caption@lsep\@empty
173 \let\caption@tfmt\@firstofone
174 }%
175 % \@setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179 {\ifcaption@star\else
180 \begingroup
181 \captionlabelfont
182 #1%
183 \endgroup
184 \fi}%
185 {\ifcaption@star\else
186 \begingroup
187 \caption@iflf\captionlabelfont
188 \relax\caption@lsep
189 \endgroup
190 \fi}%
191 {\captiontextfont
192 \caption@ifstrut
193 {\vrule\@height\ht\strutbox\@width\z@}%
194 {}%
195 \nobreak\hskip\z@skip % enable hyphenation
196 \caption@tfmt{#2}
197 \LWR@ensuredoingapar% new
198 \caption@ifstrut
199 {\ifhmode\@finalstrut\strutbox\fi}%
200 {}%
201 \par}}
202 \LWR@stoppars% new
203 }

```

Patches for `\sf@sub@label`:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \@ifnextchar(% %) match left parenthesis
207 {\sf@sub@label}
208 {\sf@sub@label(Sub\@capytype\space
209 \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210 \@nameuse{p@sub\@capytype}%
211 \@nameuse{thesub\@capytype}.)}}

```

Patches for `\subref`.

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```
212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }
```

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
215 \renewcommand{\sf@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

```
216 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
217
218 \def\@newsubfloat[#1]#2{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}
225 % \def\subtable{\subfloat}
```

Package 116**lwarp-syntonly.sty****193 Syntonly**

Pkg syntonly Emulated.

for HTML output: Discard all options for lwarp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{}
6
7 \@onlypreamble\syntaxonly
```

Package 117

lwarp-tabularx.sty

194 Tabularx

Pkg tabularx tabularx is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabularx:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

Package 118

lwarp-tabulary.sty

195 Tabulary

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.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

Package 119

lwarp-textcomp.sty

196 Textcomp

Pkg textcomp textcomp is patched for use by lwarp.

 **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.

for HTML output: `1 \LWR@ProvidesPackagePass{textcomp}`

The following are restored for print when inside a lateximage:

```

2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
7 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
8 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
9 \LetLtxMacro\LWR@origcapitalacute\capitalacute
10 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
11 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
12 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
13 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
14 \LetLtxMacro\LWR@origcapitalring\capitalring
15 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
16 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
17 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron
18 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
19 \LetLtxMacro\LWR@origtextcircled\textcircled

```

For HTML, use HTML entities or direct Unicode, depending on the engine:

```

20 \ifPDFTeX
21 \renewcommand*{\textdegree}{\HTMLentity{deg}}
22 \renewcommand*{\textcelsius}{\HTMLentity{deg}\,C}
23 \renewcommand*{\textohm}{\HTMLunicode{2126}}
24 \renewcommand*{\textmu}{\HTMLunicode{00B5}}
25 \else
26 \renewcommand*{\textdegree}{°}
27 \renewcommand*{\textcelsius}{°C}
28 \renewcommand*{\textohm}{Ω}
29 \renewcommand*{\textmu}{μ}

```

```
30 \fi
```

For HTML, Unicode diacritical marks are used:

```
31 \renewcommand*{\capitalcedilla}[1]{\#1\HTMLUnicode{0327}}
32 \renewcommand*{\capitalogonek}[1]{\#1\HTMLUnicode{0328}}
33 \renewcommand*{\capitalgrave}[1]{\#1\HTMLUnicode{0300}}
34 \renewcommand*{\capitalacute}[1]{\#1\HTMLUnicode{0301}}
35 \renewcommand*{\capitalcircumflex}[1]{\#1\HTMLUnicode{0302}}
36 \renewcommand*{\capitaltilde}[1]{\#1\HTMLUnicode{0303}}
37 \renewcommand*{\capitaldieresis}[1]{\#1\HTMLUnicode{0308}}
38 \renewcommand*{\capitalhungarumlaut}[1]{\#1\HTMLUnicode{30B}}
39 \renewcommand*{\capitalring}[1]{\#1\HTMLUnicode{30A}}
40 \renewcommand*{\capitalcaron}[1]{\#1\HTMLUnicode{30C}}
41 \renewcommand*{\capitalbreve}[1]{\#1\HTMLUnicode{306}}
42 \renewcommand*{\capitalmacron}[1]{\#1\HTMLUnicode{304}}
43 \renewcommand*{\capitaldotaccent}[1]{\#1\HTMLUnicode{307}}
```

`\textcircled` becomes a span with a rounded border:

```
44 \renewcommand*{\textcircled}[1]{%
45 \InlineClass[border: 1px solid \LWR@currenttextcolor]{\textcircled}{#1}%
46 }
```

`\LWR@restoreorigtextcomp` Called from `\restoreoriginalformatting` when a lateximage is begun.

```
47 \let\LWR@restoreorigtextcomp\relax
48
49 \newcommand*{\LWR@restoreorigtextcomp}{%
50 \let\textdegree\LWR@origtextdegree%
51 \let\textcelsius\LWR@origtextcelsius%
52 \let\textohm\LWR@origtextohm%
53 \let\textmu\LWR@origtextmu%
54 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
55 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
56 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
57 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
58 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
59 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
60 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
61 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
62 \LetLtxMacro\capitalring\LWR@origcapitalring%
63 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
64 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
65 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
66 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
67 \LetLtxMacro\textcircled\LWR@origtextcircled%
68 }
```

Package 120

lwarp-textpos.sty

197 Textpos

Pkg textpos textpos is emulated during HTML output, and the textpos package is ignored.

```
for HTML output: 1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{}{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{}{}
4 \newcommand*\TPGrid{3}[]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*\textblockcolour{1}{}
7 \newcommand*\textblockrulecolour{1}{}
8 \newcommand*\textblockcolor{1}{}
9 \newcommand*\textblockrulecolor{1}{}
10 \newcommand*\tekstblokkulur{1}{}
11 \newcommand*\tekstblokrulekulur{1}{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}{1}{}
16 \newcommand*\showtextsize{}
17 \newcommand{\textblockorigin}{2}{}

```

Package 121

lwarp-theorem.sty

198 Theorem

(Based on original code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle '#1'. Using 'plain'}}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\ifdefinable\csname #1\endcsname
```

```

18  {%
19  \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20  \@definecounter{#1}\@newctr{#1}[#3]%
21  \expandafter\xdef\csname the#1\endcsname
22    {\expandafter \noexpand \csname the#3\endcsname
23     \@thmcountersep \@thmcounter{#1}}%
24  \def\@tempa{\global\@namedef{#1}}%
25  \expandafter \@tempa \expandafter{%
26    \csname th@the \theorem@style
27      \expandafter \endcsname \the \theorem@bodyfont
28    \@thm{#1}{#2}}%
29  \global \expandafter \let \csname end#1\endcsname \@endtheorem
30  \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31  }}
32
33 \gdef\@ynthm#1#2{%
34   \expandafter\@ifdefinable\csname #1\endcsname
35   {
36     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40     \expandafter{\csname th@the \theorem@style \expandafter
41       \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48     \@nocounterr{#2}%
49   \else
50     \expandafter\@ifdefinable\csname #1\endcsname
51     {
52       \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53       \expandafter \xdef \csname the#1\endcsname
54         {\expandafter \noexpand \csname the#2\endcsname}%
55       \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56       \expandafter{\csname th@the \theorem@style \expandafter
57         \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \@endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60     }%
61   \fi}

```

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.

```

62 \gdef\th@plain{%\normalfont\itshape
63   \def\@begintheorem##1##2{%
64     \LWR@forcenewpage% new
65     \BlockClass{theorembody\LWR@thisthmstyle}% new
66     \item[\hskip\labelsep
67       \InlineClass{theoremheader}{##1\ ##2}
68     ]}%
69 \def\@opargbegintheorem##1##2##3{%
70   \LWR@forcenewpage% new
71   \BlockClass{theorembody\LWR@thisthmstyle}% new
72   \item[\hskip\labelsep
73     \InlineClass{theoremheader}{##1\ ##2\ (##3)}
74   ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79     \LWR@forcenewpage% new
80     \BlockClass{theorembody\LWR@thisthmstyle}% new
81     \item[\hskip \labelsep
82       \InlineClass{theoremheader}{##1\ ##2}\newline%
83     ]}%
84 \def\@opargbegintheorem##1##2##3{%
85   \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87   \item[\hskip \labelsep
88     \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89   ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94     \LWR@forcenewpage% new
95     \BlockClass{theorembody\LWR@thisthmstyle}% new
96     \item[\hskip\labelsep %
97       \InlineClass{theoremheader}{##2 \quad ##1}\newline
98     ]}%
99 \def\@opargbegintheorem##1##2##3{%
100   \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103     \InlineClass{theoremheader}{##2 \quad ##1\ %
104     (##3)}\newline
105   ]}
106 }
107

```

```

108 \gdef\th@changebreak{%\normalfont\slshape
109 \def\@begintheorem##1##2{
110 \LWR@forcenewpage% new
111 \BlockClass{theorembody\LWR@thisthmstyle}% new
112 \item[\hskip\labelsep
113 \InlineClass{theoremheader}{##2\ ##1}\newline
114 ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117 \BlockClass{theorembody\LWR@thisthmstyle}% new
118 \item[\hskip\labelsep
119 \InlineClass{theoremheader}{ ##2\ ##1\ %
120 (##3)}\newline
121 ]}
122 }
123
124 \gdef\th@change{%\normalfont\slshape
125 \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127 \BlockClass{theorembody\LWR@thisthmstyle}% new
128 \item[\hskip\labelsep
129 \InlineClass{theoremheader}{##2\ ##1}
130 ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133 \BlockClass{theorembody\LWR@thisthmstyle}% new
134 \item[\hskip\labelsep
135 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136 ]}
137 }
138
139 \gdef\th@margin{%\normalfont\slshape
140 \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142 \BlockClass{theorembody\LWR@thisthmstyle}% new
143 \item[\hskip\labelsep
144 \InlineClass{theoremheader}{##2 \quad ##1}
145 ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148 \BlockClass{theorembody\LWR@thisthmstyle}% new
149 \item[\hskip\labelsep
150 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
151 ]}
152 }

```

Patched for css:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

Package 122

lwarp-threeparttable.sty

199 Threeparttable

Pkg `threeparttable` `threeparttable` is emulated during HTML output, and the `threeparttable` package is ignored.

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 `` of class `tnoteitemheader`, otherwise they are plain text.

for HTML output: `1 \LWR@ProvidesPackageDrop{threeparttable}`

Prints the table note item header inside a css class of `tnoteitemheader`.

```
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

To emulate `threeparttable`:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

```
4 \newenvironment*{tablenotes}[1] []
```

```
5 {%
```

```
6 \LWR@forcenewpage
```

```
7 \BlockClass{tnotes}%
```

```
8 \ltx@ifpackageloaded{enumitem}{%
```

```
9 \setlist[description]{format=\LWR@printtablenote}%
```

```
10 }{}}%
```

```
11 \description%
```

```
12 }
```

```
13 {%
```

```
14 \enddescription%
```

```
15 \endBlockClass%
```

```
16 }
```

```
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

Package 123

lwarp-tikz.sty

200 Tikz

Pkg tikz tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz'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>

for HTML output: 2 \begin{warpHTML}

```
3 \newboolean{LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
7 {\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.

```
8 \BeforeBeginEnvironment{tikzpicture}{%
9 \lateximage%
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
11 {}%
12 {\catcode'\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlateximage%
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }
```

21 \end{warpHTML}

Package 124

lwarp-titles.sty

201 Titles

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

Discard all options for lwarp-titles:

for HTML output: 1 \LWR@ProvidesPackageDrop{titles}

\pagestyle and \thispagestyle are already disabled in the lwarp code.

```

2 \NewDocumentCommand{\newpagestyle}{m o m}{}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

4 \NewDocumentCommand{\sethead}{o o o m m m}{}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

6 \NewDocumentCommand{\settitlemarks}{s m}{}

7 \newcommand*{\headrule}{}
8 \newcommand*{\footrule}{}

9 \newcommand*{\setheadrule}[1]{}
10 \newcommand*{\setfootrule}[1]{}

11 \newcommand*{\makeheadrule}{}
12 \newcommand*{\makefootrule}{}

13 \newcommand{\setmarkboth}[1]{}

14 \NewDocumentCommand{\widenhead}{s o o m m}{}

15 \newcommand*{\bottitlemarks}{}
16 \newcommand*{\toptitlemarks}{}
17 \newcommand*{\firsttitlemarks}{}
18 \newcommand*{\nexttoptitlemarks}{}
19 \newcommand*{\outertitlemarks}{}
20 \newcommand*{\innertitlemarks}{}

21 \NewDocumentCommand{\newtitlemark}{s m}{}

```

```
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

23 \newcommand{\ifsamemark}[4]{}

24 \NewDocumentCommand{\setfloathead}{s o o m m m m m}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o m m m m m}{}

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m m}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o m m m m m}{}

28 \newcommand{\newmarkset}[1]{}

29 \NewDocumentCommand{\newextramarkset}{s m m}{}

30 \newcommand{\botextramarks}[1]{}
31 \newcommand{\topextramarks}[1]{}
32 \newcommand{\firstextramarks}[1]{}
33 \newcommand{\nexttopextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

Package 125

lwarp-titleref.sty

202 Titleref

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

Package 126

lwarp-titlesec.sty

203 Titlesec

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

```

for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

2 \newcommand*{\titlelabel}[1]{}

3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}%
5     {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

8 \@ifundefined{chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

11 \newcommand*{\filright}{}
12 \newcommand*{\filcenter}{}
13 \newcommand*{\filleft}{}
14 \newcommand*{\fillast}{}
15 \newcommand*{\filinner}{}
16 \newcommand*{\filouter}{}

17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
18   \fontdimen\thr@\font \@minus \fontdimen4\font}

19 \NewDocumentCommand{\titleline}{s o m}{}

20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

23 \newcommand{\iftitlemeasuring}[2] {#2}

24 \newcommand{\assignpagestyle}[2] {#2}

```

25 \NewDocumentCommand{\titleclass}{m o m o}

Package 127

lwarp-titletoc.sty

204 Titletoc

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{\}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m m o o}{\}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{\}

6 \newcommand{\contentsmargin}[2] []{\}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2] []{\thecontentslabel}
10 \newcommand{\contentspage}[1] []{\thecontentspage}

11 \newcommand{\contentspush}[1]{\}

12 \newcommand{\contentsuse}[2]{\}

13 \newcommand*{\startcontents}[1] []{\}
14 \newcommand*{\stopcontents}[1] []{\}
15 \newcommand*{\resumecontents}[1] []{\}

16 \newcommand{\printcontents}[4] []{\}

17 \newcommand{\startlist}[2] []{\}
18 \newcommand{\stoplist}[2] []{\}
19 \newcommand{\resumelist}[2] []{\}

20 \newcommand{\printlist}[4] []{\}

```

Package 128

lwarp-titling.sty

205 Titling

Pkg titling titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output: `1 \LWR@ProvidesPackagePass{titling}`

Patch \@bsmtitleempty:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 \global\let\published\relax%
6 \global\let\subtitle\relax%
7 }
```

Patch \keepthetitle:

```
8 \let\LWR@origkeepthetitle\keepthetitle
9 \renewcommand*{\keepthetitle}{%
10 \LWR@orig@keepthetitle%
11 \global\let\@published\@empty%
12 \global\let\@subtitle\@empty%
13 }
```

Patch \killtitle:

```
14 \let\LWR@origkilltitle\killtitle
15 \renewcommand*{\killtitle}{%
16 \LWR@orig@killtitle%
17 \global\let\thepublished\relax%
18 \global\let\thesubtitle\relax%
19 }
```

Package 129

lwarp-tocbibind.sty

206 Tocbibind

Pkg tocbibind tocbibind is patched for use by lwarp.

