



The **lwarp** package

v0.26 — 2017/03/31

L^AT_EX to HTML5

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by L^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.

A modular package-loading system uses the **lwarp** version of a package for HTML when available. Several dozen L^AT_EX packages are supported with these high-level source compatibility replacements.

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 a “beta” version of **lwarp, and some things may change in response to user feedback and further project development.**

License:

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

Contents

1	lwarp.sty	19
1	Updates	19
2	Introduction	22
2.1	Supported packages and features	23
3	Alternatives	28
3.1	Internet class	28
3.2	TeX4ht	28
3.3	Translators	28
3.4	AsciiDoc	29
3.5	Pandoc	29
3.6	Word Processors	29
3.7	Commercial Systems	30
3.8	Comparisons	30
4	Installation	31
4.1	Installing the lwarp package	31
4.2	Installing the lwarpmk utility	34
4.2.1	Using a local copy of lwarpmk	35
4.3	Installing additional utilities	36
5	Tutorial	38
5.1	Starting a new project	38
5.2	Compiling the print version with lwarpmk	42
5.3	Compiling the HTML version with lwarpmk	43
5.4	Generating the SVG images	44
5.5	Using MathJax for math	45
5.6	Changing the CSS style	46
5.7	Customizing the HTML output	47
5.8	Using latexmk	51
5.9	Using XeLaTeX or LuaLaTeX	52
5.10	Cleaning auxiliary files	53
5.11	Cleaning auxiliary and output files	53
5.12	Processing multiple projects in the same directory	53
5.13	Using the make utility	53
6	Additional details	54
6.1	Font and UTF-8 support	54
6.2	lwarp package loading and options	55
6.3	Selecting the operating system	55
6.4	Selecting actions for print or HTML output	55
6.5	Commands to be placed into the warpprint environment	56

6.6	Commands for a successful HTML conversion	57
6.7	Title page	58
6.8	HTML page meta descriptions	59
6.9	CSS	60
6.10	Special cases and limitations	60
6.10.1	Text formatting	60
6.10.2	Footnotes and page notes	60
6.10.3	Math	61
6.10.4	Ntheorem	62
6.10.5	Graphics	63
6.10.6	Xcolor	63
6.10.7	Tabular	63
6.10.8	Minipages	65
6.10.9	Mdframed	65
6.10.10	float, trivfloat, and/or algorithmicx together	66
6.10.11	caption and subcaption packages	66
6.10.12	subfig package	66
6.10.13	floatrow package	66
6.10.14	abstract package	67
6.10.15	cleveref and varioref packages	67
6.10.16	verse and memoir	67
6.10.17	siunitx package	68
6.10.18	newclude package	68
6.10.19	newtxmath package	68
6.10.20	babel package	68
6.10.21	enumitem package	69
6.10.22	enumerate package	69
7	Expanding the code base	69
7.1	Creating an lwarp version of a package	69
7.2	Testing lwarp	70
8	Troubleshooting	71
8.1	Using the <code>lwarp.sty</code> package	71
8.1.1	Debug tracing output	73
8.2	Compiling the <code>lwarp.dtx</code> file	73
9	Implementation	73
10	Stack depths	74
11	Source Code	76
12	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	77
13	Unicode Input Characters	77

14	Handling package options	77
15	Misplaced packages	80
16	Required packages	83
17	Loading Packages	88
18	Copying a file	90
19	Debugging messages	91
20	Remembering original formatting macros	91
21	Operating-System portability	92
21.1	Common portability code	92
21.2	Unix, Linux, and Mac OS	93
21.3	MS-Windows	93
22	Stacks	93
22.1	Assigning depths	94
22.2	Closing actions	94
22.3	Closing depths	95
22.4	Pushing and popping the stack	95
23	Data arrays	97
24	HTML entities	97
25	HTML filename generation	98
26	Homepage link	100
27	\PrintStack diagnostic tool	101
28	Closing stack levels	101
29	HTML tags, spans, divs, elements	102
29.1	Mapping L ^A T _E X Sections to HTML Sections	102
29.2	HTML tags	103
29.3	Block tags and comments	104
29.4	Div class and element class	105
29.5	Single-line elements	106
29.6	HTML5 semantic elements	107
29.7	High-level block and inline classes	107
29.8	Closing HTML tags	108
30	Paragraph handling	109

31	Paragraph start/stop handling	112
32	Page headers and footers	115
33	CSS	116
34	HTML meta description	116
35	Footnotes	117
35.1	Regular page footnotes	117
35.2	Minipage footnotes	117
35.3	Titlepage thanks	117
35.4	Regular page footnote implementation	118
36	Marginpars	119
37	Splitting HTML files	119
38	Sectioning	124
38.1	Book class commands	124
38.2	Sectioning support macros	125
38.3	\section and friends	130
39	Starting a new file	131
40	Starting HTML output	134
41	Ending HTML output	137
42	Titles and the titling package	138
42.1	Setting the title, etc.	139
42.2	Changes to HTML titlepage and titlingpage	142
42.3	Printing the title, etc. in HTML	143
42.4	Printing the title, etc. in print form	145
42.5	\maketitle for print output	146
42.6	\maketitle for HTML output	149
43	Abstract	152
44	Quote and verse	152
44.1	Citations and attributions	152
44.2	Quotes, quotations	153
44.3	Verse	153
45	Verbatim	154
46	Fancyvrb	154

47	Theorems	160
48	Lists	161
48.1	Itemize	161
48.2	Enumerate	162
48.3	Description	163
49	Tabular	164
49.1	Token lookahead	166
49.2	Booleans	166
49.3	Handling & and !	167
49.4	Handling \\	169
49.5	Variables	170
49.6	Parsing @, >, <, ! columns	171
49.7	Parsing 'l', 'c', or 'r' columns	173
49.8	Parsing 'p', 'm', or 'b' columns	174
49.9	Parsing 'D' columns	174
49.10	Parsing the column specifications	174
49.11	Starting a new row	178
49.12	Data opening tag	179
49.13	Midrules	180
49.14	Multicolumns	181
49.14.1	Parsing multicolumns	181
49.14.2	High-level multicolumn interface	184
49.14.3	Longtable captions	185
49.14.4	\tabledatamulticolumntag	187
49.15	Multirow	188
49.16	Utility macros inside a table	188
49.17	Checking for a new table cell	188
49.18	\mrowcell	190
49.19	New \tabular definition	190
49.20	Array	192
50	Cross-references	193
50.1	Setup	193
50.2	Zref setup	195
50.3	Labels	196
50.4	References	197
50.5	Hyper-references	200
51	Floats	203
51.1	Float captions	204
51.1.1	Caption inside a float environment	205
51.1.2	Caption and LOF linking and tracking	205
52	Table of Contents, LOF, LOT	209

52.1	Reading and printing the TOC	209
52.2	High-level TOC commands	211
52.3	Side TOC	212
52.4	Low-level TOC line formatting	213
53	Index	215
54	Math	217
54.1	Inline and display math	219
54.2	MathJax support	221
54.3	Equation environment	224
54.4	AMS Math environments	225
54.4.1	Support macros	225
54.4.2	Environment patches	227
55	Lateximages	231
56	center, flushleft, flushright	237
57	Siunitx	238
58	Graphics	238
58.1	<code>\graphicspath</code>	239
58.2	Length conversions and graphics options	239
58.3	<code>\includegraphics</code>	242
58.4	<code>\rotatebox</code> , <code>\scalebox</code> , <code>\reflectbox</code>	246
58.5	Null functions	248
59	Cleverref	249
60	Picture	250
61	Boxes and Minipages	251
61.1	Counters and lengths	251
61.2	Footnote handling	252
61.3	Minipage handling	252
61.4	<code>Parbox</code> , <code>makebox</code> , <code>framebox</code> , <code>fbox</code> , <code>raisebox</code>	256
62	Direct formatting	257
63	Skips, spaces, font sizes	260
64	<code>\phantomsection</code>	266
65	<code>\LaTeX</code> and other logos	266
65.1	HTML logos	267
65.2	Print logos	268

66	\AtBeginDocument, \AtEndDocument	269
67	lwarpmk.conf	270
68	Trademarks	271
2	lwarp-newproject.sty	272
69	lwarp-newproject	272
69.1	project_html.tex	274
69.2	project.lwarpmkconf	274
69.3	lwarp.css	275
69.4	lwarp_sagebrush.css	298
69.5	lwarp_formal.css	303
69.6	sample_project.css	307
69.7	lwarp_html.xdy	308
69.8	lwarp_mathjax.txt	308
69.9	Lwarpmk option	310
3	lwarp-abstract.sty	320
70	Abstract	320
4	lwarp-afterpage.sty	322
71	Afterpage	322
5	lwarp-algorithmicx.sty	323
72	Algorithmicx	323
6	lwarp-alltt.sty	323
73	Alltt	324
7	lwarp-amsthm.sty	325
74	AMSthm	325

8	lwarp-bookmark.sty	327
75	Bookmark	327
9	lwarp-booktabs.sty	329
76	Booktabs	329
10	lwarp-ccaption.sty	329
77	Ccaption	329
11	lwarp-changepage.sty	330
78	Changepage	330
12	lwarp-cutwin.sty	331
79	Cutwin	331
13	lwarp-dcolumn.sty	331
80	Dcolumn	332
14	lwarp-draftwatermark.sty	333
81	Draftwatermark	333
15	lwarp-ellipsis.sty	333
82	Ellipsis	333
16	lwarp-emptypage.sty	334
83	Emtypage	334
17	lwarp-endnotes.sty	335
84	Endnotes	335

18	lwarp-enumerate.sty	336
85	Enumerate	336
19	lwarp-epigraph.sty	337
86	Epigraph	337
20	lwarp-eso-pic.sty	338
87	Eso-pic	338
21	lwarp-everypage.sty	339
88	Everypage	339
22	lwarp-extramarks.sty	339
89	Extramarks	339
23	lwarp-fancyhdr.sty	340
90	Fancyhdr	340
24	lwarp-float.sty	341
91	Float and \newfloat	341
25	lwarp-floatflt.sty	343
92	Floatflt	343
26	lwarp-floatrow.sty	344
93	Floatrow	344
27	lwarp-fontenc.sty	350
94	Fontenc	350

28	lwarp-fontspec.sty	350
95	Fontspec	350
29	lwarp-footmisc.sty	350
96	Footmisc	350
30	lwarp-footnote.sty	352
97	Footnote	352
31	lwarp-footnotehyper.sty	353
98	Footnotehyper	353
32	lwarp-framed.sty	354
99	Framed	354
33	lwarp-ftnright.sty	357
100	Ftnright	357
34	lwarp-geometry.sty	357
101	Geometry	357
35	lwarp-graphics.sty	358
102	Graphics	358
36	lwarp-graphicx.sty	358
103	Graphicx	358
37	lwarp-hyperref.sty	359
104	Hyperref	359

38	lwarp-indentfirst.sty	362
105	Indentfirst	362
39	lwarp-inputenc.sty	362
106	Inputenc	362
40	lwarp-keyfloat.sty	363
107	Keyfloat	363
41	lwarp-layout.sty	363
108	Layout	364
42	lwarp-letterspace.sty	364
109	Letterspace	364
43	lwarp-lips.sty	365
110	Lips	365
44	lwarp-listings.sty	366
111	Listings	366
45	lwarp-longtable.sty	370
112	Longtable	370
46	lwarp-lscape.sty	372
113	Lscape	372
47	lwarp-ltcaption.sty	372
114	Ltcaption	372

48	lwarp-marginfix.sty	373
115	Marginfix	373
49	lwarp-marginnote.sty	374
116	Marginnote	374
50	lwarp-mcaption.sty	374
117	Mcaption	374
51	lwarp-mdframed.sty	375
118	Mdframed	375
52	lwarp-microtype.sty	382
119	Microtype	382
53	lwarp-mparhack.sty	383
120	Mparhack	383
54	lwarp-multicol.sty	384
121	Multicol	384
55	lwarp-multirow.sty	386
122	Multirow	386
56	lwarp-nameref.sty	386
123	Nameref	386
57	lwarp-needspace.sty	387
124	Needspace	387

58	lwarp-newclude.sty	387
125	Newclude	387
59	lwarp-newunicodechar.sty	388
126	Newunicodechar	388
60	lwarp-nextpage.sty	388
127	Nextpage	388
61	lwarp-nowidow.sty	388
128	Nowidow	389
62	lwarp-ntheorem.sty	390
129	Ntheorem	390
63	lwarp-pagenote.sty	401
130	Pagenote	401
64	lwarp-parskip.sty	402
131	Parskip	402
65	lwarp-placeins.sty	402
132	Placeins	402
66	lwarp-ragged2e.sty	403
133	Ragged2e	403
67	lwarp-rotating.sty	404
134	Rotating	404

68	lwarp-setspace.sty	405
135	Setspace	405
69	lwarp-showidx.sty	405
136	Showidx	406
70	lwarp-showkeys.sty	406
137	Showkeys	406
71	lwarp-sidecap.sty	407
138	Sidecap	407
72	lwarp-sidenotes.sty	408
139	Sidenotes	408
73	lwarp-subfig.sty	410
140	Subfig	410
74	lwarp-tabularx.sty	416
141	Tabularx	416
75	lwarp-textpos.sty	417
142	Textpos	417
76	lwarp-theorem.sty	418
143	Theorem	418
77	lwarp-threeparttable.sty	422
144	Threeparttable	422

78	lwarp-tikz.sty	423
145	Tikz	423
79	lwarp-titleps.sty	424
146	Titleps	424
80	lwarp-titlesec.sty	426
147	Titlesec	426
81	lwarp-titletoc.sty	427
148	Titletoc	427
82	lwarp-titling.sty	428
149	Titling	428
83	lwarp-tocloft.sty	429
150	Tocloft	429
84	lwarp-trivfloat.sty	434
151	Trivfloat	434
151.1	Combining \newfloat, \trivfloat, and algorithmicx	434
85	lwarp-verse.sty	436
152	Verse	436
86	lwarp-wallpaper.sty	438
153	Wallpaper	438

87	lwarp-wrapfig.sty	439
-----------	--------------------------	------------

154	Wrapfig	439
-----	---------	-----

88	lwarp-xcolor.sty	441
-----------	-------------------------	------------

155	Xcolor	441
-----	--------	-----

89	lwarp-xfrac.sty	444
-----------	------------------------	------------

156	Xfrac	444
-----	-------	-----

	Change History and Index	447
--	---------------------------------	------------

List of Figures

1	tutorial.tex listing	39
---	--------------------------------	----

List of Tables

1	L ^A T _E X–HTML generation — lwarp package — Supported functions	23
2	Additional supported packages	26
3	Required software programs	32
4	Files created in a new project	41
5	Section depths and HTML headings	74
6	Tabular baseline	175
7	Tabular HTML column conversions	176
8	Cross-referencing data structures	194
9	Float data structures	203

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 [447](#).

v0.26:

- Improved installation instructions for MiKTeX regarding generating the `lwarpmk` executable.
- Footnotes are now supported by L^AT_EX boxes instead of pagenotes. `pagenote` now works as per the print version. `footnote`, `footnotehyper`, `footmisc`, `endnotes`, `marginnote`, and `sidenotes` are also supported.
- L^AT_EX labels now are used to track the page numbers of `lateximages`. This allows the correct inclusion of `lateximages` in footnotes, pagenotes, and endnotes.
- `cutwin` and `floatflt` are also supported.

v0.25:

- Fix: Allows `graphicx` and `graphicsx` before `lwarp` because X_YL^AT_EX and LuaL^AT_EX use `xunicode` which uses `graphics`.
- Package support for `framed`, several theorem packages, and ellipses.

v0.24:

- `tikz`'s `babel` library is load automatically as needed.
- `subfig` has been added, along with `lofdepth` and `lotdepth`.
- `picture` and `tikzpicture` now may be inline.

v0.22:

- Support has been added for tabular column types `D`, `!`, and `X`. Unknown column types are converted to `l`.
- Additional packages are supported, including `abstract`, `dcolumn`, `tabularx`, and `varioref`.

v0.21:

- Documentation for installing on Windows has been updated and improved.
- For Windows compatibility, the `lateximages` shell script has been replaced with a `lateximages.txt` file, which is parsed by `lwarpmk` to generate `lateximages`. This does not require any changes in the user's code.
- Windows `lwarpmk` again now functions.
- For improved error handling, `lwarp` now verifies the order in which packages are loaded, and signals an error for misplaced packages. `inputenc`, `fontenc`, `newunicode`, and `fontspec` must be loaded before `lwarp`, and the other packages which `lwarp` knows about must be loaded after.
- `lwarp` no longer requires a `\title` be assigned.

v0.20:

- The `makefile` and related infrastructure has been replaced by the `lwarpmk` utility. This provides increased portability, reduced dependencies, and much simpler installation and setup.
- The `lwarp-newproject` package is now used to locally create support files.
- The print and HTML versions of a document may co-exist with their own sets of auxiliary files.
- Package handling is now controlled by a modular system which looks for and loads an `lwarp-<package>` version if available.
- High-level source compatibility is provided for all supported packages, almost totally eliminating the need for `warpprint` and `warpHTML` environments.
- A large number of additional packages are supported.
- A new tutorial is included in the documentation, and many obsolete sections have been removed.
- `\NewHTMLdescription` sets the HTML meta description tag for each file. See section 6.8.
- `\HTMLFileName` may now be empty, allowing filenames without a prefix. Lwarp no longer automatically appends a `-` character. For existing projects, add a `-` to the end of `\HTMLFileName`.
- `\HomeHTMLFileName` and `\HTMLFileName` no longer use escaped underscore characters. Underscores may be used in filenames as-is.
- `lwarp` now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows.
- Tabular column types `@`, `>`, and `<` are now supported.



- `BlockClass` and `\InlineClass` add an optional style.
- The `sidebar` and `example` environments have been moved to the test suite, and are no longer included in `lwarp`.

v0.19:

- `MATHJAX` now may be used to display math via the `mathjax` option. See sections 6.2 and 6.10.3. To use MathJax with a pre-existing project, copy or link the file `lwarp_mathjax.txt` to the project's directory.
- `\rule` added, supporting width, height, raise, `\textcolor`.
- `\LateximageFontSizeName` provides user-adjustable font size for math and `lateximages`.
- `\minipagefullwidth` requests that the next minipage be full-width in HTML, but still the assigned width in print.
- `minipage` improved side-by-side rendering.
- CSS class `tablenotes` is provided for table note items.
- `\warpprintonly` replaces `\rowprintedonly`, and `\warpHTMLonly` is added. These behave like the `warpprint` and `warpHTML` environments, and are generally useful, so they replace the previously table-specific syntax.
- `cleveref` is loaded `\AtEndPreamble` for improved reliability. See section 59.
- `\xfracHTMLfontsize` controls `xfrac` font size in HTML.
- `Tikz` improved catcode handling.

**v0.18:**

- The `verse` package and the verse-related commands from the `memoir` package are now supported.
- Responsive web design has been improved for the sideTOC.
- `\includegraphics` now maintains relative sized for `em`, `ex`, and `%`.

v0.17:

- `mdframed` package is supported.

v0.16:

- Font and input encoding are now controlled by the user, and `lwarp` is loaded after fonts have been selected.
- Support for `XYLATEX` and `LuaATEX`. See section 6.1.

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 HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers 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 HTML5 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

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!

a L^AT_EX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

A special `lwarp-newproject` package is used to set up the auxiliary files necessary for the HTML conversion. 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 L^AT_EX 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 23.)

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 L^AT_EX 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 L^AT_EX code.

Table 1: L^AT_EX–HTML generation — `lwarp` package — Supported functions

Package or Macro	Status
Engines:	pdfL ^A T _E X, X _Y L ^A T _E X, LuaL ^A T _E X
Classes:	<code>book</code> , <code>report</code> , or <code>article</code> . <code>memoir</code> is planned.
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.

lwarp Supported Functions — continued

Package or Macro	Status
Table of Contents, Figures, Tables:	Supported, with hyperlinks.
<code>\maketitle</code> , <code>titlepage</code> , titling:	Supported, with optional titling-based commands for published and subtitle.
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleveref</code> commands, and index entries.
Footnotes, <code>pkgfootnote</code> , <code>footmisc</code> , <code>pagenote</code> , <code>endnotes</code> :	Supported.
Indexing:	<code>texindy</code> is used, with hyperlinks.
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. \LaTeX environments are supported. User-defined macros are available during conversion, due to native \LaTeX processing.
<code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> :	Supported, with added CSS tags.
Floats:	Supported, appear where declared. <code>float</code> is emulated, <code>newfloat</code> is supported, <code>caption</code> and <code>subcaption</code> are supported, <code>subfig</code> is supported, <code>capt-of</code> is supported, <code>placeins</code> is ignored, <code>trivfloat</code> is supported, <code>floatrow</code> is emulated, <code>keyfloat</code> is supported, <code>wrapfig</code> is emulated.
<code>tabular</code> :	Emulated. <code>\multirow</code> and <code>\multicolumn</code> are available, but cannot be used at the same time. Nested tables are not supported.
<code>array</code> :	Supported inside math environments, emulated elsewhere.

lwarp Supported Functions — continued

Package or Macro	Status
multirow:	Emulated.
longtable:	Emulated. Converted to a tabular. Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
booktabs:	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use width or trim options.
threeparttable:	Emulated
graphics, graphicx:	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
rotating:	Emulated. All objects are displayed unrotated.
Lists:	Supported
enumitem:	Supported, although spacing is still controlled by CSS.
Environments:	Standard L ^A T _E X environments are supported.
picture and tikz:	Converted to an SVG image.
minipage:	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
fancyvrb:	Supported except for verbatim footnotes.
framed, mdframed:	Supported
multicol:	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
siunitx:	Supported except for <code>per-mode=fraction</code> .

lwarp Supported Functions — continued

Package or Macro	Status
xfrac:	Supported
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
xcolor:	Supported. Full package color names, any color models, and <code>mixing</code> is converted to hex web colors via <code>\convertcolorspec</code> . Patched commands are <code>\textcolor</code> , <code>\colorbox</code> , and <code>\fcolorbox</code> . <code>\pagecolor</code> is not supported.
Where:	
Supported:	The existing \LaTeX package is used.
Emulated:	The \LaTeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

Supported packages include everything listed in the table of contents, plus each of the following in table 2, and probably others which have not yet been tested. Many are simply nullfied during HTML output. Others are not affected by the output mode and thus work as-is.