```
for HTML output: 1 \LWR@ProvidesPackagePass{tocbibind}

2 \renewenvironment{theindex}%
3   {%
4     \if@bibchapter
5       \if@donumindex
6         \chapter{\indexname}
7       \else
8         \if@dotocind
9           \chapter*{\indexname}
10          \addcontentsline{toc}{chapter}{\indexname}
11        \else
12          \chapter*{\indexname}
13        \fi
14      \fi
15    \else
16      \if@donumindex
17        \section{\indexname}
18      \else
19        \if@dotocind
20          \section*{\indexname}
21          \addcontentsline{toc}{\@tocextra}{\indexname}
22        \else
23          \section*{\indexname}
24        \fi
25      \fi
26    \fi
27 \let\item\LWR@indexitem%
28 \let\subitem\LWR@indexsubitem%
29 \let\subsubitem\LWR@indexsubsubitem%
30 }
```

Package 130

lwarp-tocloft.sty

207 Tocloft

`\pkg{tocloft}` `tocloft` is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

Discard all options for `lwarp-tocloft`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*\cftmarktoc{}
4 \newcommand*\cfttoctitlefont{}
5 \newcommand*\cftaftertocitle{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertocitleskip}

8 \newcommand*\cftmarklof{}
9 \newcommand*\cftloftitlefont{}
10 \newcommand*\cftafterloftitle{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

13 \newcommand*\cftmarklot{}
14 \newcommand*\cftlottitlefont{}
15 \newcommand*\cftafterlottitle{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*\cftdot}{.}
19 \providecommand*\cftdotsep}{1}
20 \newcommand*\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*\cftsetpnumwidth}[1]{}
24 \newcommand*\cftsetrmarg}[1]{}

```

```
25 \newcommand*\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartindent}
29 \newlength{\cftpartnumwidth}
30 \newcommand*\cftpartfont{}
31 \newcommand*\cftpartpresnum{}
32 \newcommand*\cftpartaftersnum{}
33 \newcommand*\cftpartaftersnumb{}
34 \newcommand*\cftpartleader{}
35 \newcommand*\cftpartdotsep}{1}
36 \newcommand*\cftpartpagefont{}
37 \newcommand*\cftpartafterpnum{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*\cftchapfont{}
42 \newcommand*\cftchappresnum{}
43 \newcommand*\cftchapaftersnum{}
44 \newcommand*\cftchapaftersnumb{}
45 \newcommand*\cftchapleader{}
46 \newcommand*\cftchapdotsep}{1}
47 \newcommand*\cftchappagefont{}
48 \newcommand*\cftchapafterpnum{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*\cftsecfont{}
53 \newcommand*\cftsecpresnum{}
54 \newcommand*\cftsecaftersnum{}
55 \newcommand*\cftsecaftersnumb{}
56 \newcommand*\cftsecleader{}
57 \newcommand*\cftsecdotsep}{1}
58 \newcommand*\cftsecpagefont{}
59 \newcommand*\cftsecafterpnum{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*\cftsubsecfont{}
64 \newcommand*\cftsubsecpresnum{}
65 \newcommand*\cftsubsecaftersnum{}
66 \newcommand*\cftsubsecaftersnumb{}
67 \newcommand*\cftsubsecleader{}
68 \newcommand*\cftsubsecdotsep}{1}
```

```
69 \newcommand*\cftsubsecpagefont}{  
70 \newcommand*\cftsubsecafterpnum}{  
  
71 \newlength{\cftbeforesubsubsecskip}  
72 \newlength{\cftsubsubsecindent}  
73 \newlength{\cftsubsubsecnumwidth}  
74 \newcommand*\cftsubsubsecfont}{  
75 \newcommand*\cftsubsubsecpresnum}{  
76 \newcommand*\cftsubsubsecaftersnum}{  
77 \newcommand*\cftsubsubsecaftersnumb}{  
78 \newcommand*\cftsubsubsecleader}{  
79 \newcommand*\cftsubsubsecdotsep}{1}  
80 \newcommand*\cftsubsubsecpagefont}{  
81 \newcommand*\cftsubsubsecafterpnum}{  
  
82 \newlength{\cftbeforeparaskip}  
83 \newlength{\cftparaindent}  
84 \newlength{\cftparanumwidth}  
85 \newcommand*\cftparafont}{  
86 \newcommand*\cftparapresnum}{  
87 \newcommand*\cftparaaftersnum}{  
88 \newcommand*\cftparaaftersnumb}{  
89 \newcommand*\cftparaleader}{  
90 \newcommand*\cftparadotsep}{1}  
91 \newcommand*\cftparapagefont}{  
92 \newcommand*\cftparaafterpnum}{  
  
93 \newlength{\cftbeforesubparaskip}  
94 \newlength{\cftsubparaindent}  
95 \newlength{\cftsubparanumwidth}  
96 \newcommand*\cftsubparafont}{  
97 \newcommand*\cftsubparapresnum}{  
98 \newcommand*\cftsubparaaftersnum}{  
99 \newcommand*\cftsubparaaftersnumb}{  
100 \newcommand*\cftsubparaleader}{  
101 \newcommand*\cftsubparadotsep}{1}  
102 \newcommand*\cftsubparapagefont}{  
103 \newcommand*\cftsubparaafterpnum}{  
  
104 \newlength{\cftbeforefigskip}  
105 \newlength{\cftfigindent}  
106 \newlength{\cftfignumwidth}  
107 \newcommand*\cftfigfont}{  
108 \newcommand*\cftfigpresnum}{  
109 \newcommand*\cftfigaftersnum}{  
110 \newcommand*\cftfigaftersnumb}{  
111 \newcommand*\cftfigleader}{  
112 \newcommand*\cftfigdotsep}{1}  
113 \newcommand*\cftfigpagefont}{
```

```

114 \newcommand*\cftfigafterpnum}{ }

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*\cftsubfigfont}{ }
119 \newcommand*\cftsubfigpresnum}{ }
120 \newcommand*\cftsubfigaftersnum}{ }
121 \newcommand*\cftsubfigaftersnumb}{ }
122 \newcommand*\cftsubfigleader}{ }
123 \newcommand*\cftsubfigdotsep}{1}
124 \newcommand*\cftsubfigpagefont}{ }
125 \newcommand*\cftsubfigafterpnum}{ }

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*\cfttabfont}{ }
130 \newcommand*\cfttabpresnum}{ }
131 \newcommand*\cfttabaftersnum}{ }
132 \newcommand*\cfttabaftersnumb}{ }
133 \newcommand*\cfttableader}{ }
134 \newcommand*\cfttabdotsep}{1}
135 \newcommand*\cfttabpagefont}{ }
136 \newcommand*\cfttabafterpnum}{ }

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*\cftsubtabfont}{ }
141 \newcommand*\cftsubtabpresnum}{ }
142 \newcommand*\cftsubtabaftersnum}{ }
143 \newcommand*\cftsubtabaftersnumb}{ }
144 \newcommand*\cftsubtableader}{ }
145 \newcommand*\cftsubtabdotsep}{1}
146 \newcommand*\cftsubtabpagefont}{ }
147 \newcommand*\cftsubtabafterpnum}{ }

148 \newcommand{\cftsetindents}[3]{ }

149 \newcommand{\pagenumbersoff}[1]{ }
150 \newcommand{\pagenumberon}[1]{ }

Emulated through the \newfloat mechanism.

151 \NewDocumentCommand{\newlistof}{o m m m}
152 { %
153 \IfValueTF{#1}

```

```
154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }
```

\cftchapterprecis from tocloft:

```
170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }
```

Package 131

lwarp-transparent.sty

208 Transparent

(Based on original code by HEIKO OBERDIEK.)

Pkg transparent Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not Xe_{La}TeX!** Note that transparent does not work with Xe_{La}TeX.

for HTML output: Discard all options for lwarp-transparent:

```

1 \LWR@ProvidesPackageDrop{transparent}
2 \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 }

```

Package 132

lwarp-trivfloat.sty

209 Trivfloat

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.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

for HTML & PRINT:

```
3 \begin{warpall}
```

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
4 \end{warpall}
```

\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:

```
5 \begin{warpHTML}

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

7 \end{warpHTML}
```

209.1 Combining \newfloat, \trivfloat, and algorithmicx

for HTML & PRINT:

```
8 \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
```

```
9 \end{warpall}
```

Package 133

lwarp-ulem.sty

210 Ulem

(Based on original code by DONALD ARSENEAU.)

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \LetLtxMacro\LWR@ulemorigemph\emph
2 \LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using CSS:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \InlineClass{uline}{#1}%
5 }
6
7 \NewDocumentCommand{\uuline}{+m}{%
8 \InlineClass{uuline}{#1}%
9 }
10
11 \NewDocumentCommand{\uwave}{+m}{%
12 \InlineClass{uwave}{#1}%
13 }
14
15 \NewDocumentCommand{\sout}{+m}{%
16 \InlineClass{sout}{#1}%
17 }
18
19 \NewDocumentCommand{\xout}{+m}{%
20 \InlineClass{xout}{#1}%
21 }
22
23 \NewDocumentCommand{\dashuline}{+m}{%
24 \InlineClass{dashuline}{#1}%
25 }
26
27 \NewDocumentCommand{\dotuline}{+m}{%
28 \InlineClass{dotuline}{#1}%
29 }
```

Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{-}{-}
31 \newlength{\ULdepth}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}{m}{-}
33 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
34 \NewDocumentCommand{\useunder}{m m m}{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37   \def#3{#1}\MakeRobust{#3}\fi
38 }
```

Triggered by package options, also available for the users:

```
39 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
40 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
41 \ULforem% default
```

Package options:

```
42 \DeclareOption{normalem}{\normalem}
43 \DeclareOption{ULforem}{\ULforem}
44 \DeclareOption{normalbf}{-}
45 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}
```

Emulate the original package:

```
46 \LWR@ProvidesPackageDrop{ulem}
```

Package 134**lwarp-upref.sty****211 Upref**

Pkg upref Ignored.

for HTML output: Discard all options for lwarp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

Package 135

lwarp-verse.sty

212 Verse

(Based on original code by PETER WILSON.)

Pkg `verse` `verse` is supported and patched by `lwarp`.

for HTML output: Pass all options for `lwarp-verse`:

```
1 \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` 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.

Len `\leftmargini`

Len `\TMLvleftskip`

Len `\TMLleftmargini`

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`.

```

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}

```

At the beginning of the verse environment:

```

4 \AtBeginEnvironment{verse}
5 {%

```

Pkg verse The verse or memoir packages can place stanza numbers to the left with their
Pkg memoir \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:
Len \leftskip

```

6 \ifdef{\vleftskip}{%
7 \setlength{\vleftskip}{\HTMLvleftskip}
8 \setlength{\leftmargini}{\HTMLleftmargini}
9 }{}
10 \LWR@forcenewpage
11 \LWR@atbeginverbatim{verse}
12 \unskip\LWR@origvspace{-\baselineskip}
13 }

```

After the end of the verse environment, which places the pre tag at the regular left margin:

```

14 \AfterEndEnvironment{verse}{
15 \unskip\LWR@origvspace{-\baselineskip}
16 \LWR@afterendverbatim
17 }

```

Patch to place poemtitle inside an HTML span of class poemtitle:

```

18 \ifdef{\poemtitle}{
19 \DeclareDocumentCommand{\vstypeptitle}{m}{%
20 \vspace{\beforepoemtitleskip}%
21 {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22 \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 }% AfterEndPreamble

```

Package 136

lwarp-wallpaper.sty

213 Wallpaper

Pkg wallpaper wallpaper is emulated during HTML output, and the wallpaper package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{wallpaper}
2 \newcommand*\CenterWallPaper}[2]{}
3 \newcommand*\ThisCenterWallPaper}[2]{}
4 \newcommand*\TileWallPaper}[3]{}
5 \newcommand*\ThisTileWallPaper}[3]{}
6 \newcommand*\TileSquareWallPaper}[2]{}
7 \newcommand*\ThisTileSquareWallPaper}[2]{}
8 \newcommand*\ULCornerWallPaper}[2]{}
9 \newcommand*\ThisULCornerWallPaper}[2]{}
10 \newcommand*\LLCornerWallPaper}[2]{}
11 \newcommand*\ThisLLCornerWallPaper}[2]{}
12 \newcommand*\URCornerWallPaper}[2]{}
13 \newcommand*\ThisURCornerWallPaper}[2]{}
14 \newcommand*\LRCornerWallPaper}[2]{}
15 \newcommand*\ThisLRCornerWallPaper}[2]{}
16 \newcommand*\ClearWallPaper[2]{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

Package 137

lwarp-wrapfig.sty

214 Wrapfig

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the HTML style.

```

2 \newlength{\LWR@wrapwidth}

3 \newcommand*\LWR@wrapposition{-}
4
5 \newcommand*\LWR@subwrapfigure[2]{%
6 \LWR@maybeinthisfloat%
7 \renewcommand*\LWR@wrapposition{-}%
8 \ifthenelse{%
9   \equal{#1}{r}\OR\equal{#1}{R}\OR%
10  \equal{#1}{o}\OR\equal{#1}{O}%
11 }%
12 {\renewcommand*\LWR@wrapposition}{float:right}}%
13 {\renewcommand*\LWR@wrapposition}{float:left}}%
14 \setlength{\LWR@wrapwidth}{#2}%
15 \addtolength{\LWR@wrapwidth}{4em}%
16 \uselengthunit{PT}%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
20 style="width:\rndprintlength{\LWR@wrapwidth} ; %
21 \LWR@wrapposition"%
22 }%
23 \LWR@startpars
24 }
25
26
27 \NewDocumentEnvironment{wrapfigure}{o m o m}
28 {%
29 \LWR@subwrapfigure{#2}{#4}%
30 \captionsetup{type=figure}%
31 }
32 {%
33 \LWR@htmldivclassend{div}
34 }

```

```
35
36
37 \NewDocumentEnvironment{wraptable}{o m o m}
38 {%
39 \LWR@subwrapfigure{#2}{#4}%
40 \captionsetup{type=table}%
41 }
42 {%
43 \LWR@htmldivclassend{div}
44 }
45
46
47 \NewDocumentEnvironment{wrapfloat}{m o m o m}
48 {%
49 \LWR@subwrapfigure{#3}{#5}%
50 \captionsetup{type=#1}%
51 }
52 {%
53 \LWR@htmldivclassend{div}
54 }
55
56 \newlength{\wrapoverhang}
```

Package 138

lwarp-xcolor.sty

215 Xcolor

`Pkg xcolor` xcolor is supported by lwarp.

`\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 `<div>` instead of a ``. 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.

215.1 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 redefinitions 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 lwarp is loaded in print or HTML mode: `\LWR@restoreorigxcolor` and `\LWR@restoremorigxcolor` are defined as null. Section 69.1

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@restoreorigxcolor` is defined 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 69.2. 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@restoremorigxcolor` is defined to temporarily set these functions to their print-mode versions.

In a lateximage in HTML mode: `\LWR@restoreorigxcolor` and `\LWR@restoremorigxcolor` are used to temporarily restore the print-mode definitions of xcolor’s functions. See `\LWR@restoreorigformatting` on page 323.

\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 571.

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 571.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes \colorbox.

HTML: Newly defined by lwarp-xcolor to use a <div>, page 572.

HTML lateximage: Remembers and reuses the print version \colorbox.

\fcolorbox:

Print: Modified to allow a background of none.