Table 2: Additional supported packages

babel, bm, calc, cleveref, csquotes, enumitem, fancyvrb, fileerr, newtxmath, siunitx, somedefs, tikz, trace, varioref, xspace

These packages and features probably works with little or no change to the user's source code. Special environments are provided to mark blocks of code which are for print only, HTML only, or both, should it be necessary.

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** **L^AT_EXML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **L^AT_EX2HTML**: <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 L^AT_EX 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 L^AT_EX 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 L^AT_EX-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 L^AT_EX 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 L^AT_EX-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success.

Prog LibreOffice

Prog OpenOffice See recent developments in Microsoft® Word® and LibreOffice™ Writer.

3.7 Commercial Systems

Prog	Adobe	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™.
Prog	FrameMaker	
Prog	InDesign	
	Flare	
Prog	Madcap	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 3 shows the tools which are used for the L^AT_EX 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.

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

```
Enter ⇒ tlmgr install lwarp
```

MiK_TE_X: If using MiK_TE_X, 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 T_EX-related packages.

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

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

T_EX Live:

```
Enter ⇒ kpsewhich -var-value TEXMFLOCAL
```

MiK_TE_X:

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 3: Required software programs

Provided by your L^AT_EX distribution:

From T_EXLive: <http://tug.org/texlive/>.

L^AT_EX: pdf_latex, xelatex, or lualatex.

The lwarp package: This package.

The lwarp-newproject package: Accompanies lwarp, and used to create configuration files.

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-newproject package generate a local copy of lwarpmk called lwarpmk.lua. See table 4.

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_TE_X 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 L^AT_EX code. On a MiK_TE_X system, Perl may need to be installed first.

pdftocrop: Used to pull images out of the L^AT_EX PDF.

Poppler PDF utilities:

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

From Poppler: poppler.freedesktop.org.

For Windows, also see:

<https://sourceforge.net/projects/poppler-win32/> and:
<http://blog.alivate.com.au/poppler-windows/>

pdftotext: Used to convert PDF to text.

pdfseparate: Used to pull images out of the L^AT_EX PDF.

pdftocairo: Used to convert images to SVG.

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 ⇒ `mktextlsr` — 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-newproject` package:
`\usepackage[lwarpmk]{lwarp-newproject}`
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

The T_EX utilities `latexmk` and `pdfcrop` may be provided by the installation of additional T_EX-relatex operating-system packages.

The tools from the POPPLER project may also be provided by your operating system's package-installation tools. For Windows, see table 3 on page 32.

Perl may require installation as well. See table 3.

To test for the existence of the additional utilities: Enter the following in a command line:

```
Enter ⇒ luatex -version

Enter ⇒ xindy -version

Enter ⇒ latexmk -version

Enter ⇒ perl -version

Enter ⇒ pdfcrop -version

Enter ⇒ pdftotext -v

Enter ⇒ pdfseparate -version

Enter ⇒ pdftocairo -v
```

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

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

To add the Poppler utilities to a Windows machine:

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

8. Click "Ok" when done.

Prog `perl`

To add 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.

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 \Rightarrow `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 \Rightarrow `pdflatex tutorial.tex` (several times)

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

A number of new files are created when `tutorial.tex` is compiled, as shown in table 4. These files are created by the `lwarp-newproject` package.

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created while the `lwarp-newproject` package is used, 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

% --- HTML FILENAME AND LATEXMK SETTINGS ---
% \newcommand{\HomeHTMLFilename}{index} % Filename of the homepage.
% \newcommand{\HTMLFileName}{node-} % Filename prefix of other pages.
% \newcommand{\UseLatexmk}{true}% Uncomment to use latexmk

% --- LWARP IS LOADED NEXT ---
\usepackage{lwarp-newproject}   % Possibly with the [lwarpmk] option.
\usepackage{lwarp}
% \boolfalse{FileSectionNames} % If false, numbers the files.

% --- OTHER PACKAGES ARE LOADED AFTER LWARP ---
\usepackage{makeidx} \makeindex
\usepackage{xcolor}

\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 sideTOC
\SetFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\SetPageTop{\fbox{LOGO}}
\SetPageBottom{Contact Information and Copyright}
\NewCSS{lwarp_sagebrush.css}
```

```
\begin{document}
\maketitle % (or a titlepage environment)

% --- An abstract may be placed here. ---

\tableofcontents \listoffigures % --- MUST BE BEFORE THE FIRST SECTION.

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

\printindex

\end{document}
```


Table 4: Files created in a new project

- tutorial.pdf:** The PDF output from L^AT_EX.
- 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 L^AT_EX files `tutorial.aux`, `tutorial.log`. 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_html.xdy:** Used by `lwarp` while creating an index. This file should not need to be modified by the user.
- lwarp_mathjax.txt:** Inserted into the HTML files when `MathJax` is used to display math. This file should not need to 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-newproject` 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 \Rightarrow `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 \Rightarrow `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 \Rightarrow `lwarpmk again`

Enter \Rightarrow `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.

Enter \Rightarrow `lwarpmk printindex`

6. Recompile again to include the index.

Enter \Rightarrow `lwarpmk print`

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.

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 into the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`⁴, `Some-math.html`, and the document's index in `_Index.html`.⁵

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 not yet 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:

```
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.

⁴`First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

⁵`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

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` tags. 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.


⚠ **Lots of files!** Expressing math as SVG images has the advantage of representing the math exactly as \LaTeX would, but has the disadvantage of requiring an individual file for each math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time $\$x\$$ is used, for example, yet another file is created. For a document with a large amount of math, see section 5.5 to use MathJax instead.

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, change
`\usepackage{lwarp}`
to
`\usepackage[mathjax]{lwarp}`
2. Recompile

Enter \Rightarrow `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 .

5.6 Changing the CSS style

\NewCSS **\NewCSS** may be used to choose which `.css` file is used to display each section of the web page. Use **\NewCSS** before `\begin{document}` to assign the style of the home page. If different parts of the website should have different styles, call **\NewCSS** again before each section heading which creates a new file.

The styles provided by `lwarp` include:


lwarp.css: A default style if **\NewCSS** is not used. This style is comparable to a plain L^AT_EX document. To set this style, you may use **\NewCSS{lwarp.css}**, or no **\NewCSS** 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 **\NewCSS** entry in the tutorial `lwarpmk.html`, 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 **\NewCSS{<project>.css}**.

 **Rename it!** Note that `sample_project.css` is overwritten whenever `lwarp-newproject` 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 use `lwarpmk clearall` to clear all the HTML, PDF, and auxiliary files. Also, if `\HomeHTMLFilename`, `\HTMLFileName`, or `\UseLatexmk` are changed, it is necessary to reenale the `lwarp-newproject` package and then recompile the print version in order to recreate the configuration files for `lwarpmk`.

Placed in the preamble just before `lwarp` is loaded:

<code>\HomeHTMLFilename</code>	\HomeHTMLFilename: Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is: <code>\newcommand{HomeHTMLFilename}{index}</code> causing the homepage to be the file <code>index.html</code> . Underscores are allowed in <code>\HomeHTMLFilename</code> and <code>\HTMLFileName</code> , but usually must be escaped elsewhere, such as <code>\item [\href{file_name.pdf}{text}] \</code>
filename underscore	
<code>\HTMLFileName</code>	\HTMLFileName: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.
<code>\UseLatexmk</code>	\UseLatexmk: A macro which controls whether <code>lwarp</code> uses <code>latexmk</code> to compile the document. This setting is written to <code>lwarpmk</code> ’s configuration files. Defaults to false. Set to true with: <code>\newcommand*{\UseLatexmk}{true}</code>

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

Ctr <code>tocdepth</code>	tocdepth: Sectioning depth of the table of contents. See section 10 for a list of L ^A T _E X stack depths.
Ctr <code>SideTOCDepth</code>	SideTOCDepth: Sectioning depth of the sideTOC. Defaults to 1, causing the sidetoc to show sections but not subsections.
sideTOC	Each subpage of the website has its own small table of contents on the side (the “sideTOC”). Its depth is set by <code>SideTOCDepth</code> . This sideTOC is only shown if the web page is wide enough. When using a narrow web browser window, “responsive web design” is used to show the sideTOC at the top of the page and a link back to “Home” at the bottom. It is recommended to set: <code>SideTOCDepth = FileDepth</code> , or <code>SideTOCDepth = FileDepth+1</code> If <code>SideTOCDepth < FileDepth</code> , web pages will be inaccessible via the sideTOC.
⚠	
Ctr <code>FileDepth</code>	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` is prefixed.

Example HTML filenames:

HTML filenames

Numbered HTML nodes:

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

```
\newcommand{\HomeHTMLFileName}{index}
\newcommand{\HTMLFileName}{node-}
\usepackage{lwarp}
\boolfalse{FileSectionNames}
```

Named HTML sections, no prefix:

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

```
\newcommand{\HomeHTMLFileName}{index}
\newcommand{\HTMLFileName}{}
```

```
\usepackage{lwrap}
\booltrue{FileSectionNames}
```

Named HTML sections, with prefix:

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

```
\newcommand{\HomeHTMLFileName}{mywebsite}
\newcommand{\HTMLFileName}{mywebsite-}
\usepackage{lwrap}
\booltrue{FileSectionNames}
```

<code>\abstractname</code>	\abstractname: The name of the abstract. This may also be over-written by the babel package. Defaults to “Abstract”.
<code>\MetaLanguage</code>	\MetaLanguage: The HTML language meta header. Defaults to en-US.
<code>\SetFirstPageTop</code>	\SetFirstPageTop: <code>{\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>\SetPageTop</code>	\SetPageTop: <code>{\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>\SetPageBottom</code>	\SetPageBottom: <code>{\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 section break:

<code>\tableofcontents</code>	\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 <code>tocdepth</code> .
-------------------------------	---

Placed before any sectioning command which causes a file break:

<code>\NewCSS</code>	\NewCSS: <code>{\langle filename.css \rangle}</code> 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 <code>\NewCSS</code> command. If <code>\NewCSS</code> is not used, a default plain style is used to mimic printed L ^A T _E X output. <code>lwrap_sagebrush.css</code> is a semi-fancy colored style as shown in this tutorial. Change it to <code>lwrap_formal.css</code> for a more formal look, or comment out the <code>\NewCSS</code> command to see the default. <code>\NewCSS</code> may be used before each file break to set the CSS for individual pages of the website.
----------------------	---

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

Placed in the document wherever necessary:

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

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

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

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

5.8 Using latexmk

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

1. In the tutorial's source code, at "HTML FILENAME AND LATEXMK SETTINGS", insert the line:

```
\newcommand{\UseLatexmk}{true}
```

2. Directly recompile the printed version of the document.

⚠ NOT `lwrapmk`!

```
Enter ⇒ pdflatex tutorial.tex
```

```
(Or xelatex or lualatex)
```

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

3. Recompile the document.

```
Enter ⇒ lwrapmk print and/or
```

```
Enter ⇒ lwrapmk html
```

Changes are detected by comparing checksums rather than modification times, so `lwrapmk` again will not trigger a recompile, but `latexmk` has a much better awareness of changes than the `lwrapmk` 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.

5.9 Using XeLaTeX or LuaLaTeX

X_YLaTeX or LuaLaTeX may be used instead of LaTeX.

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

2. Be sure that

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

is still enabled in `tutorial.tex`.

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

```
Enter ⇒ xelatex tutorial.tex -or-
```

```
Enter ⇒ lualatex tutorial.tex
```

When the recompile occurs and `lwarp-newproject` is loaded, the configuration files for `lwarpmk` are modified to remember which TeX engine was used. X_YLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

4. To recompile the document:

```
Enter ⇒ lwarpmk print -and-
```

```
Enter ⇒ lwarpmk html
```

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

5.10 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

5.11 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

5.12 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

```
project_a.lwarpmkconf, project_b.lwarpmkconf
```

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

5.13 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_YLaTeX and LuaLaTeX users must use the `fontspec` package. Do NOT use `fontenc`!

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

1. `documentclass{article/book/report}` goes here, followed by any of:
2. Font and UTF-8 related commands:

- For X_YLaTeX or LuaLaTeX:

Pkg `fontspec`

- `fontspec` and font choices

`ligatures`

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

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
```

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File `glyphtounicode`

- `\input glyphtounicode.tex`

- `\input glyphtounicode-cmr.tex`% from the `pdfx` package

- `\pdfgentounicode=1`

Pkg	<code>cmap</code>	– <code>cmap</code>
Pkg	<code>textcomp</code>	– <code>textcomp</code>
Pkg	<code>microtype</code> <code>ligatures</code>	– <code>microtype</code> is automatically used by <code>lwarp</code> to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off \TeX ligatures.

3. `\usepackage{lwarp}` (section 6.2) goes after any of the above, followed by:

4. ... the rest of the preamble and the main document.

6.2 lwarp package loading and options

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

Pkg	<code>lwarp</code>	Load the <code>lwarp</code> package immediately after the font and UTF-8 setup commands.
Opt lwarp	<code>warpprint</code>	Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When <code>lwarp-newproject</code> 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 lwarp	<code>warpHTML</code>	
Opt lwarp	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 6.10.3.
Opt lwarp	<code>mathjax</code>	

6.3 Selecting the operating system

`lwarp` tries to detect which operating system is being used.

Prog	<code>MS-Windows</code>	If MS-WINDOWS is not correctly detected, use the command <code>\warpOSwindows</code> in the document preamble after <code>lwarp</code> is loaded. This modifies the operating-system path separator used by <code>lwarp</code> .
Prog	<code>Windows</code>	
	<code>\warpOSwindows</code>	

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 \LaTeX print-formatted PDF generation, or to HTML generation.

For most of built-in \LaTeX and many additional packages there is user-level source code support or emulation, so no special handling will be required. For those cases

which `lwarp` does not handle by itself, the following environments and macros may be used to isolate sections of code for print-only or HTML-only.

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

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

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

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

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

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

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

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

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

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`
- Variable spaces such as `\vspace`. `\hfill` is turned into a `\quad`. Fixed spaces such as `\quad` are emulated correctly.
- 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 8: [Troubleshooting](#).

6.6 Commands for a successful HTML conversion

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

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.

\bfseries, etc: Use `\textbf` instead.

\centering, \raggedright, \raggedleft:

Use the environments `center`, `flushright`, `flushleft` instead.

Superscripts and other non-math uses of math mode:

Use `x` instead of x

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

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

Filenames in lists:

[filename underscore](#)

Escape underscores in the filenames:

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

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.

\fbox around a minipage:

\fbox can only be used around inline items during HTML output.

For an \fbox around a minipage, you may:

- Place the \fbox command and its closing brace inside `warpprint` environments.
- Use `\mdframed` instead.
- 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}
```

Also see section 8: [Troubleshooting](#).

6.7 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 <code>titlepage</code> will be set in a <code>div class 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. |

`\title` $\{\langle title \rangle\}$



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.

6.8 HTML page meta descriptions

`\NewHTMLdescription` $\{\langle A \text{ description of the web page.} \rangle\}$ 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 `\NewHTMLdescription` 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 `\NewHTMLdescription` 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 `\NewHTMLdescription`. It is best to use a unique description for each HTML file.

disabling To disable the generation of HTML description meta tags, use:
`\NewCSSdescription{}`

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`.

`\NewCSS` For each section at which HTML files are split, `\NewCSS` 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.

6.10 Special cases and limitations

6.10.1 Text formatting

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

6.10.2 Footnotes and page notes

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

6.10.3 Math

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 L^AT_EX into the initial PDF file using the current font⁶, 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 L^AT_EX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML ALT tag carries the L^AT_EX code which generated the math, allowing copy/paste of the L^AT_EX 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.

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 L^AT_EX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of L^AT_EX, and
2. As detokenized printed L^AT_EX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used

⁶See section 156 regarding fonts and fractions.

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

For an example of using custom \LaTeX macros with MathJax, see page 219.

6.10.4 Ntheorem

⚠ Font control

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

⚠ Equation numbering

`ntheorem` has a bug with equation numbering in 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.


6.10.5 Graphics

`\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 CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \LaTeX , so expect some ugly results for scaling and rotating.

6.10.6 Xcolor

`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.

6.10.7 Tabular

`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.
- `\newcolumnntype` is ignored; unknown column types are set to 1.
- `tabularx` ignores the width, but `X` columns do produce paragraph columns or multicolumns.
- Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
- 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.
- 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.
- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.
- Multiple paragraphs in one cell of a p, b, m column must have `\newline` between paragraphs.
- `\cmidrule` does not support width or trim options due to CSS limitations.
- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.
- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a “misplaced `\noalign`” error occurs, change
`This & That \endhead`
to
`\warpprintonly{This & That \endhead}`
and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.
- For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `lwarp`’s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:
`\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}`
`\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}`

⚠ `\multirow &`
`\multicolumn`

⚠ `\multirow`

`\multirow with rules`

rule at last row

⚠ paragraphs


`\cmidrule width, trim`
`longtable headings`

⚠ `\warpprintonly`

⚠ `S` columns

6.10.8 Minipages

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

 **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`.


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 are ignored. 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` tag, 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 L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

6.10.9 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, `lwrap` loads `mdframed` in HTML with `framemethod=none`.

For title font, use

```
frametitlefont=\textbf,
```

font 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.

6.10.10 `float`, `trivfloat`, and/or `algorithmicx` together

⚠ `package conflicts` If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 151.1.

6.10.11 `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`.

6.10.12 `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.

6.10.13 `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.

6.10.14 `abstract` package

`abstract` is supported. 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.

6.10.15 `cleveref` and `varioref` packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML. See section 6.6 to redefine the message which is printed for page number references.

6.10.16 `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}
```


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

Horizontal spacing relies on `pdftotext`'s ability to discern the `-layout` 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.

6.10.17 siunitx package

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


6.10.18 newclue package

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

Ex:

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


6.10.19 newtxmath package

Pkg `newtxmath` The proper load order is:
 `loading sequence`

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

6.10.20 babel package

Pkg `babel` If using `babel` with French, use

 **French** `\frenchbsetup{StandardLists=true}` to preserve the special HTML and `enumitem` list handling.

\CaptionSeparator Also, 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}{:~}
```

6.10.21 `enumitem` package

Pkg `enumitem` `enumitem` is pre-loaded during HTML output. Many of the spacing options are rendered irrelevant by `pdftotext` and HTML. Numbering, labels, and `\newlist` function correctly.

6.10.22 `enumerate` package

Pkg `enumerate` `enumerate` conflicts with `enumitem` if both are loaded at the same time, but `lwarp` does not actually load `enumerate`. While generating HTML, `lwarp` only loads `enumitem`, and `enumerate` is simulated by `enumitem` using the functionality of the `shortlabels` option.

A problem may occur during print output if `enumitem` is loaded, either manually or by some other package such as `siunitx`. If these are used, `enumerate` will conflict with `enumitem` during print output.

7 Expanding the code base

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 L^AT_EX output. See section 60 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 29.7.

7.1 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 129 (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.

7.2 Testing lwarp

When changes have been made, test the print output before testing the HTML. The print output compiles faster, and any errors in the printed version will be easier to figure out than the HTML version.

The package `lwarp-newproject` must be loaded to set up the configuration files, but may be commented out from then on unless the configuration changes. A small amount of time is taken each time `lwarp-newproject` is run.

Remember that the configuration files are only rewritten when compiling the printed version of the document, and when `lwarp-newproject` is enabled.

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.

8 Troubleshooting

8.1 Using the `lwarp.sty` package

Also see:

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

Section 6.6: [Commands for a successful HTML conversion](#)

Section 6.10: [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.

Obscure error messages:

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

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.

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

Use \XeLaTeX or \LuaLaTeX .

Floats out of sequence:

Mixed “Here” and floating: Floats [H]ere and regular floats may become out of order. `\clearpage` if necessary.

Caption setup: With `\captionsetup` set the positions for the captions above or below to match their use in the source code.

Print document contains HTML tags:

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

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.

“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 `\NewCSS` 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.

8.1.1 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.

8.2 Compiling the `lwarp.dtx` file

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}`

9 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.

Table 5: Section depths and HTML headings

Section	L ^A T _E X depth	HTML headings
title of the entire website		h1
none	-5	new for this package
book	-2	not yet used
part	-1	h2
chapter	0	h3
section	1	h4
subsection	2	h5
subsubsection	3	h6
paragraph	4	span class = "paragraph"
subparagraph	5	span class = "subparagraph"
listitem	7	new for this package, used for list items

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.

10 Stack depths

Stacks are created to track depth inside the L^AT_EX 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 L^AT_EX 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 will push the HTML headings down a step, and also cause `subsubsection` to become a `div` due to a limit of six HTML headings.

It is possible to use HTML5 `section` and `H1` for all levels, but this may not be well-recognized by older browsers.

11 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 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:
for PRINT output:
for HTML & PRINT: Green-colored tags in the left margin show which sections of source code apply to the generation of HTML, print, or both forms of output.

lwarp source code begins on the following page.