\LWRprint@fcolorbox at section 69.2

HTML: Redefined by lwarp-xcolor, page 572.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes \fcolorbox. Section 69.2

HTML: Newly defined by lwarp-xcolor to use a <div>, page 573.

HTML lateximage: Remembers and reuses the print version \fcolorbox.

fcolorminipage:

Print: Newly defined in the lwarp core.

LWRprint@fcolorminipage at section 69.2

HTML: Newly defined by lwarp-xcolor, page 573.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 574.

HTML lateximage: Remembers and reuses the print version.

215.2 Code

```
for HTML output: 1 \LWR@ProvidesPackagePass{xcolor}
                  2 \begin{warpHTML}
```

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```

3 \LetLtxMacro\LWRprint@textcolor\textcolor
4 \LetLtxMacro\LWRprint@pagecolor\pagecolor
5 \LetLtxMacro\LWRprint@nopagecolor\nopagecolor
6 \LetLtxMacro\LWRprint@colorbox\colorbox
7 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
9 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
10 \LetLtxMacro\LWRorigprint@boxframe\boxframe

```

`\LWR@restoreorigxcolor` Inside a `lateximage` the following gets restored to their print-mode actions:

```

11 \renewcommand*{\LWR@restoreorigxcolor}{%
12 \LWR@traceinfo{\LWR@restoreorigxcolorformatting}%
13 \LetLtxMacro\textcolor\LWRprint@textcolor%
14 \LetLtxMacro\pagecolor\LWRprint@pagecolor%
15 \LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
16 \LetLtxMacro\colorbox\LWRprint@colorbox%
17 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
18 \LetLtxMacro\boxframe\LWRorigprint@boxframe%
19 }

```

`\LWR@tempcolor` The color converted to HTML colorspace.

```

20 \newcommand*{\LWR@tempcolor}{}
21 \newcommand*{\LWR@tempcolortwo}{}

```

`\LWR@colorstyle` `{\langle 1: styletext \rangle}{\langle 2: model \rangle}{\langle 3: color \rangle}`

For a color style, prints the color converted to HTML colors.

```

22 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
23 \begingroup%
24 \LWR@FBcancel%

```

Use the `xcolor` package to convert to an HTML color space:

```

25 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%

```

Print the converted color:

```

26 \#\LWR@tempcolor%
27 \endgroup%
28 }

```

`\LWR@borderpadding` `{\langle colorstyle \rangle}{\langle color \rangle}` Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
29 \newcommand*\LWR@borderpadding}[2]{%
30 \uselengthunit{PT}%
31 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
32 padding:\rndprintlength{\fboxsep}%
33 }
```

`\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.

```
34 \RenewDocumentCommand{\textcolor}{0{named} m m}{%
35 \begingroup%
36 \LWR@FBCancel%
37 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
38 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
39 \renewcommand*\LWR@currenttextcolor}{#\LWR@tempcolor}%
40 #3%
41 }%
42 \endgroup%
43 }
```

`\pagecolor` [*model*] {*color*}

Ignored. Use `\CSSFilename` instead.

```
44 \renewcommand*\pagecolor}[2][named]{}
```

`\nopagecolor` Ignored.

```
45 \let\nopagecolor\relax
```

`\colorbox` [*model*] {*color*} {*text*}

Converted into an HTML hex background color ``.

```
46 \RenewDocumentCommand{\colorbox}{0{named} m +m}{%
47 \begingroup%
```

```

48 \LWR@FBcancel%
49 \uselengthunit{PT}%
50 \InlineClass[%
51 background:\LWR@colorstyle{#1}{#2} ; %
52 padding:\rndprintlength{\fboxsep}%
53 ]{colorbox}{#3}%
54 \endgroup%
55 }

```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```

56 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
57 \begingroup%
58 \LWR@FBcancel%
59 \uselengthunit{PT}%
60 \begin{BlockClass}[%
61 background:\LWR@colorstyle{#1}{#2} ; %
62 padding:\rndprintlength{\fboxsep}%
63 ]{colorboxBlock}
64 #3
65 \end{BlockClass}%
66 \endgroup%
67 }

```

`\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.

```

68 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
69 \LWR@traceinfo{HTML fcolorbox #2 #4}%
70 \begingroup%
71 \LWR@FBcancel%
72 \uselengthunit{PT}%
73 \LWR@forceminwidth{\fboxrule}%
74 \ifthenelse{\equal{#4}{none}}%
75 {% no background color
76   \InlineClass[%
77     \LWR@borderpadding{#1}{#2}%
78   ]{fcolorbox}{#5}%
79 }%
80 {% yes background color
81   \InlineClass[%
82     \LWR@borderpadding{#1}{#2} ; %

```

```

83   background:\LWR@colorstyle{#3}{#4}%
84   ]{fcolorbox}{#5}%
85 }%
86 \endgroup%
87 }

```

`\fcolorboxBlock` [*⟨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.

```

88 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
89 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
90 \begingroup%
91 \LWR@FBcancel%
92 \uselengthunit{PT}%
93 \LWR@forceminwidth{\fboxrule}%
94 \ifthenelse{\equal{#4}{none}}{%
95 {% no background color
96   \begin{BlockClass}[%
97   \LWR@borderpadding{#1}{#2}%
98   ]{fcolorboxBlock}
99   #5
100  \end{BlockClass}%
101 }%
102 {% yes background color
103   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
104   \begin{BlockClass}[%
105   background:\#\LWR@tempcolortwo; %
106   \LWR@borderpadding{#1}{#2}%
107   ]{fcolorboxBlock}
108   #5
109   \end{BlockClass}%
110 }%
111 \endgroup%
112 \LWR@traceinfo{HTML fcolorboxBlock done}%
113 }

```

Env `fcolorminipage` [*⟨1:framemodel⟩*] {*⟨2:framecolor⟩*} [*⟨3:boxmodel⟩*] {*⟨4:boxcolor⟩*} [*⟨5:align⟩*] [*⟨6:height⟩*] [*⟨7:inner-align⟩*] {*⟨8:width⟩*}

Creates a framed HTML `<div>` around its contents.

A print-output version is defined in the lwarp core: section [69.2](#)

`\LWR@subfcolorminipage` {*⟨framemodel⟩*} {*⟨framecolor⟩*} {*⟨background tag⟩*} {*⟨height⟩*}

```

114 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
115 \begin{BlockClass}[%
116 #3%
117 \LWR@borderpadding{#1}{#2} ; %
118 \IfValueTF{#4}{height:\rndprintlength{\LWR@tempheight} ; }{%
119 width:\rndprintlength{\LWR@tempwidth}%
120 }{fcolorminipage}%
121 }

```

`fcolorminipage` [$\langle 1:framestyle \rangle$] [$\langle 2:framecolor \rangle$] [$\langle 3:bkgcolorstyle \rangle$] [$\langle 4:bkgcolor \rangle$] [$\langle 5:align \rangle$] [$\langle 6:height \rangle$] [$\langle 7:inneralign \rangle$] [$\langle 8:width \rangle$]

```

122 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
123 {%
124 \LWR@FBcancel%
125 \setlength{\LWR@tempwidth}{#8}%
126 \IfValueTF{#6}{\setlength{\LWR@tempheight}{#6}}{%
127 \uselengthunit{PT}%
128 \LWR@forceminwidth{\fboxrule}%
129 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
130 \ifthenelse{\equal{#4}{none}}{%
131 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
132 {%
133     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
134     \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ;}{#6}%
135 }%
136 }
137 {\end{BlockClass}}

```

`\boxframe` [$\langle width \rangle$] [$\langle height \rangle$] [$\langle depth \rangle$]

The depth is added to the height, but the box is not decended below by the depth.
`\textcolor` is honored.

```

138 \renewcommand*\boxframe}[3]{%
139 {%
140 \setlength{\LWR@tempwidth}{#1}%
141 \setlength{\LWR@tempheight}{#2}%
142 \addtolength{\LWR@tempheight}{#3}%
143 \uselengthunit{PT}%
144 \LWR@forceminwidth{\fboxrule}%
145 \InlineClass[%
146 display:inline-block ; %
147 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
148 width:\rndprintlength{\LWR@tempwidth} ; %
149 height:\rndprintlength{\LWR@tempheight}%
150 ]{boxframe}{}%
151 }%

```

152 }

153 \end{warpHTML}

Package 139

lwarp-xfrac.sty

216 Xfrac

Pkg xfrac Supported by adding xfrac instances.

for HTML output: 1 \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: 2 \begin{warpall}

\xfracHTMLfontsize User-redefinable macro which controls the font size of the fraction.

3 \newcommand*\xfracHTMLfontsize{.6em}

4 \end{warpall}

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
```

```

11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }

```

`\scalebox` A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```

16 \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.

```

17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
21 denominator-format = {%
22 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
23 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

24 scaling = false
25 }
26 \DeclareInstance{xfrac}{lmr}{text}{
27 numerator-format = {%
28 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
29 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
30 denominator-format = {%
31 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
32 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{
36 numerator-format = {%
37 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
38 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
39 denominator-format = {%
40 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
41 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{
45 numerator-format = {%
46 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
47 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
48 denominator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
51 scaling = false
52 }

53 \end{warpHTML}
```

Package 140

lwarp-xltxtra.sty

217 Xltxtra

Pkg xltxtra xltxtra is emulated.

(Based on original code by WILL ROBERTSON, JONATHAN KEW.)

for HTML output:

```
1 \LWR@ProvidesPackageDrop{xltxtra}
2 \RequirePackage{realscripts}
3 \RequirePackage{metalogo}
4 \newcommand*\TeX@logo@spacing[6]{
5
6 \newcommand*\vfrac[2]{%
7 \textsuperscript{#1}/\textsubscript{#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11 \@tempcnta=\XeTeXglyphindex "#1"\relax
12 \ifnum\@tempcnta>0
13 \XeTeXglyph\@tempcnta
14 \else
15 \xxt@namedglyph@fallback{#1}%
16 \fi}
17
18 \newcommand\xxt@namedglyph@fallback[1]{[#1]}
19
20 \newcommand{\showhyphens}[1]{}
```

Package 141

lwarp-xmpincl.sty

218 Xmpincl

Pkg xmpincl Emulated.

for HTML output: Discard all options for lwarp-xmpincl:

```
1 \LWR@ProvidesPackageDrop{xmpincl}
```

```
2 \newcommand*{\includemp}[1]{}
```

Change History and Index

Change History

| | | |
|-------|--|---|
| v0.10 | General: 2016/03/08 Initial version . . . 1 | Docs: Commands for a successful HTML conversion. 71 |
| v0.11 | General: 2016/03/11 1 | Docs: Commands into a warpprint environment. 68 |
| | Added section: Operating-System portability. 106 | Docs: Newclude limitations. . . . 89 |
| | Added section: Selecting the operating system. 66 | Docs: Table: Cross-referencing data structures. 292 |
| | Test Suite: limages and index in README.txt 1 | Docs: Table: Float data structures. 302 |
| | Test Suite: MS-Windows in README.txt 1 | Docs: Trademarks section. 385 |
| v0.12 | | Docs: Troubleshooting cross-references. 98 |
| | \LWR@newhtmlfile: Bugfix: TOC with numbered files. 217 | Test Suite: Assigned cleveref name for Test Float. 1 |
| | General: 2016/03/14 1 | Test Suite: Floatrow 1 |
| | Global: Uses \p@(type) in float captions. 1 | v0.15 |
| | Test Suite: Sub-figures 1 | \printpublished: No div if empty. 242 |
| v0.13 | | \printsubtitle: No div if empty. . 242 |
| | \CaptionSeparator: Fix for newer babel package. 304 | General: 2016/04/06 1 |
| | \LWR@LwarpStart: \up and \fup . . 232 | Added 435 |
| | \published: Default to empty published. 238 | Ampersand (&): Fixed handling when passed as an argument. . . 263 |
| | \subtitle: Default to empty subtitle. 239 | Docs: Added warning icons for items needing special attention. 103 |
| | General: 2016/03/24 1 | Docs: Clarify print/HTML output. 67 |
| | Removed package: subfig 1 | Docs: Moved the supported functions table to the introduction. 32 |
| | Test Suite: Ordinals, Subcaption . . 1 | Files: lwarp_formal.css added. . . . 1 |
| | tikzpicture: Fix dollar-redefined bug for newer package. 542 | Fix: steps counter 434 |
| v0.14 | | Fixed & handling. 433 |
| | \LWR@htmlsectionfilename: Fix: Links to home page. 187 | Test Suite: test_suite_formal.css file added. 1 |
| | General: 2016/03/31 1 | v0.16 |
| | floatrow: Added. 430 | General: 2016/04/11 1 |
| | | \titlingpage: Improved print-output spacing. 243 |
| | | Added XeLaTeX, LuaLaTeX support. 115 |

| | | | |
|---|---------|---|-----|
| Docs: Font and UTF-8 support. | 64 | Files: lwarp.css and lwarp_formal.css: Improved responsive design. | 1 |
| Docs: Moved location of <code>\usepackage{lwarp}</code> | 65 | Microtype disabled during HTML generation | 116 |
| Docs: Text not converting. | 98 | PDF Unicode input characters. | 104 |
| Fix: amsmath options clash | 119 | Test Suite: Verse package | 1 |
| Fix: newtxmath compatibility. | 119 | <code>lateximage: pdfcrop: -- hires</code> added. | 337 |
| Lwarp no longer selects fonts. | 64, 115 | Reorganize <code>\HomeHTMLFilename</code> logic. | 337 |
| Removed package: suffix | 1 | Surpress extra space. | 337 |
| Test Suite: Improved titlingpage. | 243 | <code>verse: Supports verse, memoir</code> packages. | 562 |
| Test Suite: Lwarp no longer selects fonts. | 1 | <code>minipage: Fix: \linewidth,</code> <code>\textwidth, \textheight</code> inside a minipage. | 363 |
| Test Suite: Supports XeLaTeX, LuaLaTeX. | 1 | | |
| <code>xfrac: Adjusted for the use of any</code> font: | 577 | | |
| v0.17 | | v0.19 | |
| <code>\LWR@htmlsectionfilename: Fix:</code> Links when entire doc is one HTML page. | 187 | <code>\HTMLFilename: Docs: Escape</code> filename underscores. | 187 |
| General: 2016/04/14 | 1 | <code>\HomeHTMLFilename: Docs: Escape</code> filename underscores. | 187 |
| <code>mdframed: Added.</code> | 472 | <code>\LWR@LwarpStart: Enabled \ \ equal</code> to <code>\newline</code> | 231 |
| Test Suite: Fix: Print-version front-matter page numbers. | 1 | <code>\LWR@doubledollar: MathJax</code> support. | 320 |
| Test Suite: Mdfamed | 1 | <code>\LWR@filestart: lwarp_mathjax.txt</code> loaded. | 229 |
| v0.18 | | <code>\LWR@hspace: Fix: \hspace length</code> computations. | 377 |
| <code>\LWR@hspace: \hspace</code> supported. | 377 | <code>\LWR@minipagestartpars:</code> Surpresses paragraph tags between minipages. | 376 |
| <code>\LWR@includegraphicsb: Add: svgz</code> file extension. | 347 | <code>\LWR@singledollar: MathJax</code> support. | 320 |
| em, ex, %, px dimensions preserved. | 347 | <code>\LWR@tabledatamultirowtag:</code> Added optional args. | 286 |
| Fix: <code>\linewidth, \textwidth,</code> <code>\textheight</code> inside a minipage. | 347 | <code>\LateximageFontSizeName: Add:</code> User-adjustable math/lateximage font size. | 336 |
| Improved HTML output linebreaks. | 347 | <code>\minipagefullwidth: Added: No</code> width tag for the next minipage in HTML. | 362 |
| <code>\LWR@myshorttoc: Reorganize</code> <code>\HomeHTMLFilename</code> logic. | 308 | <code>\rule: Added</code> | 381 |
| <code>\LWR@newhtmlfile: sideroc</code> after title, improving responsive design. | 216 | <code>\warpHTMLonly: Added.</code> | 110 |
| <code>\LWR@requesttoc: Reorganize</code> <code>\HomeHTMLFilename</code> logic. | 234 | <code>\warpprintonly: Replaces</code> <code>\rowprintedonly</code> | 110 |
| <code>\LWR@subhyperref: Improved HTML</code> output linebreaks. | 299 | <code>\xfracHTMLfontsize: Added.</code> | 576 |
| <code>\LWR@subhyperrefclass: Improved</code> HTML output linebreaks. | 299 | General: 2016/06/08 | 1 |
| <code>\LWR@subinlineimage: Surpress</code> extra space. | 301 | File: lwarp.css: Improved TOC outline display. | 1 |
| General: 2016/05/19 | 1 | | |