12 Detecting the T_EX Engine — pdf_latex, lua_latex, xelatex

```

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

```

13 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

```

14 Handling package options

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

```

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

```

Pkg	ifplatform	Provides <code>\ifwindows</code> to try to automatically detect Windows OS.
		24 <code>\RequirePackage{ifplatform}% sense op-system platform</code>
Pkg	comment	Provides conditional code blocks.
		25 <code>\RequirePackage{comment}</code>
		26 <code>\excludecomment{testing}</code>
Pkg	kvoptions	Allows key/value package options.
		27 <code>\RequirePackage{kvoptions}</code>
		28 <code>\SetupKeyvalOptions{family=LWR,prefix=LWR@}</code>
Bool	warpingprint	Set to true/false depending on the package option selections for print/HTML output
Bool	warpingHTML	and mathsvg/mathjax:
Bool	mathjax	29 <code>\newbool{warpingprint}</code>
		30 <code>\newbool{warpingHTML}</code>
		31 <code>\newbool{mathjax}</code>
	\warpprintonly	<code>{\langle contents \rangle}</code>
		Only process the contents if producing printed output.
		32 <code>\newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}</code>
	\warpHTMLonly	<code>{\langle contents \rangle}</code>
		Only process the contents if producing HTML output.
		33 <code>\newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}</code>
Env	warpall	Anything in the <code>warpall</code> environment will be generated for print or HTML outputs.
		34 <code>\includecomment{warpall}</code>
Env	warpprint	Anything in the <code>warpprint</code> environment will be generated for print output only.
Opt lwarp	warpprint	If the <code>warpprint</code> option is given, boolean <code>warpingprint</code> is true and boolean <code>warpingHTML</code> is false, and may be used for <code>\ifbool</code> tests.
		35 <code>\DeclareVoidOption{warpprint}{%</code>
		36 <code>\PackageInfo{lwarp}{Using option 'warpprint'}</code>
		37 <code>\includecomment{warpprint}%</code>
		38 <code>\excludecomment{warpHTML}%</code>

```

39 \booltrue{warpingprint}%
40 \boolfalse{warpingHTML}%
41 }

```

Env **warpHTML** Anything in the **warpHTML** environment will be generated for HTML output only.

Opt **lwarp** **warpHTML** If the **warpHTML** option is given, boolean **warpingHTML** is true and boolean **warpingprint** is false, and may be used for **\ifbool** tests.

```

42 \DeclareVoidOption{warpHTML}{%
43 \PackageInfo{lwarp}{Using option 'warpHTML'}
44 \excludcomment{warpprint}%
45 \includecomment{warpHTML}%
46 \booltrue{warpingHTML}%
47 \boolfalse{warpingprint}%
48 }

```

Opt **lwarp** **mathsvg** Option **mathsvg** selects SVG math display: If the **mathsvg** option is given, boolean **mathjax** is false, and may be used for **\ifbool** tests.

```

49 \DeclareVoidOption{mathsvg}{%
50 \PackageInfo{lwarp}{Using option 'mathsvg'}
51 \boolfalse{mathjax}%
52 }

```

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

```

53 \DeclareVoidOption{mathjax}{%
54 \PackageInfo{lwarp}{Using option 'mathjax'}
55 \booltrue{mathjax}%
56 }

```

Opt **lwarp** **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.

```

57 \DeclareStringOption[\jobname]{BaseJobname}

```

The default is print output, and SVG math if the user chose HTML output.

```

58 \includecomment{warpprint}%
59 \excludcomment{warpHTML}%
60 \booltrue{warpingprint}%
61 \boolfalse{warpingHTML}%
62 \boolfalse{mathjax}%

```

```

63
64
65 % \ExecuteOptions{warpprint,mathsvg}
66
67 \ProcessKeyvalOptions*\relax

```

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

```

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

```

`\@nameedef` $\{\langle name \rangle\} \{\langle defn \rangle\}$

Expands `<defn>` before assigning to `<name>`

```

69 \def\@nameedef#1{\expandafter\edef\csname #1\endcsname}

```

15 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` $\{\langle packagename \rangle\}$ Error if this package was loaded before `lwarp`.

```

70 \newcommand*\LWR@loadafter[1]{%
71 \ifpackageloaded{#1}
72 {
73 \PackageError{lwarp}
74 {Package #1, or one which uses #1, must be loaded after lwarp}
75 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
76 Package #1 may also be loaded by something else, which must also be moved
77 after lwarp.}
78 }
79 {}
80 }

```

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

```

81 \newcommand*\LWR@loadbefore[1]{%
82 \PackageError{lwarp}

```



```

83 {Package #1 must be loaded before lwarp}
84 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
85 }

```

`\LWR@loadnever` $\{\langle badpackagename \rangle\} \{\langle replacementpkgnam \rangle\}$

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

```

86 \newcommand*{\LWR@loadnever}[2]{%
87 \PackageError{lwarp}
88 {Package #1 does not work with lwarp's HTML conversion.
89 Please use the #2 package instead}
90 {Package #1 conflicts with lwarp in some way, but package #2 probably will work instead.}
91 }

```

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`:

```

92 \LWR@loadafter{abstract}
93 \LWR@loadafter{afterpage}
94 \LWR@loadafter{algorithmicx}
95 \LWR@loadafter{alltt}
96 \LWR@loadafter{amsthm}
97 \LWR@loadafter{bookmark}
98 \LWR@loadafter{booktabs}
99 \LWR@loadafter{ccaption}
100 \LWR@loadafter{changepage}
101 \LWR@loadafter{cutwin}
102 \LWR@loadafter{dcolumn}
103 \LWR@loadafter{draftwatermark}
104 \LWR@loadafter{ellipsis}
105 \LWR@loadafter{emptypage}
106 \LWR@loadafter{enumerate}
107 \LWR@loadafter{epigraph}
108 \LWR@loadafter{eso-pic}
109 \LWR@loadafter{everypage}
110 \LWR@loadafter{extramarks}
111 \LWR@loadafter{fancyhdr}
112 \LWR@loadafter{floatrow}
113 \LWR@loadafter{float}
114 \LWR@loadafter{floatflt}
115 \LWR@loadafter{ftnright}
116 \LWR@loadafter{geometry}
117 % \LWR@loadafter{graphics}% pre-loaded by xunicode
118 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
119 \LWR@loadafter{hyperref}
120 \LWR@loadafter{indentfirst}

```

```
121 \LWR@loadafter{keyfloat}
122 \LWR@loadafter{layout}
123 \LWR@loadafter{letterspace}
124 \LWR@loadafter{lips}
125 \LWR@loadafter{listings}
126 \LWR@loadafter{longtable}
127 \LWR@loadafter{lscape}
128 \LWR@loadafter{ltcaption}
129 \LWR@loadafter{marginfix}
130 \LWR@loadafter{marginnote}
131 \LWR@loadafter{mcaption}
132 \LWR@loadafter{mdframed}
133 \LWR@loadafter{microtype}
134 \LWR@loadafter{mparhack}
135 %\LWR@loadafter{multicol}% loaded by ltxdoc
136 \LWR@loadafter{multirow}
137 \LWR@loadafter{nameref}
138 \LWR@loadafter{needspace}
139 \LWR@loadafter{newtxmath}
140 \LWR@loadafter{nextpage}
141 \LWR@loadafter{nowidow}
142 \LWR@loadafter{ntheorem}
143 \LWR@loadafter{pagenote}
144 \LWR@loadafter{parskip}
145 \LWR@loadafter{placeins}
146 \LWR@loadafter{ragged2e}
147 \LWR@loadafter{rotating}
148 \LWR@loadafter{setspace}
149 \LWR@loadafter{showidx}
150 \LWR@loadafter{showkeys}
151 \LWR@loadafter{sidecap}
152 \LWR@loadafter{sidenotes}
153 \LWR@loadafter{subfig}
154 \LWR@loadafter{tabularx}
155 \LWR@loadafter{textpos}
156 \LWR@loadafter{theorem}
157 \LWR@loadafter{threeparttable}
158 \LWR@loadafter{tikz}
159 \LWR@loadafter{titleps}
160 \LWR@loadafter{titlesec}
161 \LWR@loadafter{titletoc}
162 \LWR@loadafter{tocloft}
163 \LWR@loadafter{trivfloat}
164 \LWR@loadafter{varioref}
165 \LWR@loadafter{verse}
166 \LWR@loadafter{wallpaper}
167 \LWR@loadafter{wrapfig}
168 \LWR@loadafter{xcolor}
169 \LWR@loadafter{xfrac}
```

16 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: 170 `\begin{warpall}`

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

Detects \TeX and \LuaTeX :

```
171 \RequirePackage{iftex}
172 \newif\ifxetexorluatex
173 \ifXeTeX
174   \xetexorluatextrue
175 \else
176   \ifLuaTeX
177     \xetexorluatextrue
178   \else
179     \xetexorluatexfalse
180   \fi
181 \fi

182 \end{warpall}
```

for HTML output: 183 `\begin{warpHTML}`

```
184 \ifxetexorluatex
185 % ^^A \usepackage[no-math]{fontspec}
```

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

```
186 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
187 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
188 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
189 \else
```

pdflatex only: Only pre-loaded if `pdflatex` is being used.

Pkg microtype

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

```

190 \RequirePackage {microtype}
191
192 \microtypesetup{
193 protrusion=false,
194 expansion=false,
195 tracking=false,
196 kerning=false,
197 spacing=false}
198
199 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

200 \fi

201 \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.
- Will also use a scriptsize font.
- Also uses extra space at the margin to avoid HTML tag overflow off the page.
- Will also force a new page before environments.

for HTML output:

```

202 \begin{warpHTML}
203 \RequirePackage[paperheight=190in,paperwidth=20in,%
204 left=7in,right=7in,%
205 top=1in,bottom=1in,%
206 ]{geometry}
207 \end{warpHTML}

```

for HTML & PRINT:

```

208 \begin{warpall}

```

Pkg **xparse**

L^AT_EX3 command argument parsing

```

209 \RequirePackage{xparse}

210 \end{warpall}

```

for HTML output:

```

211 \begin{warpHTML}

```

Pkg `expl3`

L^AT_EX3 programming

```
212 \RequirePackage{expl3}
```

Pkg `getttitlestring`

Used to emulate `\nameref`.

```
213 \RequirePackage{getttitlestring}
```

Pkg `everyhook`

`everyhook` is used to patch paragraph handling.

```
214 \RequirePackage{everyhook}
```

```
215 \end{warpHTML}
```

for HTML & PRINT:

```
216 \begin{warpall}
```

Pkg `fancyvrb`

Used for Verbatim, verse.

```
217 \RequirePackage{fancyvrb}
```

```
218 \end{warpall}
```

for HTML output:

```
219 \begin{warpHTML}
```

Pkg `xifthen`

```
220 \RequirePackage{xifthen}
```

Pkg `xstring`

```
221 \RequirePackage{xstring}
```

Pkg `makeidx`

```
222 \RequirePackage{makeidx}
```

```
223 \makeindex
```

Pkg `calc`

```
224 \RequirePackage{calc}
```

Pkg **refcount**

```
225 \RequirePackage{refcount}
```

Pkg **newfloat**

```
226 \RequirePackage{newfloat}
```

Pkg **caption**

```
227 \RequirePackage{caption}
```

Pkg **enumitem**

enumitem is patched to support `\newlist` with HTML.

```
228 \RequirePackage{enumitem}
229 \setlist[itemize]{leftmargin=0em}
230 \setlist[enumerate]{leftmargin=0em}
231 \setlist[description]{leftmargin=0em}
```

```
232 \end{warpHTML}
```

for HTML & PRINT: 233 \begin{warpall}

Pkg **titling**

Used for `\maketitle` and the title page. See section 42.

```
234 \RequirePackage{titling}
```

```
235 \end{warpall}
```

for HTML output: 236 \begin{warpHTML}

Pkg **zref**

Used for cross-references.

```
237 \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.

```
238 \PassOptionsToPackage{leqno,fleqn}{amsmath}
239 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
240 \RequirePackage{environ}
```

Pkg **titleps**

Used to place an HTML comment into the footer of a page below the footnotes. This comment is used for `lateximage` environments, including math.

The `nopatches` option prevents titleps from trying to patch sectioning commands.

`\pagestyle` and `\thispagestyle` are nullified for HTML output.

```
241 \RequirePackage[nopatches]{titleps}
```

`\pagestyle` $\{\langle style \rangle\}$

```
242 \let\LWR@origpagestyle\pagestyle
243 \renewcommand*{\pagestyle}[1]{}
```

`\thispagestyle` $\{\langle style \rangle\}$

```
244 \let\LWR@origthispagestyle\thispagestyle
245 \renewcommand*{\thispagestyle}[1]{}
```

`\pagenumbering` $\{\langle commands \rangle\}$

```
246 \let\LWR@origpagenumbering\pagenumbering
247 \renewcommand*{\pagenumbering}[1]{}
```

Pkg **xfrac**

Patched for HTML use. See section 156.

```
248 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
249 \RequirePackage{printlen}
```

```
250 \end{warpHTML}
```

17 Loading Packages

for HTML output: 251 \begin{warpHTML}

Remember the original \RequirePackage:

```
252 \let\LWR@origRequirePackage\RequirePackage
```

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

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

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

Note that argument 4 is passed directly to \StrBetween.

```
254 \newcommand*\LWR@findword[3][,]{%
255   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
256 }
```

\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.

```
257 \newcommand*\LWR@lookforpackagename[1]{
```

Find the n'th package name from the list:

```
258 \LWR@findword{\LWR@requirepackagenames}{#1}[\LWR@strresult]
```

See if the package name was found:

```
259 \IfStrEq{\LWR@strresult}{%
260 {}% no filename
261 {}% yes filename
```

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

```
262 \IfFileExists{lwarp-\LWR@strresult.sty}
263 {% latex_html_ file found
264 \StrSubstitute
265 {\LWR@requirepackagenames}
266 {\LWR@strresult}
267 {lwarp-\LWR@strresult}[\LWR@requirepackagenames]
268 }
269 {% no latex_html_* file
270 }% yes filename
271 }
```


`\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 L^AT_EX version.

```
272 \RenewDocumentCommand{\RequirePackage}{o m o}{%
```

Redirect up to nine names:

```
273 \renewcommand*{\LWR@requirepackagenames}{#2}
274 \LWR@lookforpackagename{1}
275 \LWR@lookforpackagename{2}
276 \LWR@lookforpackagename{3}
277 \LWR@lookforpackagename{4}
278 \LWR@lookforpackagename{5}
279 \LWR@lookforpackagename{6}
280 \LWR@lookforpackagename{7}
281 \LWR@lookforpackagename{8}
282 \LWR@lookforpackagename{9}
```

`\RequirePackage` depending on the options and version:

```
283 \IfValueTF{#1}
284 {% options given
285 \IfValueTF{#3}% version given?
286 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
287 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
288 }
289 {% no options given
290 \IfValueTF{#3}% version given?
291 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
292 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
293 }
294 }
295 \let\usepackage\RequirePackage
```

`\LWR@ProvidesPackagePass` {*pkgnam*e} [*version*]

Uses the original package, including options.

```
296 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
297 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
298 \IfValueTF{#2}
299 {\ProvidesPackage{lwarp-#1}[#2]}
300 {\ProvidesPackage{lwarp-#1}}
301 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
302 \ProcessOptions\relax
303
304 \IfValueTF{#2}
```

```

305 {\LWR@origRequirePackage{#1}[#2]}
306 {\LWR@origRequirePackage{#1}}
307 }

```

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

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

```

308 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
309 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
310 \IfValueTF{#2}
311 {\ProvidesPackage{lwarp-#1}[#2]}
312 {\ProvidesPackage{lwarp-#1}}
313 \DeclareOption*{}
314 \ProcessOptions\relax
315 }

316 \end{warpHTML}

```

18 Copying a file

for HTML output: 317 \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 sideTOC navigation pane.

```

318 \newcommand*{\LWR@copyfile}[2]{%
319 \newwrite\copyfile % open the file to write to
320 \immediate\openout\copyfile=#2
321 \newread\file % open the file to read from
322 \openin\file=#1
323 \begingroup\endlinechar=-1
324 \makeatletter
325 \loop\unless\ifeof\file
326 \read\file to\fileline % Read one line and store it into \fileline
327 % \fileline\par % print the content into the pdf
328 % print the content:
329 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
330 \repeat
331 \closeout\copyfile
332 \endgroup
333 }

```

```
334 \end{warpHTML}
```

19 Debugging messages

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

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

`\tracinglwarp` Turns on the debug tracing messages.

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

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

```
337 \newcommand{\LWR@traceinfo}[1]{%
338 \ifbool{LWR@tracinglwarp}%
339 {%
340 \typeout{*** lwarp: #1}%
341 % \PackageInfo{lwarp}{#1 : }%
342 }%
343 {}%
344 }
```

20 Remembering original formatting macros

for HTML output:

```
345 \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.

```
346 \let\LWR@origtextit\textit
347 \let\LWR@origtextbf\textbf
348 \let\LWR@origtexttt\texttt
349 \let\LWR@origtextsc\textsc
350 \let\LWR@origtextsf\textsf
351 \let\LWR@origtextrm\textrm
352 \let\LWR@origbfseries\bfseries
353 \let\LWR@origrmfamily\rmfamily
354 \let\LWR@origttfamily\ttfamily
355 \let\LWR@orignormalfont\normalfont
356
```

```

357 \let\LWR@origraggedright\raggedright
358 \let\LWR@origonecolumn\onecolumn
359
360 \let\LWR@origtextsuperscript\textsuperscript
361 \let\LWR@origtextsubscript\textsubscript
362
363 \let\LWR@origscriptsize\scriptsize
364
365 \let\LWR@orignewpage\newpage
366
367 \let\LWR@origminipage\minipage
368 \let\LWR@origendminipage\endminipage
369
370 \let\LWR@orignewline\newline
371
372 \let\LWR@origitem\item
373
374 \let\LWR@origpar\par
375
376
377 \let\LWR@origfootnote\footnote
378 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
379
380 \let\LWR@origclearpage\clearpage
381 \let\clearpage\relax
382 \let\cleardoublepage\relax
383 \end{warpHTML}

```

21 Operating-System portability

lwarp tries to detect which operating system is being used.

Prog	MS-Windows	If MS-WINDOWS is not correctly detected, use the command <code>\warpOSwindows</code> in the document preamble after <code>lwarp</code> is loaded. This modifies the operating-system path separator used by <code>lwarp</code> .
	Prog Windows	
	<code>\warpOSwindows</code>	

for HTML & PRINT: 384 \begin{warppall}

21.1 Common portability code

Bool	usingOSwindows	Set if <code>\warpOSwindows</code> .
------	----------------	--------------------------------------

```

385 \newbool{usingOSwindows}
386 \boolfalse{usingOSwindows}

```

21.2 Unix, Linux, and Mac OS

`\OSPathSymbol` Symbol used to separate directories in a path.

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

21.3 MS-Windows

For MS-Windows:

`\warpOSwindows` Set defaults for the MS-Windows operating system. `lwarp` attempts to auto-detect the operating system, so `\warpOSwindows` may not be necessary.

```
388 \newcommand*{\warpOSwindows}
389 {
390 \booltrue{usingOSwindows}
391 \renewcommand*{\OSPathSymbol}{\@backslashchar}
392 }
```

Test for windows during compile. The user may also specify `\warpOSwindows` later in case this test fails.

```
393 \ifwindows
394 \warpOSwindows
395 \fi
```

```
396 \end{warpall}
```

22 Stacks

for HTML output: 397 \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 (unnested) 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@closedeptune`, etc.

22.1 Assigning depths

initial depths for empty stack entries:

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

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

```
399 \newcommand*\LWR@depthfinished}{-4}
400 \newcommand*\LWR@depthpart}{-1}
401 \newcommand*\LWR@depthchapter}{0}
402 \newcommand*\LWR@depthsection}{1}
403 \newcommand*\LWR@depthsubsection}{2}
404 \newcommand*\LWR@depthsubsubsection}{3}
405 \newcommand*\LWR@depthparagraph}{4}
406 \newcommand*\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

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

22.2 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.

```
409 \newcommand*\LWR@closeone}{}% top of the stack
410 \newcommand*\LWR@closetwo){}
411 \newcommand*\LWR@closethree){}
412 \newcommand*\LWR@closefour){}
413 \newcommand*\LWR@closefive){}
414 \newcommand*\LWR@closesix){}
415 \newcommand*\LWR@closeseven){}
416 \newcommand*\LWR@closeeight){}
417 \newcommand*\LWR@closenine){}
418 \newcommand*\LWR@closeten){}
419 \newcommand*\LWR@closeeleven){}
420 \newcommand*\LWR@closetwelve){}
```

22.3 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}
```

```
421 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
422 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
423 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
424 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
425 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
426 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
427 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
428 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
429 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
430 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
431 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
432 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

22.4 Pushing and popping the stack

`\pushclose` $\{ \langle action \rangle \} \{ \langle depth \rangle \}$

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

```
433 \NewDocumentCommand{\pushclose}{m m}
434 {
435   \let\LWR@closetwelve\LWR@closeeleven
436   \let\LWR@closeeleven\LWR@closeten
437   \let\LWR@closeten\LWR@closenine
438   \let\LWR@closenine\LWR@closeeight
439   \let\LWR@closeeight\LWR@closeseven
440   \let\LWR@closeseven\LWR@closesix
441   \let\LWR@closesix\LWR@closefive
442   \let\LWR@closefive\LWR@closefour
443   \let\LWR@closefour\LWR@closethree
```

```

444 \let\LWR@closethree\LWR@closetwo
445 \let\LWR@closetwo\LWR@closeone
446 \let\LWR@closeone#1
447 \let\LWR@closedepthtwelve\LWR@closedeptheleven
448 \let\LWR@closedeptheleven\LWR@closedepthten
449 \let\LWR@closedepthten\LWR@closedepthnine
450 \let\LWR@closedepthnine\LWR@closedeptheight
451 \let\LWR@closedeptheight\LWR@closedepthseven
452 \let\LWR@closedepthseven\LWR@closedepthsix
453 \let\LWR@closedepthsix\LWR@closedepthfive
454 \let\LWR@closedepthfive\LWR@closedepthfour
455 \let\LWR@closedepthfour\LWR@closedepththree
456 \let\LWR@closedepththree\LWR@closedepthtwo
457 \let\LWR@closedepthtwo\LWR@closedepthone
458 \let\LWR@closedepthone#2
459 }

```

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