| | |
|--|---------|
| cleveref: Loaded | |
| \AtEndPreamble. | 358 |
| CSS for table note item. | 541 |
| Docs: multirow browser bug workaround. | 86, 259 |
| Docs: Math options. | 65 |
| Docs: Table: Cross-referencing data structures, updated. | 292 |
| File: lwarp_mathjax.txt added. | 1 |
| File: lwarp.css: nteitemheader added | 1 |
| Introduction: MathJax support mentioned. | 31 |
| MathJax support added. 321, 327, 328 | |
| Options: mathsvg and mathjax | 108 |
| Supports colored \rule. | 570 |
| titles: null \pagestyle and \thispagestyle for HTML. | 544 |
| tikzpicture: Adapts to tikz version. | 542 |
| equation: MathJax support. | 326 |
| v0.20 | |
| \BlockClassSingle: Renamed from "LWR@htmldivclassline". | 199 |
| \HTMLDescription: Added \NewHTMLdescription. (Renamed in v0.30.) | 209 |
| \HTMLFilename: No longer escape underscores. | 187 |
| \HomeHTMLFilename: No longer escape underscores. | 187 |
| \InlineClass: Renamed from "inlineclass". | 200 |
| \LWR@LwarpStart: Fix: math cross references. | 232 |
| \LWR@closeparagraph: \unskip extra spaces. | 203 |
| No break tags in the start/end of a tabular. | 203 |
| \LWR@endoffline: Fix: \\ | 376 |
| \LWR@filestart: Adds meta description. | 229 |
| \LWR@hspace: Add: Supports HTML thin breakable space. | 377 |
| \LWR@htmldivclass: Added optional style. | 198 |
| \LWR@htmlclass: Added optional style. | 198 |
| \LWR@htmlsectionfilename: HTMLFilename: removed | |
| additional trailing '-', and may be empty. | 187 |
| Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix. | 187 |
| \LWR@includegraphicsb: Fix: \linewidth in a floatrow. | 347 |
| Fix: Expands filename. | 347 |
| \LWR@longtabledatacaptiontag: Fix: Pars in captions. | 282 |
| \LWR@section: Combined higher-level sections together into files. | 221 |
| \LWR@setOSWindows: Auto-detects operating system. | 106 |
| \LWR@subhtmlclass: Factored code. | 197 |
| \SetHTMLFileNumber: Add: Control file numbers. | 187 |
| \cpagerefFor: User-redefinable word for page references. | 359 |
| \dotfill: Inserts an ellipsis. | 374 |
| \hfill: Inserts a \qqquad. | 374 |
| \hrulefill: Inserts a short rule. | 374 |
| \hyperindexref: Print mode provided in case hyperref not used. | 316 |
| \pageref: Added. | 298 |
| \tracinglwarp: Added. | 125 |
| General: 2017/02/09 | 1 |
| afterpage: Added. | 388 |
| alltt: Added. | 390 |
| bookmark: Added. | 396 |
| caption and subcaption supported. | 1 |
| cleveref and referencing patches: Applied \AfterEndPreamble. | 359 |
| draftwatermark: Added. | 405 |
| eso-pic: Added. | 411 |
| everypage: Added. | 412 |
| extramarks: Added. | 413 |
| fancyhdr: Added. | 418 |
| hyperref: Additional user macros. | 452 |
| keyfloat: Added. | 458 |
| letterspace: User-interface emulated. | 460 |
| listings: Added. | 463 |
| ltxcaption: Added. | 469 |
| lwarp-newproject: Added. | 130 |
| microtype: User-interface emulated. | 482 |

| | | | |
|--|-----|--|-----|
| needspace: Added. | 489 | \LWR@filestart: Skip title if not given. | 229 |
| nowidow: Added. | 492 | \LWR@newhtmlfile: Skip title if not given. | 216 |
| placeins: Added. | 506 | \marginpar: Fixed source listing. . . | 213 |
| ragged2e: Added. | 508 | General: 2017/02/23 | 1 |
| setspace: Improved support. . . . | 516 | fontenc: Added. | 438 |
| textpos: Added. | 536 | fontspec: Added. | 438 |
| titles: Added. | 544 | inputenc: Added. | 457 |
| titlesec: Added. | 546 | newclude: Added. | 489 |
| titletoc: Added. | 548 | newunicodechar: Added. | 490 |
| titling: Improved compatibility. . | 549 | lwarpmk: Fix: lwarpmk again for Windows. | 170 |
| tocloft: Added. | 551 | lwarpmk: Fix: lwarpmk images for Windows. | 170 |
| wallpaper: Added. | 564 | lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script. | 170 |
| wrapfig: Added. | 565 | Add: Errors for misplaced packages. | 111 |
| Added @, <, > columns. | 260 | Docs: Added internet class. . . . | 36 |
| Added single-expansion data arrays. | 185 | Docs: Added TeX2page, GladTeX. . . | 36 |
| Code factored into independent lwrap_html files. | 385 | Docs: Installing on Windows. . . . | 42 |
| Docs: Examples for generating HTML file names. | 57 | File lwrap_tutorial.txt added. . . . | 46 |
| Docs: Improved index. | 1 | v0.22 | |
| Enhanced titling support. | 240 | \LWR@parseDcolumn: Added tabular D column. | 270 |
| File: lwrap.css: Minor fixes for validation. | 1 | \LWR@parsebangcolumn: Added tabular ! column. | 267 |
| File: lwarpmk used to compile print, HTML, indexes, and lateximages. | 1 | \LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns. . . . | 272 |
| Fix: \linewidth in a floatrow. . . | 433 | \LWR@printmccoldata: Added tabular D, !, and X columns. . . . | 279 |
| Improved float caption type handling. | 426 | General: 2017/03/02 | 1 |
| Moved sidebar and example code to test suite. | 1 | abstract: Added. | 386 |
| Page geometry set to 6in wide with large margins. | 116 | change page: Added. | 400 |
| Parallel versions of aux files for print/HTML. | 1 | dcolumn: Added. | 404 |
| Removed reliance on make, grep, gawk. | 1 | ftnright: Added. | 446 |
| Tabular: \unskip extra spaces. . . | 260 | geometry: Nullified commands. . . | 447 |
| Test Suite: HTML meta descriptions. | 1 | indentfirst: Added. | 457 |
| Verbatim: Added. | 419 | layout: Added. | 460 |
| verbatim: Added. | 254 | lscap: Added. | 469 |
| BlockClass: Added optional style. . | 199 | mcaption: Added. | 471 |
| Renamed from "blockclass". . . . | 199 | nameref: Added. | 488 |
| LWR@nestspan: Fix: Minipages inside a span. | 194 | nextpage: Added. | 490 |
| v0.21 | | parskip: Added. | 505 |
| \LWR@LwrapStart: Changed lateximages to a .txt file. . . . | 231 | showkeys: Added. | 518 |
| | | sidecap: Added. | 519 |
| | | tabularx: Added. | 532 |

| | | | |
|---|--------|---|--------|
| varioref: Supported. | 77 | Fix: Allows Xe \LaTeX and Lua \LaTeX to | |
| verse: Added. | 562 | preload graphics and graphicx. | 112 |
| v0.23 | | v0.26 | |
| \LWR@parsetablecols: Fix for vert | | General: 2017/03/31 | 1 |
| bar column type. | 272 | cutwin: Added. | 403 |
| \LWR@printmccoldata: Fix for vert | | endnotes: Added. | 407 |
| bar column type. | 279 | floatflt: Added. | 428 |
| General: 2017/03/02 | 1 | footmisc: Added. | 439 |
| v0.24 | | footnotehyper: Added. | 442 |
| \LWR@hspace: Add: \hspace \fill | | footnote: Added. | 441 |
| converts to 2em | 377 | marginfix: Added. | 470 |
| \LWR@htmlfileref: Fix: Index links | | marginnote: Added. | 471 |
| while \tracinglwrap. | 295 | mparhack: Added. | 485 |
| \hypertocfloat: List of floats | | pagenote: Supported as-is. | 504 |
| responds to lofdepth, | | sidenotes: Added. | 520 |
| lotdepth. | 314 | lwrap.css: Improved responsive | |
| General: 2017/03/15 | 1 | marginpar and marginblock. | 132 |
| floatrow: Support for subfig. | 430 | Docs: Improved Mi \TeX install | |
| subfig: Added. | 525 | instructions. | 39, 42 |
| tikz: For tikz v3.0.0 or later, | | Dollar span avoided in a | |
| auto-loads tikz babel library if | | lateximage. | 319 |
| necessary. | 542 | Footnotes now are \LaTeX boxes | |
| Docs: Filename underscore. | 55, 72 | instead of pagenotes. | 209 |
| No longer preloads subcaption; | | lateximage: Labels track page | |
| conflicted with subfig. | 118 | numbers of lateximages. | 337 |
| picture: Fix for inline images. | 360 | Print mode now uses a minipage | |
| tikzpicture: Fix for inline images. | 542 | of \linewidth. | 337 |
| v0.25 | | picture: Fix for \makebox in | |
| \LWR@loadnever: Added the ability | | picture. | 360 |
| to prevent conflicting packages. | 112 | v0.27 | |
| \addcontentsline: Handles | | \LWR@footnotetext: Fix for table | |
| theorems. | 307 | footnote par tags. | 211 |
| General: 2016/03/22 | 1 | General: 2017/04/04 | 1 |
| amsthm: Added. | 391 | lettrine: Added. | 461 |
| ccaption: Prevented. | 400 | microtype: Fix with Xe \LaTeX , | |
| ellipsis: Added. | 405 | Lua \LaTeX | 482 |
| emptypage: Added. | 406 | soul: Added. | 523 |
| framed: Added. | 443 | ulem: Added. | 559 |
| lips: Added. | 462 | Docs: Installing utilities for | |
| mdframed: Help avoid | | MacOS. | 44 |
| hyphenation. | 473 | Docs: Limitations of saveboxes. | 73 |
| ntheorem: Added. | 493 | Page geometry modified to reduce | |
| showidx: Added. | 518 | line overflow. | 116 |
| theorem: Added. | 537 | v0.28 | |
| Basic \LaTeX theorems: improved | | \@wrindex: Improved indexing. | 316 |
| css. | 254 | \HTMLAuthor: Added \HTMLauthor. | |
| Docs: Adds credits for patched | | (Renamed in v0.30.) | 209 |
| code. | 1 | \LWR@LwrapEnd: If FormatEpub or | |
| Docs: Testing lwrap. | 96 | FormatWordProcessor, no | |
| | | bottom nav. | 234 |

| | | | |
|-------------------------------------|-----|-----------------------------------|-----|
| \LWR@LwarpStart: | | Docs: Using a glossary | 62 |
| FormatWordProcessor forces | | File *.lwarpmkconf: Add: | |
| single-file output. | 231 | language option for config files. | 131 |
| \LWR@filestart: Adds HTML meta | | File lwarpmk.conf: Add: | |
| author. | 229 | language option for config files. | 131 |
| \LWR@forcenewpage: Forces new | | v0.30 | |
| PDF page before major | | \CSSFilename: Renamed from | |
| environments. | 191 | \NewCSS. | 208 |
| \LWR@htmlcomment: Breaks ligatures | | \HTMLAuthor: Renamed from | |
| in HTML comments. | 197 | \HTMLauthor. | 209 |
| \LWR@includegraphicsb: Adapts to | | \HTMLDescription: Renamed from | |
| graphics syntax. | 347 | \NewHTMLdescription. | 209 |
| \LWR@newhtmlfile: If FormatEPUB | | \HTMLFirstPageTop: Renamed from | |
| or FormatWordProcessor: skips | | \SetFirstPageTop. | 207 |
| headers, footers, nav. | 216 | \HTMLLanguage: Renamed from | |
| \LWR@parsetablecols: Added L, C, | | \MetaLanguage. | 228 |
| R, J column types. | 272 | \HTMLPageBottom: Renamed from | |
| \LWR@startref: Removed space. . . | 297 | \SetPageBottom. | 208 |
| \chapter: If EPUB, prints footnotes | | \HTMLPageTop: Renamed from | |
| before each section. | 227 | \SetPageTop. | 207 |
| \hyperindexref: Improved | | General: 2017/04/29 | 1 |
| indexing. | 316 | lwarp-newproject removed, and | |
| \textup: Fixed span class. | 371 | combined with lwarp. | 130 |
| General: 2017/04/14 | 1 | lwarpmk: Add: xdyfile | |
| glossaries: Added. | 448 | configuration option. | 170 |
| graphics: Added. | 450 | lwarpmk: Fix: xindy and texindy | |
| tabularx: Fix for optional pos. . . | 532 | adjusted for pdflatex, xelatex | |
| tabulary: Added. | 533 | and lualatex. | 170 |
| lwarpmk: Add: printglossary | | lwarpmk: Fix: xindy now used for | |
| and htmlglossary commands. | 170 | print index generation with | |
| Added boolean FormatEPUB. | 126 | latexmk. | 170 |
| Added boolean | | lwarpmk: language now used for | |
| FormatWordProcessor. | 126 | both index and glossary | |
| Added boolean | | generation. | 170 |
| HTMLDebugComments. | 125 | File: lwarp_html.xdy renamed to | |
| Added boolean HTMLMarkFloats | 126 | lwarp.xdy. | 168 |
| Docs: Modfying lwarpmk and | | Fix: *.css files only written in | |
| index processing. | 96 | print mode. | 132 |
| File lwarp_mathjax.txt: | | Fix: lwarp.xdy only written in | |
| Updated CDN repository. | 168 | print mode. | 168 |
| Forced oneside to maintain large | | Fix: lwarp_mathjax.txt only | |
| right margin. | 116 | written in print mode. | 168 |
| v0.29 | | Option OSWindows replaces macro | |
| \LWR@includegraphicsb: Fix: Error | | \warpOSWindows. | 109 |
| when no optional arguments. . . | 347 | Option latexmk replaces macro | |
| General: 2017/04/15 | 1 | \UseLatexmk. | 109 |
| lwarpmk: Add: language option | | Option lwarpmklang changed to | |
| for config files. | 170 | IndexLanguage. | 108 |
| Add: lwarpmklang option for | | Option xdyFilename added. | 108 |
| lwarp. | 108 | | |