```

460 \newcommand*{\popclose}
461 {
462 \let\LWR@closeone\LWR@closetwo
463 \let\LWR@closetwo\LWR@closethree
464 \let\LWR@closethree\LWR@closefour
465 \let\LWR@closefour\LWR@closefive
466 \let\LWR@closefive\LWR@closesix
467 \let\LWR@closesix\LWR@closeseven
468 \let\LWR@closeseven\LWR@closeeight
469 \let\LWR@closeeight\LWR@closenine
470 \let\LWR@closenine\LWR@closeten
471 \let\LWR@closeten\LWR@closeeleven
472 \let\LWR@closeeleven\LWR@closetwelve
473 \let\LWR@closedepthone\LWR@closedepthtwo
474 \let\LWR@closedepthtwo\LWR@closedepththree
475 \let\LWR@closedepththree\LWR@closedepthfour
476 \let\LWR@closedepthfour\LWR@closedepthfive
477 \let\LWR@closedepthfive\LWR@closedepthsix
478 \let\LWR@closedepthsix\LWR@closedepthseven
479 \let\LWR@closedepthseven\LWR@closedeptheight
480 \let\LWR@closedeptheight\LWR@closedepthnine
481 \let\LWR@closedepthnine\LWR@closedepthten
482 \let\LWR@closedepthten\LWR@closedeptheleven
483 \let\LWR@closedeptheleven\LWR@closedepthtwelve
484 }

485 \end{warpHTML}

```


23 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}{\relax}
```

for HTML output: 486 \begin{warpHTML}

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

```
487 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
488 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
489 }
```

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

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

```
491 \end{warpHTML}
```

24 HTML entities

for HTML output: 492 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
493 \let\LWR@origampersand\&
```

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

```
494 \newcommand*{\HTMLentity}[1]{\LWR@origampersand#1;}
```

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

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

```
\&
```

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

```

\textless
\textgreater
497 \let\LWR@origtextless\textless
498 \renewcommand*{\textless}{\HTMLentity{lt}}
499
500 \let\LWR@origtextgreater\textgreater
501 \renewcommand*{\textgreater}{\HTMLentity{gt}}

502 \end{warpHTML}

```

25 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: 503 \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.

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

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

```
505 \providecommand*{\HTMLFileName}{}
```

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

```
506 \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.

```

507 \newcommand*{\SetHTMLFileNumber}[1]{%
508 \setcounter{LWR@htmlfilenumber}{#1}%
509 }

```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
510 \newbool{FileSectionNames}
511 \booltrue{FileSectionNames}

512 \end{warpall}
```

for HTML output: 513 \begin{warpHTML}

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

```
514 \newcounter{LWR@htmlfilenumber}
515 \setcounter{LWR@htmlfilenumber}{0}
```

\LWR@htmlsectionfilename *{(htmlfilenumber or name)}*

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

```
516 \newcommand*{\LWR@htmlsectionfilename}[1]{%
517 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
518 \LWR@traceinfo{about to assign temp}%
519 \edef\LWR@tempone{#1}%
520 \LWR@traceinfo{about to compare with ??}%
521 \ifthenelse{\equal{\LWR@tempone}{??}}{%
522 {%
523 \LWR@traceinfo{found ??}%
524 }{%
525 \LWR@traceinfo{not found ??}%
526 }%
527 \LWR@traceinfo{about to compare with zero or empty}%
528 \ifthenelse{%
529 \equal{\LWR@tempone}{0}}%
530 \OR \equal{\LWR@tempone}{}}%
531 \OR \equal{\LWR@tempone}{??}}%
532 }%
533 {%
534 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
535 \HomeHTMLFileName.html%
536 }%
```

For a L^AT_EX section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```

537 {%
538 \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
539 \ifthenelse{%
540 \equal{\HTMLFileName}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
541 }%
542 {%
543 \LWR@traceinfo{prefixing the index name with an underscore.}%
544 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

545 {\HTMLFileName#1.html}%
546 }%
547 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
548 }

```

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

Prints the filename for the given label

```

549 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
550 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: '#1!}%
551 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
552 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
553 }

554 \end{warpHTML}

```

26 Homepage link

for HTML output: 555 `\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.

```

556 \newcommand*{\LinkHome}{%
557 \LWR@subhyperrefclass{%
558 \HomeHTMLFileName.html}%
559 {Home}{linkhome}%
560 }

```

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

```
561 \newcommand*{\LWR@topnavigation}{
562 \LWR@html-elementclassline{nav}{topnavigation}{\LinkHome}
563 }
```

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

```
564 \newcommand*{\LWR@botnavigation}{
565 \LWR@html-elementclassline{nav}{botnavigation}{\LinkHome}
566 }
```

```
567 \end{warpHTML}
```

27 \PrintStack diagnostic tool



Diagnostics tool: Prints the LaTeX 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: 568 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```
569 \newcommand*{\PrintStack}{
570 \LWR@startpars
571 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
572 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
573 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
574 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
575 }
```

```
576 \end{warpHTML}
```

28 Closing stack levels

for HTML output: 577 `\begin{warpHTML}`

Close one nested level:

```

578 \newcommand*{\LWR@closeoneprevious}{%
579
580 \LWR@closeone{
581
582 \popclose{
583 }

```

`\LWR@closeprevious` `{<depth>}` Close everything up to the given depth:

```

584 \newcommand*{\LWR@closeprevious}[1]{

```

Close any pending paragraph:

```

585 \LWR@stoppars

```

Close anything nested deeper than the desired depth:

```

586 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
587 }

```

```

588 \end{warpHTML}

```

29 HTML tags, spans, divs, elements

for HTML output: 589 \begin{warpHTML}

29.1 Mapping L^AT_EX Sections to HTML Sections

```

590 \newcommand*{\LWR@tagpart}{h2}
591 \newcommand*{\LWR@tagpartend}{/h2}
592 \newcommand*{\LWR@tagchapter}{h3}
593 \newcommand*{\LWR@tagchapterend}{/h3}
594 \newcommand*{\LWR@tagsection}{h4}
595 \newcommand*{\LWR@tagsectionend}{/h4}
596 \newcommand*{\LWR@tagsubsection}{h5}
597 \newcommand*{\LWR@tagsubsectionend}{/h5}
598 \newcommand*{\LWR@tagsubsubsection}{h6}
599 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
600 \newcommand*{\LWR@tagparagraph}{span class="paragraph">{}}
601 \newcommand*{\LWR@tagparagraphend}{/span}
602 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
603 \newcommand*{\LWR@tagsubparagraphend}{/span}
604
605 \newcommand*{\LWR@tagregularparagraph}{p}

```


29.2 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.

```
606
607 \newcommand*{\LWR@htmltagc}[1]{%
608 {%
609 \protect\LWR@origttfamily%
610 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
611 }%
612 }
```


Env `LWR@nestspan` Disable minipage, `\parbox` inside a `span`.

 `\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 `/span` or an extra `<p>` may appear.

```
613 \newenvironment*{LWR@nestspan}
614 {%
615 \addtocounter{LWR@spandepth}{1}%
616 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
617 }%
618 {\addtocounter{LWR@spandepth}{-1}}
```

`\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`.

```
619 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
620 \LWR@ensuredoingapar%
621 \LWR@htmltagc{#1}%
622 \begin{LWR@nestspan}%
623 #2%
624 \LWR@htmltagc{/#1}%
625 \end{LWR@nestspan}%
626 }
```

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

```
627 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
628 \LWR@ensuredoingapar%
```

```

629 \LWR@subhtmllementclass{span}{#1}{#2}%
630 \begin{LWR@nestspan}%
631 #3%
632 \LWR@htmltagc{/span}%
633 \end{LWR@nestspan}%
634 }

```

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

print an HTML tag: `<tag>`

```

635 \newcommand*{\LWR@htmltagb}[1]{%
636 \LWR@htmltagc{#1}%
637 \endgroup%
638 }
639
640 \newcommand*{\LWR@htmltag}{%
641 \begingroup\catcode'\_ =12
642 \LWR@htmltagb%
643 }

```

29.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
644 \newcommand*{\LWR@htmlopencomment}{%
645 % \LWR@origtexttt{\LWR@origtextbf{\LWR@origtextless{!}{-}{-}}}%
646 \begingroup\LWR@origttfamily\LWR@origtextless{!}{-}{-}\endgroup%
647 }
648
649 \newcommand*{\LWR@htmlclosecomment}{%
650 % \LWR@origtexttt{\LWR@origtextbf{{-}{-}\LWR@origtextgreater{}}}%
651 \begingroup\LWR@origttfamily{-}{-}\LWR@origtextgreater{}\endgroup%
652 }

```

`\LWR@htmlcomment` $\{\langle comment \rangle\}$

```

653 \newcommand{\LWR@htmlcomment}[1]{%
654 \LWR@htmlopencomment}%
655 \LWR@origtextrm{#1}%
656 \LWR@htmlclosecomment}%

```


`\LWR@htmlblockcomment` $\{\langle comment \rangle\}$

```

657 \newcommand{\LWR@htmlblockcommentb}[1]
658 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
659
660 \newcommand{\LWR@htmlblockcomment}
661 {%
662 \begingroup\catcode'\_ =12%
663 \LWR@htmlblockcommentb%
664 }
```

`\LWR@htmlblocktag` $\{\langle tag \rangle\}$ print a stand-alone HTML tag

```

665 \newcommand*\LWR@htmlblocktag}[1]{%
666 \LWR@stoppars%
667 \LWR@htmltag{#1}%
668 \LWR@startpars%
669 }
```

`\LWR@htmlblocktagcomment` $\{\langle tag \rangle\} \{\langle HTML comment \rangle\}$

```

670 \NewDocumentCommand{\LWR@htmlblocktagcomment}{m +m}{%
671 \LWR@stoppars%
672 \LWR@htmltag{#1}\LWR@htmlcomment{#2}\LWR@orignewline%
673 \LWR@startpars%
674 }
```

29.4 Div class and element class

`\LWR@subhtml elementclass` $\{\langle element \rangle\} \{\langle class \rangle\} [\langle style \rangle]$

Factored and reused in several places.

```

675 \NewDocumentCommand{\LWR@subhtml elementclass}{m m o}{%
676 \IfValueTF{#3}%
677 {% option
678 \ifthenelse{\equal{#3}{}}{%
679 {\LWR@htmltag{#1 class="#2"}}% empty option
680 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
681 }% option
682 {\LWR@htmltag{#1 class="#2"}}% no option
683 }
```

`\LWR@html elementclass` $\{\langle element \rangle\} \{\langle class \rangle\} [\langle style \rangle]$

```

684 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
685 \LWR@stoppars%
686 \LWR@subhtmlelementclass{#1}{#2}[#3]%
687 \LWR@startpars%
688 }

```

`\LWR@htmlelementclassend` $\{\langle element \rangle\} \{\langle class \rangle\}$

```

689 \newcommand*{\LWR@htmlelementclassend}[2]{%
690 \LWR@stoppars%
691 \LWR@htmltag{/#1}%
692 \LWR@htmlcomment{End of #1 ‘‘#2’’}%
693 \LWR@startpars%
694 }

```

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

```

695 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
696 \LWR@htmlelementclass{div}{#1}[#2]%
697 }

```

`\LWR@htmldivclassend` $\{\langle class \rangle\}$

```

698 \newcommand*{\LWR@htmldivclassend}[1]{%
699 \LWR@htmlelementclassend{div}{#1}%
700 }

```

29.5 Single-line elements

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

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

```

701 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
702 \LWR@stoppars
703 \LWR@subhtmlelementclass{#1}{#2}[#3]%
704 #4%
705 \LWR@htmltag{/#1}
706 \LWR@startpars
707 }

```

29.6 HTML5 semantic elements

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

```
708 \newcommand*\LWR@htmlelement}[1]{%
709 \LWR@htmlblocktag{#1}
710 }
```

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

```
711 \newcommand*\LWR@htmlelementend}[1]{%
712 \LWR@stoppars
713 \LWR@htmltag{/#1}
714 \LWR@startpars
715 }
716
717 \end{warpHTML}
```

29.7 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 [62](#).