| | |
|---|----------|
| Options HomeHTMLFilename and HTMLFilename replace macros | |
| \HomeHTMLFilename and \HTMLFilename. | 109 |
| v0.31 | |
| General: 2017/05/15 | 1 |
| keyfloat: Improved compatibility. | 458 |
| v0.32 | |
| \RequirePackage: Fix: Ignores blanks in package list. | 121 |
| General: 2016/06/09 | 1 |
| glossaries: Prevent error with \glo@name not defined. | 317 |
| lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes. | 170 |
| v0.33 | |
| \HTMLAuthor: Fix: Provides empty default author if none given. . . | 209 |
| \LWR@loadbefore: Fix: No \PackageError if already loaded. | 111 |
| \LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . . | 266 |
| \LWR@tabledatasinglecolumnstag: Fix: Macros in tabular could cause extra data cell. | 275 |
| \LWR@vspace: Add: \vspace nullified. | 379 |
| \StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. . . . | 263 |
| General: 2017/07/10 | 1 |
| amsmath: Removed fleqn option. | 119 |
| fancyhdr: Fix: Optional args for \thead, etc. | 418 |
| Add: Tabular at and bang columns now have their own HTML columns. | 260 |
| cleveref: Fix: Loaded \AtEndPreamble. | 358 |
| Fix: Incorrectly-inline math environments. | 328 |
| New handling of & to localize catcode changes. | 260 |
| v0.34 | |
| \@fnsymbol: Text symbols instead of math. | 247 |
| \InlineClass: Moved optional argument in front of mandatory. | 200 |
| \LWR@htmldivclass: Moved optional argument in front of mandatory. | 198 |
| \LWR@htmlclass: Moved optional argument in front of mandatory. | 198 |
| \LWR@htmlspanclass: Moved optional argument in front of mandatory. | 195 |
| \LWR@nullfonts: Improved font control. | 372 |
| \LWR@restoreorigformatting: booktabs: Works inside lateximage. | 323 |
| Improved font control. | 323 |
| \LWR@subhtmlclass: Moved optional argument in front of mandatory. | 197 |
| \LWR@tabledatacolumnstag: booktabs: Works inside lateximage. | 287 |
| \fbboxBlock: Added. | 367 |
| \makebox: Fix: Handles paren arg. . | 365 |
| General: 2017/08/08 | 1 |
| babel-french: Adds fixed-width HTML spaces to punctuation. . | 193 |
| balance: Added. | 395 |
| booktabs: Works inside lateximage. | 289, 396 |
| boxedminipage2e: Added. | 398 |
| boxedminipage: Prevented. | 398 |
| crop: Added. | 402 |
| enumerate: Added. | 408 |
| enumitem: Added, no longer required. | 409 |
| everyshi: Added. | 413 |
| fancybox: Added. | 415 |
| fancyvrb: Added, no longer required. | 419 |
| figcaps: Added. | 425 |
| filecontents: Required. Patched for morewrites. | 117 |
| floatpag: Added. | 429 |
| flushend: Added. | 436 |
| fullpage: Added. | 446 |
| hyperxmp: Added. | 455 |
| idxlayout: Added. | 456 |

| | | | |
|-------------------------------------|-----|--|-----|
| marginfit: Added. | 470 | \LWR@closeparagraph: Extra HTML | |
| mdframed: Improved mdtheorem | | source space after paragraphs. | 203 |
| patch. | 478 | \LWR@currenttextcolor: Fix for | |
| moreverb: Added. | 483 | \rule when xcolor not loaded. | 379 |
| paralist: Added. | 504 | \LWR@footnotetext: Extra HTML | |
| pdflscape: Added. | 505 | source space after paragraphs. | 211 |
| pdfsync: Added. | 506 | Force HTML superscripts. | 211 |
| prelim2e: Added. | 507 | \LWR@nullfonts: Fix: Filenames | |
| rotfloat: Added. | 514 | while using MathJax. | 372 |
| savetrees: Added. | 515 | \LWR@restoreorigformatting: | |
| shadow: Added. | 517 | siunitx: Improved | |
| syntonly: Added. | 531 | super/subscripts in a | |
| titles: No longer required. | 544 | lateximage. | 323 |
| titleref: Prevented. | 545 | \LWR@section: Improved spacing. | 221 |
| xmpincl: Added. | 580 | \LWR@stoppars: Extra HTML source | |
| Docs: Horizontal space | | space after paragraphs. | 205 |
| limitations. | 1 | \fbox: Fix: Uses \fboxrule and | |
| Docs: Misplaced alignment | | \fboxsep. | 367 |
| character. | 98 | \framebox: Fix: Handles width and | |
| File: README.txt: updated. | 1 | horiz position. | 366 |
| File: lwarp_mathjax.txt version | | \makebox: Fix: Handles width and | |
| change. | 168 | horiz position. | 365 |
| Fix: Added the eqnarray | | General: 2017/08/17 | 1 |
| environments. | 328 | babel-french: Adjustements for | |
| Improved font control. | 370 | French variants, load order, | |
| Lists refactored to remove | | footnotes, ellipses. | 193 |
| enumitem requirement. | 256 | footnote: Extra HTML source space | |
| Verbatim refactored to remove | | after paragraphs. | 441 |
| fancyvrb requirement. | 252 | siunitx: Fix for babel-french. | 342 |
| lateximage: Fix: lateximage with | | siunitx: Improved symbol support. | 522 |
| minipage, \parbox, \makebox, | | transparent: Added. | 556 |
| \fbox, \framebox, \raisebox, | | upref: Added. | 561 |
| \scalebox, \reflectbox. | 337 | xcolor: Added \fcolorboxBlock, | |
| fcolorminipage: Added. | 573 | \colorboxBlock. | 567 |
| BlockClass: Moved optional | | xcolor: Fix: Background none in | |
| argument in front of mandatory. | 199 | print mode. | 567 |
| fminipage: Added. | 367 | xcolor: Refactored | |
| LWR@nestspan: Fix: Minipages, | | \LWR@colorstyle. | 570 |
| BlocksClass, and lists inside a | | xcolor: Uses \fboxrule and | |
| span. | 194 | \fboxsep. | 567 |
| LWR@tabular: booktabs: Works | | xcolor: \fcolorbox etc. now work | |
| inside lateximage. | 289 | inside lateximage. | 567 |
| v0.35 | | Docs: Reorganized: Special cases | |
| General: 2017/08/08 | 1 | and limitations. | 71 |
| Fix: \textbf and related. | 370 | Source: Improved formatting. | 1 |
| v0.36 | | lateximage: Footnotes appear in | |
| \LWR@HTMLsanitize: Fix for | | regular text instead of the | |
| babel-french. | 336 | lateximage minipage. | 337 |
| \LWR@HTMLsanitizeexpand: Fix for | | LWR@tabular: Fix for babel-french. | 289 |
| babel-french. | 337 | | |

| | | | |
|--|-----|---|-----|
| v0.37 | | arabicfront: Added. | 395 |
| \include: Maintains independent | | caption2: Prevented. | 399 |
| . aux files for HTML. | 123 | chappg: Added. | 401 |
| General: 2017/08/19 | 1 | color: Forces xcolor as well. | 402 |
| \LaTeX accents: Added. | 128 | fix2col: Added. | 425 |
| babel-french: Adjustment for load | | fncychap: Added. | 437 |
| order. | 193 | grffile: Added. | 451 |
| color: Prevented. | 402 | metalogo: Added. | 481 |
| comment: Maintains independent | | nonumonpart: Added. | 491 |
| cutfiles for print, HTML. | 110 | nopageno: Added. | 491 |
| siunitx: Improved symbol support. | 522 | pagenote: Option page disabled. | 504 |
| textcomp: Improved support. | 534 | realscripts: Added. | 509 |
| lwarpmk: Removes additional | | relsize: Added. | 510 |
| HTML aux files. | 170 | romanbarpagenumber: Added. | 512 |
| File handles reorganized. | 122 | romanbar: Added. | 512 |
| v0.38 | | scalefnt: Added. | 515 |
| \@secntformat: Added for | | siunitx: Removed from lwrap core. | 522 |
| appendix. | 221 | textcomp: Removed from lwrap | |
| \ForceHTMLPage: Added. | 219 | core. | 534 |
| \ForceHTMLTOC: Added. | 219 | tocbibind: Added. | 550 |
| \LWR@section: \part* starts a new | | xltxtra: Added. | 579 |
| HTML page, for appendix. | 221 | lwarpmk: Added printl and html | |
| Modified spacing, uses | | actions. | 170 |
| \numberline. | 221 | Added \markboth, \sloppy, etc. | 191 |
| \numberline: Added trailing \quad. | 312 | Docs: Enhanced <i>Supported</i> | |
| \part: Fix with article class. | 226 | <i>Functions</i> table. | 32 |
| General: 2017/08/27 | 1 | Docs: Index, tocbibind. | 80 |
| appendix: Added. | 394 | Docs: Starred sections. | 78 |

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 | |
|------------------------------|------------------|
| \\$ | <i>319</i> |
| \& | <i>186, 4926</i> |
| \(. | <u>6050</u> |
| \) | <u>6050</u> |
| \, | <i>72</i> |
| \@@setcpageref | <u>7012</u> |
| \@@setcref | <u>7008</u> |
| \@@setcrefrange | <u>7009</u> |
| \@author | <i>237</i> |
| \@begintheorem | <u>4774</u> |
| \@capttype | <u>5739</u> |
| \@currentlabelname | <u>5519</u> |
| \@date | <i>237</i> |
| \@dlbfloat | <u>5717</u> |
| \@endtheorem | <u>4784</u> |
| \@ensuredmath | <u>6057</u> |
| \@float | <u>5717</u> |
| \@fnsymbol | <u>4641</u> |
| \@footnotetext | <u>3768</u> |
| \@include | <u>423</u> |
| \@makecaption | <u>5741</u> |

| | | | |
|--------------------------------------|------------------|---------------------------|---------|
| \@makefnmark | 3740 | autosec | 220 |
| \@makefntext | 3739 | | |
| \@maketitle | 4653 | | |
| \@mpfootnotetext | 3769 | B | |
| \@opargbegintheorem | 4779 | babel (package) | 90 |
| \@published | 237 | \backmatter | 3975 |
| \@secntformat | 3991 | balance (package) | 395 |
| \@subtitle | 237 | \BaseJobname | 3189 |
| \@title | 237 | BaseJobname (option) | 108 |
| \@wrglossary | 5991 | baseline | |
| \@wrindex | 5984 | tabular | 271 |
| \\ | 376 | \bfseries | 7326 |
| \$ | 320 | \BibTeX | 7517 |
| \$\$ | 320 | BlockClass (environment) | 3514 |
| ~ | 71 | \BlockClassSingle | 3526 |
| | | bookmark (package) | 396 |
| | | booktabs (package) | 396 |
| | | booleans: | |
| A | | CombineHigherDepths | 56, 214 |
| abstract (environment) | 4682 | FileSectionNames | 57, 187 |
| abstract (package) | 386 | FormatEpub | 91, 125 |
| \abstractname | 57, 4681 | FormatWordProcessor | 93, 126 |
| accents | 219 | HTMLDebugComments | 125 |
| \addcontentsline | 5793 | HTMLMarkFloats | 94, 126 |
| Adobe (program) | 38 | LWR@amsmultline | 327 |
| \affiliation | 4347 | LWR@doingapar | 201 |
| afterpage (package) | 388 | LWR@doinghline | 261 |
| algorithmic | | LWR@doingstartpars | 201 |
| with newfloat, trivfloat | 557 | LWR@doingtbrule | 261 |
| algorithmicx (package) | 389 | LWR@freezethisfloat | 304 |
| align (environment) | 6376 | LWR@infloatrow | 347 |
| align* (environment) | 6399 | LWR@intabularmetadate | 261 |
| alignment tab character &, misplaced | 86, 259, 263 | LWR@minipagefullwidth | 362 |
| alltt (package) | 390 | LWR@minipagethispar | 363 |
| \AmS | 7522 | LWR@skipatbang | 261 |
| AMSMath | | LWR@skippingmrowcell | 261 |
| split miss-numbered | 81, 83, 317, 493 | LWR@starredlongtable | 281 |
| amsmath (package) | 119 | LWR@startedrow | 261 |
| amsthm (package) | 391 | LWR@tableparcell | 261 |
| \and | 237 | LWR@tracinglwarp | 125 |
| appendix (package) | 394 | LWR@validtablecol | 270 |
| arabicfront (package) | 395 | LWR@verbtags | 252 |
| array (package) | 292 | mathjax | 107 |
| AsciiDoc (program) | 37 | usingOSWindows | 106 |
| AsciiDoctor (program) | 37 | warpingHTML | 107 |
| Asciidoctor-LaTeX (program) | 37 | warpingprint | 107 |
| \attrib | 89, 251, 562 | boxedminipage (package) | 398 |
| \attribution | 4692 | boxedminipage2e (package) | 398 |
| author | | \boxframe | 138 |
| HTML meta tag | 70, 209 | bugs | 98 |
| \author | 69, 237, 4354 | BVerbatim (environment) | 183 |

| C | |
|---|------------------|
| calc (package) | 118 |
| Calibre | 91 |
| caption (package) | 118 |
| caption2 (package) | 399 |
| \caption@begin | 5772 |
| \caption@end | 5772 |
| \captionlistentry | 5776 |
| \captionof | 5806 |
| \CaptionSeparator | 5740 |
| ccaption (package) | 400 |
| center (environment) | 6611 |
| \centering | 232 |
| changepage (package) | 400 |
| chappg (package) | 401 |
| \chapter | 4134 |
| \citetitle | 4699 |
| classes: | |
| internet | 36 |
| cleveref (package) | 358 |
| cmap (package) | 65 |
| \color | 571 |
| color (package) | 402 |
| \colorbox | 46 |
| \colorboxBlock | 56, 6908 |
| CombineHigherDepths (boolean) | 56, 214 |
| comment (package) | 110 |
| Computer Modern | 64 |
| \ConTeXt | 7514 |
| counters: | |
| FileDepth | 56, 214 |
| lofdepth | 313 |
| lotdepth | 313 |
| LWR@externalfilecnt | 319 |
| LWR@htmlfilenumber | 187 |
| LWR@latestautopage | 305 |
| LWR@lateximagedepth | 336 |
| LWR@lateximagenumber | 336 |
| LWR@Lpage | 336 |
| LWR@midrulecounter | 277 |
| LWR@minipagedepth | 361 |
| LWR@nextautofloat | 305 |
| LWR@nextautopage | 305 |
| LWR@nextequation | 321 |
| LWR@prevFileDepth | 221 |
| LWR@spandepth | 201 |
| LWR@tablecolspos | 266 |
| LWR@tablecolswidth | 266 |
| LWR@tabletotalcols | 266 |
| LWR@tabletotalcolsnxt | 266 |
| LWR@thisfloat | 304 |
| SideTOCDepth | 56, 311 |
| tocdepth | 55 |
| \cpagerefFor | 7011 |
| crop (package) | 402 |
| cross-references | |
| missing or incorrect | 98 |
| CSS | |
| file selection | 70 |
| lwrap.css | 70 |
| per HTML page | 70 |
| project-specific changes | 70 |
| \CSSFilename | 54, 57, 70, 3710 |
| cutwin (package) | 403 |
| D | |
| danger icon | 103 |
| \date | 69 |
| dcolumn (package) | 404 |
| debugging | 98 |
| HTML debug comments | 125 |
| \DeclareGraphicsExtensions | 6642 |
| \DeclareGraphicsRule | 6642 |
| Deja Vu | 64 |
| description | |
| HTML meta tag | 69, 209 |
| description (environment) | 4838 |
| displaymath (environment) | 6068 |
| \dotfill | 7342 |
| draftwatermark (package) | 405 |
| E | |
| ellipsis (package) | 405 |
| \emph | 7250 |
| emptypage (package) | 406 |
| \end@dlbfloat | 5731 |
| \end@float | 5731 |
| \EndDefiningTabulars | 4921 |
| endnotes (package) | 407 |
| \enlargethispage | 7445 |
| \enskip | 376, 7406 |
| enumerate (environment) | 4810 |
| enumerate (package) | 408 |
| enumitem (package) | 409 |
| environ (package) | 119 |
| environments: | |
| abstract | 4682 |
| align | 6376 |
| align* | 6399 |
| BlockClass | 3514 |

| | | | |
|--------------------------|------------------|------------------------------|----------------|
| BVerbatim | 183 | everyshi (package) | 413 |
| center | 6611 | expl3 (package) | 117 |
| description | 4838 | extramarks (package) | 413 |
| displaymath | 6068 | | |
| enumerate | 4810 | F | |
| eqnarray | 6232 | fancybox (package) | 414 |
| equation | 6183 | fancyhdr (package) | 418 |
| fcolorminipage | 114, 6950 | fancyvrb (package) | 419 |
| flalign | 6422 | FAQ | 98 |
| flalign* | 6445 | fbox | |
| flushleft | 6623 | around a minipage | 74 |
| flushright | 6617 | \fbbox | 74, 7181 |
| fminipage | 74, 7198, 7246 | \fbboxBlock | 74, 7190, 7245 |
| gather | 6330 | \fcolorbox | 68, 6914 |
| gather* | 6353 | \fcolorboxBlock | 88, 6949 |
| itemize | 4796 | fcolorminipage (environment) | 114, 6950 |
| lateximage | 6521 | figcaps (package) | 425 |
| longtable | 2 | file | |
| LWR@nestspan | 3362 | inaccessible | 56 |
| LWR@tabular | 5456 | underscore | 55, 72 |
| LWRprint@fminipage | 7222 | filecontents (package) | 117 |
| math | 6067 | FileDepth (counter) | 56, 214 |
| minipage | 7074 | files: | |
| multiline | 6281 | glyphtounicode | 64 |
| multiline* | 6305 | lwarpcss | 70, 132 |
| picture | 360, 7029 | lwarpcxy | 70, 168 |
| quote | 4707 | lwarpcformalcss | 163 |
| theindex | 5962 | lwarpcmathjaxtxt | 168 |
| tikzpicture | 8 | lwarpcmagebrushcss | 158 |
| titlepage | 68, 4388 | lwarpcutorialtxt | 46 |
| titlingpage | 68, 4394, 4407 | lwarpcmkconf | 130 |
| Verbatim | 13 | lwarpcmklua | 96 |
| verbatim | 4764 | projectcss | 70 |
| verse | 2 | projectlwarpcmkconf | 131 |
| warpall | 67, 100 | projecthtmltex | 130 |
| warpHTML | 50, 59, 67, 101 | sampleprojectcss | 70, 167 |
| warpprint | 58, 67, 101 | tutorialtex | 46 |
| epigraph (package) | 410 | FileSectionNames (boolean) | 57, 187 |
| Epub | | fix2col (package) | 425 |
| conversion software | 91 | \flagverse | 563 |
| HTML conversion settings | 91, 126 | flalign (environment) | 6422 |
| eqnarray (environment) | 6232 | flalign* (environment) | 6445 |
| equation | | Flare (program) | 38 |
| miss-numbered | 81, 83, 317, 493 | float (package) | 426 |
| equation (environment) | 6183 | floatflt (package) | 428 |
| error messages | 98 | \floatname | 9 |
| eso-pic (package) | 411 | floatpag (package) | 429 |
| etoolbox (package) | 105 | \floatplacement | 12 |
| everyhook (package) | 117 | floatrow (package) | 430 |
| everypage (package) | 412 | \floatstyle | 15 |

| | | | |
|---|----------------|---|---------------|
| <code>\flushbottom</code> | 3283 | <code>glossaries</code> (package) | 79, 448 |
| <code>flushend</code> (package) | 436 | <code>glossary</code> | |
| <code>flushleft</code> (environment) | 6623 | language | 62 |
| <code>flushright</code> (environment) | 6617 | processing | 62 |
| <code>fminipage</code> (environment) . | 74, 7198, 7246 | <code>glyptounicode</code> (file) | 64 |
| <code>fncychap</code> (package) | 437 | <code>graphics</code> (package) | 343, 450 |
| <code>font</code> | | <code>\graphicspath</code> | 6640 |
| <code>bitmapped</code> | 64 | <code>graphicx</code> (package) | 343, 450 |
| Computer Modern | 64 | <code>grffile</code> (package) | 451 |
| Deja Vu | 64 | | |
| ligatures | 64, 65 | H | |
| packages | 64 | <code>Hevea</code> (program) | 36 |
| selection | 64 | <code>\hfill</code> | 7340 |
| size — <code>lateximage</code> | 82, 317, 335 | <code>\HomeHTMLFilename</code> | 3191 |
| size — <code>math, SVG</code> | 82, 317, 335 | <code>HomeHTMLFilename</code> (option) | 55, 109 |
| size — <code>xfrac</code> | 576 | <code>horizontal space</code> | 71 |
| <code>fontenc</code> (package) | 64, 438 | between minipages | 376 |
| <code>fontspec</code> (package) | 64, 438 | <code>\href</code> | 5662 |
| <code>footmisc</code> (package) | 439 | <code>\hrulefill</code> | 7341 |
| <code>footnote</code> (package) | 441 | <code>\hskip</code> | 72 |
| <code>footnotehyper</code> (package) | 442 | <code>\hspace</code> | 71, 376, 7436 |
| <code>footnotes</code> | 209 | <code>HTML</code> | |
| <code>for</code> | | commands for a successful conver- | |
| HTML & PRINT | 103 | sion | 71 |
| HTML output | 103 | conversion settings | 55 |
| PRINT output | 103 | debug comments | 125 |
| <code>\ForceHTMLPage</code> | 78, 3950 | EPUB | 91, 126 |
| <code>\ForceHTMLTOC</code> | 79, 3956 | word processor | 93, 94, 126 |
| <code>FormatEPUB</code> (boolean) | 91, 125 | filename generation | 57 |
| <code>FormatWordProcessor</code> (boolean) . | 93, 126 | headings | 101 |
| <code>\framebox</code> | 7160 | in place of SVG images | 52, 99 |
| <code>framed</code> (package) | 443 | meta tag | |
| <code>framed objects</code> | 74 | author | 70, 209 |
| <code>FrameMaker</code> (program) | 38 | description | 69, 209 |
| <code>Frequently Asked Questions</code> | 98 | generator | 229 |
| <code>\frontmatter</code> | 3972 | viewport | 230 |
| <code>ftnright</code> (package) | 446 | tabular column conversion | 271 |
| <code>fullpage</code> (package) | 446 | <code>\HTMLAuthor</code> | 58, 70, 3724 |
| <code>\fup</code> | 7339 | <code>HTMLDebugComments</code> (boolean) | 125 |
| <code>\fussy</code> | 3285 | <code>\HTMLDescription</code> | 58, 69, 3730 |
| G | | <code>\HTMLentity</code> | 3172 |
| <code>gather</code> (environment) | 6330 | <code>\HTMLFilename</code> | 3190 |
| <code>gather*</code> (environment) | 6353 | <code>HTMLFilename</code> (option) | 55, 109 |
| <code>GELLMU</code> (program) | 36 | <code>\HTMLFirstPageTop</code> | 58, 3687 |
| <code>generator</code> | | <code>htmlglossary</code> (option) [lwarpmk] . | 80, 448 |
| HTML meta tag | 229 | <code>\HTMLLanguage</code> | 58, 4190 |
| <code>geometry</code> (package) | 116, 447 | <code>HTMLleftmargini</code> (length) 89, 251, 252, 562 | |
| <code>gettittlestring</code> (package) | 117 | <code>HTMLMarkFloats</code> (boolean) | 94, 126 |
| <code>GladTeX</code> (program) | 37 | <code>\HTMLPageBottom</code> | 58, 237, 3703 |
| | | <code>\HTMLPageTop</code> | 58, 3695 |

| | | | |
|--|-------------------|---|-------------------|
| <code>\HTMLUnicode</code> | 3180 | index | 62 |
| <code>HTMLvleftskip</code> (length) . | 89, 251, 252, 562 | language HTML metadata | 228 |
| <code>\hyperindexref</code> | 5998 | <code>\LaTeX</code> | 7500 |
| <code>hyperref</code> (package) | 299, 452 | <code>LaTeX2HTML</code> (program) | 36 |
| <code>\hypertoc</code> | 5908 | <code>\LaTeXe</code> | 7500 |
| <code>\hypertocfloat</code> | 5921 | <code>lateximage</code> (environment) | 6521 |
| <code>hyperxmp</code> (package) | 455 | <code>\LateximageFontSizeName</code> | 6479 |
| I | | | |
| icon | | <code>lateximages</code> | |
| warning | 103 | font size | 82, 317, 335 |
| <code>idxlayout</code> (package) | 456 | <code>latexmk</code> (option) | 55, 109 |
| <code>ifplatform</code> (package) | 105 | <code>LaTeXML</code> (program) | 36 |
| images | | <code>layout</code> (package) | 460 |
| <code>graphicx</code> package | 343 | lengths: | |
| in strange places | 99 | <code>HTMLleftmargini</code> .. | 89, 251, 252, 562 |
| showing as HTML | 52, 99 | <code>HTMLvleftskip</code> | 89, 251, 252, 562 |
| <code>\include</code> | 423 | <code>LWR@minipageheight</code> | 361 |
| <code>\includegraphics</code> | 6832 | <code>LWR@minipagewidth</code> | 361 |
| <code>indentfirst</code> (package) | 457 | <code>LWR@tempheight</code> | 377 |
| <code>InDesign</code> (program) | 38 | <code>LWR@tempraise</code> | 377 |
| index | | <code>LWR@tempwidth</code> | 377 |
| language | 62 | <code>\VerbatimHTMLWidth</code> | 252 |
| placement and toc options | 80 | <code>vleftmargini</code> | 89, 251, 562 |
| processing | 50, 51 | <code>vleftskip</code> | 89, 251, 562, 563 |
| UTF-8 | 65 | <code>letltxmacro</code> (package) | 105 |
| <code>IndexLanguage</code> (option) . | 62, 80, 108, 448 | <code>letterspace</code> (package) | 460 |
| <code>\InlineClass</code> | 3534 | <code>lettrine</code> (package) | 461 |
| <code>inputenc</code> (package) | 64, 457 | <code>LibreOffice</code> (program) | 37 |
| <code>internet</code> (class) | 36 | ligatures | 64, 65, 116 |
| item | | line numbers | 103 |
| empty | 72, 255 | <code>\linebreak</code> | 7439 |
| <code>itemize</code> (environment) | 4796 | <code>\LinkHome</code> | 3241 |
| <code>\itshape</code> | 7331 | <code>Linux</code> (program) | 66, 106 |
| J | | | |
| JavaScript | | <code>lips</code> (package) | 462 |
| <code>MathJax</code> | 81, 317 | list item, empty | 72, 255 |
| K | | | |
| <code>\kern</code> | 72 | <code>listings</code> (package) | 463 |
| <code>keyfloat</code> (package) | 458 | <code>\listof</code> | 5874 |
| <code>kvoptions</code> (package) | 107 | <code>lmodern</code> (package) | 64 |
| L | | | |
| label | | <code>lofdepth</code> (counter) | 313 |
| in HTML | 232 | <code>longtable</code> (environment) | 2 |
| math environment | 327 | <code>longtable</code> (package) | 467 |
| language | | <code>lotdepth</code> (counter) | 313 |
| glossary | 62 | <code>lscape</code> (package) | 469 |
| | | <code>lrcaption</code> (package) | 469 |
| | | <code>LuaLaTeX</code> | |
| | | detection | 115 |
| | | file & section names | 219 |
| | | <code>\LuaLaTeX</code> | 7508 |
| | | <code>LuaLaTeX</code> (program) [requirement] . . . | 40 |
| | | <code>\LuaTeX</code> | 7508 |

| | | | |
|-------------------------------------|---------|--|------------------|
| lwrap | | \backslash LWR@doubledollar | 6024 |
| loading | 65 | \backslash LWR@endhideamsmath | 6225 |
| options | 65 | \backslash LWR@endminipage | 7038 |
| lwrap (package) | 65 | \backslash LWR@endofline | 7389 |
| lwrap.css (file) | 70, 132 | \backslash LWR@endsubminipage | 7059 |
| lwrap.xdy | | \backslash LWR@ensuredoingapar | 3575 |
| customizing | 70 | \backslash LWR@epubprintpendingfootnotes | 3800 |
| lwrap.xdy (file) | 70, 168 | LWR@externalfilecnt (counter) | 319 |
| lwrap_formal.css (file) | 163 | \backslash LWR@filenamenoblanks | 3820 |
| lwrap_mathjax.txt (file) | 168 | \backslash LWR@filestart | 4197 |
| lwrap_sagebrush.css (file) | 158 | \backslash LWR@findword | 352 |
| lwrap_tutorial.txt (file) | 46 | \backslash LWR@floatbegin | 5701 |
| lwrapmk | | \backslash LWR@floatend | 5719 |
| customizing | 96 | \backslash LWR@footnotetext | 3741 |
| [lwrapmk]: | | \backslash LWR@forceminwidth | 7165 |
| htmlglossary (option) | 80, 448 | \backslash LWR@forcenewpage | 3275 |
| printglossary (option) | 80, 448 | LWR@freezethisfloat (boolean) | 304 |
| lwrapmk (option) | 108 | \backslash LWR@futurenonspacilet | 4864 |
| lwrapmk (program) | 96, 170 | \backslash LWR@FVstyle | 17 |
| lwrapmk.conf (file) | 130 | \backslash LWR@getexparray | 3168 |
| lwrapmk.lua (file) | 96 | \backslash LWR@getmynexttoken | 4871 |
| \backslash LWR@addmathjax | 6172 | \backslash LWR@hidelatexequation | 6158 |
| \backslash LWR@afterendverbatim | 4748 | \backslash LWR@hspace | 7414 |
| LWR@amsmultline (boolean) | 327 | \backslash LWR@htmlblockcomment | 3456 |
| \backslash LWR@atbeginverbatim | 4734 | \backslash LWR@htmlblocktag | 3464 |
| \backslash LWR@beginhideamsmath | 6217 | \backslash LWR@htmlclosecomment | 3428 |
| \backslash LWR@blackborderpadding | 7176 | \backslash LWR@htmlcomment | 3449 |
| \backslash LWR@borderpadding | 29 | \backslash LWR@htmldivclass | 3491 |
| \backslash LWR@botnavigation | 3249 | \backslash LWR@htmldivclassend | 3494 |
| \backslash LWR@caption@begin | 5753 | \backslash LWR@html element | 3504 |
| \backslash LWR@caption@end | 5763 | \backslash LWR@html element class | 3478 |
| \backslash LWR@clearmidrules | 5180 | \backslash LWR@html element class end | 3483 |
| \backslash LWR@closeparagraph | 3602 | \backslash LWR@html element class line | 3497 |
| \backslash LWR@closeprevious | 3269 | \backslash LWR@html element end | 3507 |
| \backslash LWR@closetabledatacell | 4887 | LWR@htmlfilenumber (counter) | 187 |
| LWR@colafterspec | 266 | \backslash LWR@htmlfileref | 5545 |
| LWR@colatspec | 266 | \backslash LWR@htmlmathlabel | 6203 |
| LWR@colbangspec | 266 | \backslash LWR@htmlopencomment | 3428 |
| LWR@colbeforespec | 266 | \backslash LWR@htmlrefsectionfilename | 3234 |
| \backslash LWR@copyfile | 458 | \backslash LWR@HTMLsanitize | 6482 |
| \backslash LWR@createautosec | 3981 | \backslash LWR@HTMLsanitizeexpand | 6503 |
| \backslash LWR@currentcss | 3709 | \backslash LWR@htmlsectionfilename | 3201 |
| \backslash LWR@currenttextcolor | 7446 | \backslash LWR@htmlspan | 3403 |
| \backslash LWR@descitem | 4824 | \backslash LWR@htmlspanclass | 3411 |
| \backslash LWR@docmidrule | 5199 | \backslash LWR@htmltag | 3419 |
| LWR@doingapar (boolean) | 201 | \backslash LWR@htmltagc | 3352 |
| LWR@doinghline (boolean) | 261 | \backslash LWR@includegraphicsb | 6734 |
| LWR@doingstartpars (boolean) | 201 | \backslash LWR@indexitem | 5972, 5976, 5980 |
| LWR@doingtbrule (boolean) | 261 | LWR@infloatrow (boolean) | 347 |
| \backslash LWR@domulticolumn | 5273 | LWR@intabularmetadata (boolean) | 261 |

| | | | |
|---|------------|--|------|
| <code>\LWR@itemizeitem</code> | 4789 | <code>\LWR@parseDcolumn</code> | 5044 |
| <code>\LWR@latestautopage</code> (counter) | 305 | <code>\LWR@parsenormalcolumn</code> | 5029 |
| <code>\LWR@lateximagedepth</code> (counter) | 336 | <code>\LWR@parsepcolumn</code> | 5040 |
| <code>\LWR@lateximagedepthref</code> | 5549 | <code>\LWR@parseskipcolumn</code> | 5026 |
| <code>\LWR@lateximagenumber</code> (counter) .. | 336 | <code>\LWR@parsetablecols</code> | 5049 |
| <code>\LWR@lateximagenumberref</code> | 5552 | <code>\LWR@patchlists</code> | 4852 |
| <code>\LWR@lateximagesfile</code> | 420 | <code>\LWR@prevFileDepth</code> (counter) | 221 |
| <code>\LWR@Llpage</code> (counter) | 336 | <code>\LWR@printatbang</code> | 5126 |
| <code>\LWR@loadafter</code> | 111 | <code>\LWR@printmccoldata</code> | 5233 |
| <code>\LWR@loadbefore</code> | 122 | <code>\LWR@printmccoltype</code> | 5203 |
| <code>\LWR@loadnever</code> | 131 | <code>\LWR@printpendingfootnotes</code> | 3790 |
| <code>\LWR@longtabledatacaptiontag</code> ... | 5292 | <code>\LWR@printthetitle</code> | 4483 |
| <code>\LWR@lookforpackagename</code> | 355 | <code>\LWR@ProvidesPackageDrop</code> | 407 |
| <code>\LWR@LwarpEnd</code> | 4324, 7544 | <code>\LWR@ProvidesPackagePass</code> | 395 |
| <code>\LWR@LwarpStart</code> | 4247, 7544 | <code>\LWR@pushoneclose</code> | 3984 |
| <code>\LWR@maketitlesetup</code> | 4632 | <code>\LWR@quickfile</code> | 417 |
| <code>\LWR@maybeinthisfloat</code> | 5736 | <code>\LWR@requesttoc</code> | 4317 |
| <code>\LWR@maybenewtablerow</code> | 5110 | <code>\LWR@requirepackagenames</code> | 351 |
| <code>\LWR@midrulecounter</code> (counter) | 277 | <code>\LWR@restoremoreorigxcolor</code> | 6988 |
| <code>\LWR@midrules</code> | 276 | <code>\LWR@restoreorigaccents</code> | 600 |
| <code>\LWR@minipagedepth</code> (counter) | 361 | <code>\LWR@restoreorigformatting</code> | 6091 |
| <code>\LWR@minipagefullwidth</code> (boolean) .. | 362 | <code>\LWR@restoreorigrelsize</code> | 6090 |
| <code>\LWR@minipageheight</code> (length) | 361 | <code>\LWR@restoreorigtextcomp</code> | 47 |
| <code>\LWR@minipagestartpars</code> | 7394 | <code>\LWR@rotboxorigin</code> | 6843 |
| <code>\LWR@minipagestoppars</code> | 7397 | <code>\LWR@rotstyle</code> | 6723 |
| <code>\LWR@minipagethispar</code> (boolean) ... | 363 | <code>\LWR@rule</code> | 7447 |
| <code>\LWR@minipagewidth</code> (length) | 361 | <code>\LWR@scalestyle</code> | 6726 |
| <code>\LWR@multicolother</code> | 5224 | <code>\LWR@section</code> | 3992 |
| <code>\LWR@multicolparttext</code> | 5219 | <code>\LWR@sectionnumber</code> | 3978 |
| <code>\LWR@multicolskip</code> | 5230 | <code>\LWR@setexparray</code> | 3163 |
| <code>\LWR@mynexttoken</code> | 4863 | <code>\LWR@setlatestname</code> | 5521 |
| <code>\LWR@myshorttoc</code> | 5818 | <code>\LWR@setOSWindows</code> | 29 |
| <code>\LWR@nameref</code> | 5542 | <code>\LWR@sidetoc</code> | 5887 |
| <code>\LWR@nestspan</code> (environment) | 3362 | <code>\LWR@singledollar</code> | 6036 |
| <code>\LWR@newtmlfile</code> | 3886 | <code>\LWR@skipatbang</code> (boolean) | 261 |
| <code>\LWR@newlabel</code> | 5579 | <code>\LWR@skippingmrowcell</code> (boolean) ... | 261 |
| <code>\LWR@newref</code> | 5629 | <code>\LWR@spandepth</code> (counter) | 201 |
| <code>\LWR@nextautofloat</code> (counter) | 305 | <code>\LWR@splabel</code> | 5555 |
| <code>\LWR@nextautopage</code> (counter) | 305 | <code>\LWR@starredlongtable</code> (boolean) ... | 281 |
| <code>\LWR@nextequation</code> (counter) | 321 | <code>\LWR@startedrow</code> (boolean) | 261 |
| <code>\LWR@nohspace</code> | 7435 | <code>\LWR@startnewdepth</code> | 3985 |
| <code>\LWR@nullfonts</code> | 7294 | <code>\LWR@startpars</code> | 3639 |
| <code>\LWR@openparagraph</code> | 3580 | <code>\LWR@startref</code> | 5592 |
| <code>\LWR@origcolspec</code> | 4956 | <code>\LWR@stoppars</code> | 3655 |
| <code>\LWR@originname</code> | 6844 | <code>\LWR@stripperiod</code> | 5520 |
| <code>\LWR@originnames</code> | 6851 | <code>\LWR@strresult</code> | 4954 |
| <code>\LWR@parseaftercolumn</code> | 5016 | <code>\LWR@subcmidrule</code> | 5191 |
| <code>\LWR@parseatcolumn</code> | 4962 | <code>\LWR@subhtmelementclass</code> | 3469 |
| <code>\LWR@parsebangcolumn</code> | 4986 | <code>\LWR@subhyperref</code> | 5653 |
| <code>\LWR@parsebeforecolumn</code> | 5007 | <code>\LWR@subhyperrefclass</code> | 5657 |

| | | | |
|--|---------------|---|--------------|
| <code>\LWR@subinlineimage</code> | 5694 | <code>marginfix</code> (package) | 470 |
| <code>\LWR@subminipage</code> | 7053 | <code>marginnote</code> (package) | 471 |
| <code>\LWR@subnewref</code> | 5623 | <code>\marginpar</code> | 3807 |
| <code>\LWR@subsublabel</code> | 5558 | <code>\markboth</code> | 3280 |
| <code>\LWR@subtableofcontents</code> | 5833 | <code>\markright</code> | 3281 |
| <code>\LWR@syncmathjax</code> | 6070 | markup languages | 38 |
| <code>\LWR@tablecolspec</code> | 4953 | <code>math</code> | |
| <code>LWR@tablecolspos</code> (counter) | 266 | font size — SVG | 82, 317, 335 |
| <code>LWR@tablecolwidth</code> (counter) | 266 | <code>mathjax</code> option | 108 |
| <code>\LWR@tabledatacolumnntag</code> | 5399 | MathJax summary | 82, 318 |
| <code>\LWR@tabledatamulticolumnntag</code> | 5377 | <code>mathsvg</code> option | 108 |
| <code>\LWR@tabledatamultirowntag</code> | 5385 | showing as HTML | 52, 99 |
| <code>\LWR@tabledatasinglecolumnntag</code> | 5140 | SVG summary | 82, 317 |
| <code>LWR@tableparcell</code> (boolean) | 261 | <code>math</code> (environment) | 6067 |
| <code>LWR@tabletotalcols</code> (counter) | 266 | MathJax | |
| <code>LWR@tabletotalcolsnext</code> (counter) | 266 | <code>mathjax</code> option | 108 |
| <code>LWR@tabular</code> (environment) | 5456 | subequations | 83, 318 |
| <code>\LWR@tabularendofline</code> | 4937 | summary | 82, 318 |
| <code>\LWR@tabularhtmlcolumns</code> | 5358 | tagged equations | 83, 318 |
| <code>LWR@tempheight</code> (length) | 377 | MathJax (program) | 82, 318 |
| <code>LWR@tempraise</code> (length) | 377 | MathJax (program) [requirement] | 40 |
| <code>LWR@tempwidth</code> (length) | 377 | <code>mathjax</code> (boolean) | 107 |
| <code>\LWR@thisfilename</code> | 3818 | <code>mathjax</code> (option) | 55, 65, 108 |
| <code>LWR@thisfloat</code> (counter) | 304 | <code>mathsvg</code> (option) | 55, 65, 108 |
| <code>\LWR@thisnewfilename</code> | 3819 | <code>mcaption</code> (package) | 471 |
| <code>\LWR@titlingmaketitle</code> | 4670 | <code>mdframed</code> (package) | 472 |
| <code>\LWR@topnavigation</code> | 3246 | <code>\mdseries</code> | 7325 |
| <code>\LWR@traceinfo</code> | 479 | <code>memoir</code> | |
| <code>LWR@tracinglwarp</code> (boolean) | 125 | verse | 563 |
| <code>LWR@validtablecol</code> (boolean) | 270 | <code>memoir</code> (package) | 563 |
| <code>LWR@verbtags</code> (boolean) | 252 | meta tag, HTML | |
| <code>\LWR@vspace</code> | 7437 | author | 70, 209 |
| <code>LWR#print@fminipage</code> (environment) | 7222 | description | 69, 209 |
| <code>\LWRsetnextfloat</code> | 5744 | generator | 229 |
| <code>\LyX</code> | 7525 | viewport | 230 |
| | | <code>metalogo</code> (package) | 481 |
| | | <code>microtype</code> (package) | 65, 116, 482 |
| | | <code>\MiKTeX</code> | 7524 |
| M | | <code>minipage</code> | |
| <code>Mac OS</code> (program) | 66, 106 | framed | 74 |
| <code>Madcap</code> (program) | 38 | horizontal space between | 376 |
| <code>\mainmatter</code> | 3968 | <code>minipage</code> (environment) | 7074 |
| <code>\makebox</code> | 7137 | <code>\minipagefullwidth</code> | 7066 |
| <code>makeidx</code> (package) | 118 | misplaced alignment tab character & | 86, 259, 263 |
| <code>\MakeIndex</code> | 7517 | missing sections | 56 |
| <code>\maketitle</code> | 68, 244, 4645 | <code>moreverb</code> (package) | 483 |
| <code>\maketitlehookaa</code> | 236 | <code>mparhack</code> (package) | 485 |
| <code>\maketitlehookaaa</code> | 236 | <code>\mrowcell</code> | 5445 |
| <code>margin</code> | | MS-Windows (program) | 66, 106 |
| numbers | 103 | | |
| tags | 103 | | |
| <code>marginfit</code> (package) | 470 | | |

| | | | |
|---|------------------|------------------------------|--------------|
| multicol (package) | 486 | mathsvg | 55, 65, 108 |
| multitrow (package) | 286, 488 | OSWindows | 66, 106, 109 |
| multline (environment) | 6281 | warpHTML | 65, 107 |
| multline* (environment) | 6305 | warpprint | 65, 107 |
| | | xdyFilename | 71, 108 |
| N | | | |
| \Nameref | 5650 | \OSPathSymbol | 28 |
| \nameref | 5641 | OSWindows (option) | 66, 106, 109 |
| nameref (package) | 488 | P | |
| needspace (package) | 489 | packages | |
| newclue (package) | 89, 489 | required | 115 |
| newfloat | | packages: | |
| with trivfloat, algorithmic | 557 | abstract | 386 |
| \newfloat | 2, 4 | afterpage | 388 |
| newfloat (package) | 118 | algorithmicx | 389 |
| \newline | 7387 | alltt | 390 |
| \newtheorem | 254 | amsmath | 119 |
| newtxmath (package) | 84 | amsthm | 391 |
| newunicodechar (package) | 64, 490 | appendix | 394 |
| nextpage (package) | 490 | arabicfront | 395 |
| nicefrac (package) | 84 | array | 292 |
| \nolinebreak | 7440 | babel | 90 |
| \nolinkurl | 5674 | balance | 395 |
| nonumonpart (package) | 491 | bookmark | 396 |
| \nopagebreak | 7444 | booktabs | 396 |
| \nopagecolor | 45 | boxedminipage | 398 |
| nopageno (package) | 491 | boxedminipage2e | 398 |
| \normalfont | 7333 | calc | 118 |
| nowidow (package) | 492 | caption | 118 |
| ntheorem (package) | 493 | caption2 | 399 |
| \numberline | 5905 | ccaption | 400 |
| numbers | | change page | 400 |
| left margin | 103 | chappg | 401 |
| numindex (option) [tocbibind] | 81 | cleveref | 358 |
| O | | | |
| OpenOffice (program) | 37 | cmap | 65 |
| options: | | color | 402 |
| [lwarpmk]: | | comment | 110 |
| htmlglossary | 80, 448 | crop | 402 |
| printglossary | 80, 448 | cutwin | 403 |
| [tocbibind]: | | dcolumn | 404 |
| numindex | 81 | draftwatermark | 405 |
| BaseJobname | 108 | ellipsis | 405 |
| HomeHTMLFilename | 55, 109 | emptypage | 406 |
| HTMLFilename | 55, 109 | endnotes | 407 |
| IndexLanguage | 62, 80, 108, 448 | enumerate | 408 |
| latexmk | 55, 109 | enumitem | 409 |
| lwarpmk | 108 | environ | 119 |
| mathjax | 55, 65, 108 | epigraph | 410 |
| | | eso-pic | 411 |
| | | etoolbox | 105 |

| | | | |
|-----------------|----------|--------------------|--------------|
| everyhook | 117 | makeidx | 118 |
| everypage | 412 | marginfit | 470 |
| everyshi | 413 | marginfix | 470 |
| expl3 | 117 | marginnote | 471 |
| extramarks | 413 | mcaption | 471 |
| fancybox | 414 | mdframed | 472 |
| fancyhdr | 418 | memoir | 563 |
| fancyvrb | 419 | metalogo | 481 |
| figcaps | 425 | microtype | 65, 116, 482 |
| filecontents | 117 | moreverb | 483 |
| fix2col | 425 | mparhack | 485 |
| float | 426 | multicol | 486 |
| floatflt | 428 | multirow | 286, 488 |
| floatpag | 429 | nameref | 488 |
| floatrow | 430 | needspace | 489 |
| flushend | 436 | newclude | 89, 489 |
| fncychap | 437 | newfloat | 118 |
| fontenc | 64, 438 | newtxmath | 84 |
| fontspec | 64, 438 | newunicodechar | 64, 490 |
| footmisc | 439 | nextpage | 490 |
| footnote | 441 | nicefrac | 84 |
| footnotehyper | 442 | nonumonpart | 491 |
| framed | 443 | nopageno | 491 |
| ftnright | 446 | nowidow | 492 |
| fullpage | 446 | ntheorem | 493 |
| geometry | 116, 447 | pagenote | 504 |
| getttitlestring | 117 | paralist | 504 |
| glossaries | 79, 448 | parskip | 505 |
| graphics | 343, 450 | pdfscape | 505 |
| graphicx | 343, 450 | pdfsync | 506 |
| grffile | 451 | placeins | 506 |
| hyperref | 299, 452 | prelim2e | 507 |
| hyperxmp | 455 | ragged2e | 508 |
| idxlayout | 456 | realscripts | 509 |
| ifplatform | 105 | refcount | 118 |
| indentfirst | 457 | relsize | 510 |
| inputenc | 64, 457 | romanbar | 512 |
| keyfloat | 458 | romanbarpagenumber | 512 |
| kvoptions | 107 | rotating | 513 |
| layout | 460 | rotfloat | 514 |
| letltxmacro | 105 | savetrees | 515 |
| letterspace | 460 | scalefnt | 515 |
| lettrine | 461 | setspace | 516 |
| lips | 462 | shadow | 517 |
| listings | 463 | showidx | 518 |
| lmodern | 64 | showkeys | 518 |
| longtable | 467 | sidecap | 519 |
| lscape | 469 | sidenotes | 520 |
| lrcaption | 469 | siunitx | 83, 342, 522 |
| lwrap | 65 | soul | 523 |

| | | | |
|----------------------------------|----------|-------------------------------------|-----------------|
| subfig | 525 | pdflscape (package) | 505 |
| syntonly | 531 | pdfseparate (program) [requirement] | |
| tabularx | 532 | | 40, 44 |
| tabulary | 533 | pdfsync (package) | 506 |
| textcomp | 65, 534 | pdftocairo (program) [requirement] | 40, 44 |
| textpos | 536 | pdftotext (program) [requirement] | 40, 44 |
| theorem | 537 | Perl | 45 |
| threeparttable | 541 | perl (program) [requirement] | 45 |
| tikz | 542 | \phantomsection | 7492 |
| titles | 544 | picture (environment) | 360, 7029 |
| titleref | 545 | placeins (package) | 506 |
| titlesec | 546 | Plastex (program) | 36 |
| titletoc | 548 | \popclose | 3136 |
| titling | 119, 549 | Poppler | 40, 44 |
| tocbibind | 81, 550 | \postpublished | 236 |
| tocloft | 551 | \postsubtitle | 236 |
| transparent | 556 | prelim2e (package) | 507 |
| trivfloat | 557 | \prepublished | 236 |
| ulem | 559 | \presubtitle | 236, 246 |
| units | 84 | \printauthor | 237, 4500, 4515 |
| upref | 561 | \printdate | 237, 4505, 4517 |
| verse | 562, 563 | printglossary (option) [lwarpmk] | 80, 448 |
| wallpaper | 564 | \printindex | 5955 |
| wrapfig | 565 | \printpublished | 237, 4466, 4512 |
| xcolor | 567 | \PrintStack | 3254 |
| xfrac | 119, 576 | \prints subtitle | 237, 4491, 4514 |
| xifthen | 118 | \printthanks | 236, 240 |
| xltxtra | 579 | \printtitle | 237, 4475, 4513 |
| xmpincl | 580 | problems | 98 |
| xparse | 117 | programs | |
| xstring | 118 | utility | 39 |
| zref | 119 | programs: | |
| page | | [requirement]: | |
| inaccessible | 56 | LuaLaTeX | 40 |
| \pagebreak | 7441 | MathJax | 40 |
| \pagecolor | 44 | pdfcrop | 40 |
| pagenote (package) | 504 | pdfLaTeX | 40 |
| \pagenumbering | 3286 | pdfseparate | 40, 44 |
| \pageref | 5636 | pdftocairo | 40, 44 |
| \pagerefPageFor | 5635 | pdftotext | 40, 44 |
| \pagestyle | 3278 | perl | 45 |
| Pandoc (program) | 37 | XeLaTeX | 40 |
| \paragraph | 4172 | Adobe | 38 |
| paralist (package) | 504 | AsciiDoc | 37 |
| \parbox | 7130 | AsciiDoctor | 37 |
| \parsemulticolumnalignment | 5260 | Asciidoctor-LaTeX | 37 |
| parskip (package) | 505 | Flare | 38 |
| \part | 4126 | FrameMaker | 38 |
| pdfcrop (program) [requirement] | 40 | GELLMU | 36 |
| pdfLaTeX (program) [requirement] | 40 | GladTeX | 37 |

| | | | |
|----------------------------|-----------|------------------------------|--------------|
| Hevea | 36 | pdftotext (program) | 40, 44 |
| InDesign | 38 | perl (program) | 45 |
| LaTeX2HTML | 36 | XeLaTeX (program) | 40 |
| LaTeXML | 36 | \RequirePackage | 371 |
| LibreOffice | 37 | \resizebox | 6898 |
| Linux | 66, 106 | \restylefloat | 17 |
| lwarpmk | 96, 170 | \rmfamily | 7327 |
| Mac OS | 66, 106 | romanbar (package) | 512 |
| Madcap | 38 | romanbarpagenumber (package) | 512 |
| MathJax | 82, 318 | \rotatebox | 6863 |
| MS-Windows | 66, 106 | rotating (package) | 513 |
| OpenOffice | 37 | rotfloat (package) | 514 |
| Pandoc | 37 | \rule | 7489 |
| Plastex | 36 | | |
| TeX2page | 36 | S | |
| TeX4ht | 36 | sample_project.css (file) | 70, 167 |
| TtH | 36 | savetrees (package) | 515 |
| Unix | 66, 106 | \sb | 7335 |
| Windows | 66, 106 | \scalebox | 6880 |
| Word | 37 | scalefnt (package) | 515 |
| xindy | 70 | \scshape | 7332 |
| project.css (file) | 70 | section | |
| project.lwarpmkconf (file) | 131 | depths | 101 |
| project_html.tex (file) | 130 | missing | 56 |
| \published | 69, 4364 | \section | 4147 |
| \pushclose | 3109 | \SetHTMLFileNumber | 3192 |
| | | setspace (package) | 516 |
| | | settings | |
| Q | | css project-specific | 70 |
| \qqquad | 376, 7405 | CSS selection | 70 |
| \quad | 376, 7400 | HTML conversion | 55 |
| quote (environment) | 4707 | lwarpmk package options | 65 |
| | | selecting output | 67 |
| | | title page | 68 |
| R | | \sffamily | 7328 |
| ragged2e (package) | 508 | \sfrac | 576 |
| \raggedbottom | 3282 | shadow (package) | 517 |
| \raggedleft | 232 | showidx (package) | 518 |
| \raggedright | 232 | showkeys (package) | 518 |
| \raisebox | 7215 | sidecap (package) | 519 |
| realscripts (package) | 509 | sidenotes (package) | 520 |
| \ref | 5629 | SideTOCDepth (counter) | 56, 311 |
| refcount (package) | 118 | \sitetocname | 5884 |
| \reflectbox | 6893 | siunitx (package) | 83, 342, 522 |
| relsize (package) | 510 | \sloppy | 3284 |
| [requirement]: | | soul (package) | 523 |
| LuaLaTeX (program) | 40 | \sp | 7334 |
| MathJax (program) | 40 | space | |
| pdfcrop (program) | 40 | horizontal | 71 |
| pdfLaTeX (program) | 40 | between minipages | 376 |
| pdfseparate (program) | 40, 44 | | |
| pdftocairo (program) | 40, 44 | | |

| | | | |
|---|------------------|---|----------------|
| split | | <code>\thedata</code> | 237 |
| miss-numbered | 81, 83, 317, 493 | <code>theindex</code> (environment) | 5962 |
| stack depths | 101 | <code>theorem</code> (package) | 537 |
| <code>\StartDefiningTabulars</code> | 4918 | <code>\thepublished</code> | 237 |
| subequations | | <code>\thesubtitle</code> | 237 |
| MathJax | 83, 318 | <code>\thetitle</code> | 237 |
| <code>subfig</code> (package) | 525 | <code>\thispagestyle</code> | 3279 |
| <code>\subparagraph</code> | 4180 | <code>threeparttable</code> (package) | 541 |
| <code>\subsection</code> | 4155 | tikz | |
| <code>\subsubsection</code> | 4163 | catcodes | 542 |
| <code>\subtitle</code> | 69, 4375 | dollar redefined | 542 |
| SVG | | <code>tikz</code> (package) | 542 |
| images showing as HTML | 52, 99 | <code>tikzpicture</code> (environment) | 8 |
| math summary | 82, 317 | <code>\title</code> | 68 |
| <code>mathsvg</code> option | 108 | <code>titlepage</code> (environment) | 68, 4388 |
| <code>syntonly</code> (package) | 531 | <code>titlesps</code> (package) | 544 |
| | | <code>titleref</code> (package) | 545 |
| | | <code>titlesec</code> (package) | 546 |
| | | <code>titletoc</code> (package) | 548 |
| | | <code>titling</code> (package) | 119, 549 |
| | | <code>titlingpage</code> (environment) | 68, 4394, 4407 |
| | | [<code>tocbibind</code>]: | |
| | | <code>numindex</code> (option) | 81 |
| | | <code>tocbibind</code> (package) | 81, 550 |
| | | <code>tocdepth</code> (counter) | 55 |
| | | <code>tocloft</code> (package) | 551 |
| | | <code>\tracinglwrap</code> | 100, 478 |
| | | <code>transparent</code> (package) | 556 |
| | | trivfloat | |
| | | with <code>newfloat</code> , <code>algorithmic</code> | 557 |
| | | <code>trivfloat</code> (package) | 557 |
| | | troubleshooting | 98 |
| | | HTML debug comments | 125 |
| | | <code>\ttfamily</code> | 7329 |
| | | <code>TtH</code> (program) | 36 |
| | | <code>tutorial.tex</code> (file) | 46 |
| | | | |
| | | U | |
| | | <code>ulem</code> (package) | 559 |
| | | underscore | |
| | | filename | 55, 72 |
| | | Unicode | |
| | | enhanced coverage | 64 |
| | | file & section names | 219 |
| | | input characters | 104 |
| | | selection | 64 |
| | | <code>units</code> (package) | 84 |
| | | <code>Unix</code> (program) | 66, 106 |
| | | <code>\up</code> | 7338 |
| | | <code>upref</code> (package) | 561 |
| <code>split</code> | | | |
| miss-numbered | 81, 83, 317, 493 | | |
| stack depths | 101 | | |
| <code>\StartDefiningTabulars</code> | 4918 | | |
| subequations | | | |
| MathJax | 83, 318 | | |
| <code>subfig</code> (package) | 525 | | |
| <code>\subparagraph</code> | 4180 | | |
| <code>\subsection</code> | 4155 | | |
| <code>\subsubsection</code> | 4163 | | |
| <code>\subtitle</code> | 69, 4375 | | |
| SVG | | | |
| images showing as HTML | 52, 99 | | |
| math summary | 82, 317 | | |
| <code>mathsvg</code> option | 108 | | |
| <code>syntonly</code> (package) | 531 | | |
| | | | |
| | | T | |
| <code>\tableofcontents</code> | 58 | | |
| tabular | | | |
| baseline | 271 | | |
| HTML columnn conversion | 271 | | |
| <code>tabularx</code> (package) | 532 | | |
| <code>tabulary</code> (package) | 533 | | |
| tagged equations | | | |
| Mathjax | 83, 318 | | |
| <code>\TeX</code> | 7497 | | |
| <code>TeX2page</code> (program) | 36 | | |
| <code>TeX4ht</code> (program) | 36 | | |
| <code>\textbf</code> | 7256 | | |
| <code>\textcolor</code> | 34 | | |
| <code>textcomp</code> (package) | 65, 534 | | |
| <code>\textgreater</code> | 3182 | | |
| <code>\textit</code> | 7271 | | |
| <code>\textless</code> | 3182 | | |
| <code>\textmd</code> | 7253 | | |
| <code>\textnormal</code> | 7280 | | |
| <code>textpos</code> (package) | 536 | | |
| <code>\textrm</code> | 7259 | | |
| <code>\textsc</code> | 7274 | | |
| <code>\textsf</code> | 7262 | | |
| <code>\textsl</code> | 7277 | | |
| <code>\textsubscript</code> | 7337 | | |
| <code>\textsuperscript</code> | 7336 | | |
| <code>\texttt</code> | 7265 | | |
| <code>\textup</code> | 7268 | | |
| <code>\tfl@chapter@fix</code> | 557 | | |
| <code>\thanks</code> | 69 | | |
| <code>\thanksmarkseries</code> | 4674 | | |
| <code>\theauthor</code> | 237 | | |

| | | | |
|--|-------------------|--|-------------|
| <code>\upshape</code> | 7330 | <code>warpingprint</code> (boolean) | 107 |
| <code>\url</code> | 5681 | <code>warpprint</code> (environment) | 58, 67, 101 |
| <code>usingOSWindows</code> (boolean) | 106 | <code>warpprint</code> (option) | 65, 107 |
| UTF-8 | | <code>\warpprintonly</code> | 59, 67, 89 |
| enhanced coverage | 64 | Windows (program) | 66, 106 |
| file & section names | 219 | Word (program) | 37 |
| index | 65 | word processor | |
| selection | 64 | HTML conversion settings .. | 93, 94, 126 |
| utility | | <code>wrapfig</code> (package) | 565 |
| programs | 39 | | |
| | | X | |
| V | | <code>xcolor</code> (package) | 567 |
| <code>Verbatim</code> (environment) | 13 | <code>xdyFilename</code> (option) | 71, 108 |
| <code>verbatim</code> (environment) | 4764 | XeLaTeX | |
| <code>\VerbatimHTMLwidth</code> (length) | 252 | detection | 115 |
| <code>\verbatiminput</code> | 4756 | file & section names | 219 |
| <code>verse</code> (environment) | 2 | <code>\XeLaTeX</code> | 7510 |
| <code>verse</code> (package) | 562, 563 | XeLaTeX (program) [requirement] | 40 |
| viewport | | <code>\XeTeX</code> | 7510 |
| HTML meta tag | 230 | <code>xfrac</code> (package) | 119, 576 |
| <code>vleftmargini</code> (length) | 89, 251, 562 | <code>\xfracHTMLfontsize</code> | 3 |
| <code>vleftskip</code> (length) | 89, 251, 562, 563 | <code>xifthen</code> (package) | 118 |
| <code>\vspace</code> | 7438 | xindy | |
| | | customizing | 70 |
| W | | xindy (program) | 70 |
| <code>wallpaper</code> (package) | 564 | <code>xltxtra</code> (package) | 579 |
| warning icon | 103 | <code>xmpinl</code> (package) | 580 |
| <code>warpall</code> (environment) | 67, 100 | <code>xparse</code> (package) | 117 |
| <code>warpHTML</code> (environment) .. | 50, 59, 67, 101 | <code>xstring</code> (package) | 118 |
| <code>warpHTML</code> (option) | 65, 107 | | |
| <code>\warpHTMLonly</code> | 59, 67, 90 | Z | |
| <code>warpingHTML</code> (boolean) | 107 | <code>zref</code> (package) | 119 |