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

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

for HTML output:

```
718 \begin{warpHTML}
719 \NewDocumentEnvironment{BlockClass}{m o}
720 {
721 \LWR@htmldivclass{#1}[#2]
722 }
723 {
724 \LWR@htmldivclassend{#1}
725 }
726 \end{warpHTML}
```

for PRINT output:

```
727 \begin{warpprint}
728 \NewDocumentEnvironment{BlockClass}{m o}{\}{\}
729 \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:

```

730 \begin{warpHTML}
731 \newcommand{\BlockClassSingle}[2]{%
732 \LWR@html@element@classline{div}{#1}{#2}%
733 }
734 \end{warpHTML}

```

for PRINT output:

```

735 \begin{warpprint}
736 \newcommand{\BlockClassSingle}[2]{#2}
737 \end{warpprint}

```

`\InlineClass` `{\langle class \rangle} [\langle style \rangle] {\langle text \rangle}` High-level interface for inline span classes.

for HTML output:

```

738 \begin{warpHTML}
739 \NewDocumentCommand{\InlineClass}{m o m}{%
740 \LWR@html@span@class{#1}{#2}{#3}%
741 }
742 \end{warpHTML}

```

for PRINT output:

```

743 \begin{warpprint}
744 \NewDocumentCommand{\InlineClass}{m o m}{#3}
745 \end{warpprint}

```

29.8 Closing HTML tags

for HTML output:

```

746 \begin{warpHTML}

```

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

```

747 \newcommand*{\LWR@printclosepart}
748     {\LWR@htmlcomment{Closing part}}
749 \newcommand*{\LWR@printclosechapter}
750     {\LWR@htmlcomment{Closing chapter}}
751 \newcommand*{\LWR@printclosesection}
752     {\LWR@htmlcomment{Closing section}}
753 \newcommand*{\LWR@printclosesubsection}
754     {\LWR@htmlcomment{Closing subsection}}
755 \newcommand*{\LWR@printclosesubsubsection}
756     {\LWR@htmlcomment{Closing subsubsection}}
757 \newcommand*{\LWR@printcloseparagraph}
758     {\LWR@htmlcomment{Closing paragraph}}
759 \newcommand*{\LWR@printclosesubparagraph}
760     {\LWR@htmlcomment{Closing subparagraph}}

```

Lists require closing HTML tags:

```

761 \newcommand*{\LWR@printcloselistitem}
762     {\LWR@htmltag{/li}}
763 \newcommand*{\LWR@printclosedescitem}
764     {\LWR@htmltag{/dd}}
765 \newcommand*{\LWR@printcloseitemize}
766     {\LWR@htmltag{/ul}}
767 \newcommand*{\LWR@printcloseenumerate}
768     {\LWR@htmltag{/ol}}
769 \newcommand*{\LWR@printclosedescription}
770     {\LWR@htmltag{/dl}}

771 \end{warpHTML}

```

30 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 31 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: 772 `\begin{warpHTML}`

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

```

773 \newcounter{LWR@spandepth}
774 \setcounter{LWR@spandepth}{0}

```

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

```

775 \newbool{LWR@doingstartpars}
776 \boolfalse{LWR@doingstartpars}

```

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

```

777 \newbool{LWR@doingapar}
778 \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:

```

779 \newcommand*{\LWR@ensuredoingapar}{%
780 \ifbool{\LWR@doingstartpars}%
781 {\global\booltrue{\LWR@doingapar}}%
782 {}%
783 }

```

\LWR@openparagraph

```

784 \newcommand*{\LWR@openparagraph}
785 {%

```

See if paragraph handling is enabled:

```

786 \ifbool{\LWR@doingstartpars}%
787 {% handling pars

```

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

```

788 \ifthenelse{%
789 \cnttest{\value{\LWR@lateximagedepth}}{>}{0} \OR%
790 \cnttest{\value{\LWR@spandepth}}{>}{0}%
791 }{% nested par tags?

```

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

```

792 {}% no nested par tags

```

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

```

793 {% yes nest par tags
794 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

L^AT_EX default list environments use `\@itemdepth` and `\@enumdepth`, but `lwarp` uses the `enumitem` package, which uses `\@listdepth`.

See if are nested inside an item list:

```

795 \ifnumcomp{\@listdepth}{>}{0}%
796 {%

```

If so, leave some horizontal room in the L^AT_EX PDF output for list labels:

```

797 \LWR@orighspace{1in}%
798 }{}%

```

Now have started a paragraph.

```
799 \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.)

```
800 \let\par\LWR@closeparagraph%
801 }% end of yes nest par tags
802 }% end of handling pars
803 {}% not handling pars
804 }
```

\LWR@closeparagraph

```
805 \newcommand*{\LWR@closeparagraph}
806 {%
```

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
808 {% handling pars
```

See if already started a lateximage or a span:

```
809 \ifthenelse{%
810 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
811 \cnttest{\value{LWR@spandepth}}{>}{0}%
812 }%
```

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

```
813 {% no nested par tags
814 \ifthenelse{%
815 \cnttest{\value{LWR@spandepth}}{>}{0}%
816 \AND%
817 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
818 }%
819 {%
820 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%
821 }%
822 {}%
823 }% no nested par tags
```

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

```
824 {% yes nest par tags
```

Print a closing tag:

```
825 \unskip%
826 \LWR@htmltagc{/LWR@tagregularparagraph}%
```

No longer doing a paragraph:

```
827 \global\boolfalse{LWR@doingapar}%
828 % Disable the special \env{minipage} \& \cs{hspace} interaction
829 % until a new minipage is found:
830 %   \begin{macrocode}
831 \global\boolfalse{LWR@minipagethispar}%
832 }% end of yes nest par tags
833 }% end of handling pars
```

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

```
834 {% not handling pars
835 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}{%
836 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
837 }%
838 }% not handling pars
```

Finish with regular paragraph processing

```
839 \LWR@origpar%
840 }
```

```
841 \end{warpHTML}
```

31 Paragraph start/stop handling

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

Section 30 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: 842 \begin{warpHTML}

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

```
843 \newcommand*{\LWR@startpars}%
844 {%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
846 {}%
```

If not currently in paragraph mode:

```
847 {%
```

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

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

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

```
849 \let\par\LWR@closeparagraph
850
851 }% an intentionally blank line
```

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

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

No `<par>` tag yet to undo:

```
853 \global\boolfalse{\LWR@doingapar}%
854 }
```

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

```
855 \newcommand*{\LWR@stoppars}%
856 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
858 {%
```

Print a closing tag:

```
859 \unskip%
860 \LWR@htmltagc{/\LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

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

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

```
862 \global\boolfalse{LWR@minipagethispar}
863
864 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
865 {}%
```

See if currently allowing HTML paragraphs:

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

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

```
867 {%
868 \ClearPreHook{par}%
869 }%
```

Else: do nothing

```
870 {}%
```

no longer in paragraph mode

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

no `<p>` tag to undo:

```
872 \global\boolfalse{LWR@doingapar}%
873 }
```

```
874 \end{warpHTML}
```

32 Page headers and footers

for HTML & PRINT: 875 \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.

```
876 \newcommand{\LWR@firstpagetop}{ } % for the home page alone
877 \newcommand{\LWR@pagetop}{ } % for all other pages
878 \newcommand{\LWR@pagebottom}{ }
879
880 \newcommand{\LWR@setfirstpagetop}[1]{%
881 \renewcommand{\LWR@firstpagetop}{#1}
882 \catcode'\_ =8
883 }
```

\SetFirstPageTop {*{text and logos}*}

```
884 \newcommand{\SetFirstPageTop}{%
885 \catcode'\_ =12
886 \LWR@setfirstpagetop
887 }
```

```
888 \newcommand{\LWR@setpagetop}[1]{%
889 \renewcommand{\LWR@pagetop}{#1}
890 \catcode'\_ =8
891 }
```

\SetPageTop {*{text and logos}*}

```
892 \newcommand{\SetPageTop}{%
893 \catcode'\_ =12
894 \LWR@setpagetop
895 }
```

```
896 \newcommand{\LWR@setpagebottom}[1]{%
897 \renewcommand{\LWR@pagebottom}{#1}
898 \catcode'\_ =8
899 }
```

\SetPageBottom {*{text and logos}*}

```
900 \newcommand{\SetPageBottom}{%
901 \catcode'\_ =12
902 \LWR@setpagebottom
903 }
```

904 \end{warpall}

33 CSS

for HTML output: 905 \begin{warpHTML}

\LWR@currentcss The CSS filename to use. This may be changed mid-document using **\NewCSS**, allowing different CSS files to be used for different sections of the document.

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

\NewCSS $\{ \langle new-css-filename.css \rangle \}$

Assigns the CSS file to be used by the following HTML pages.

```
907 \newcommand*{\LWR@newcssb}[1]{%
908 \renewcommand*{\LWR@currentcss}{#1}
909 \catcode'\_ =8
910 }
911
912 \newcommand*{\NewCSS}{
913 \catcode'\_ =12
914 \LWR@newcssb
915 }
916 \end{warpHTML}
```

for PRINT output: 917 \begin{warpprint}
 918 \newcommand*{\NewCSS}[1]{}
 919 \end{warpprint}

34 HTML meta description

for HTML output: 920 \begin{warpHTML}

\LWR@currentHTMLdescription The HTML meta description to use.

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using **\NewHTMLdescription**, allowing different HTML descriptions to be used for different sections of the document.

Do not use double quotes, and do not exceed 150 characters.

```
921 \newcommand{\LWR@currentHTMLdescription}{}
```

```
\NewHTMLdescription {⟨New HTML meta description.⟩}
```

Assigns the HTML file's description meta tag

```
922 \newcommand{\NewHTMLdescription}[1]{%
923 \renewcommand{\LWR@currentHTMLdescription}{#1}
924 }
925
926 \end{wrapHTML}
```

```
for PRINT output: 927 \begin{wrapprint}
928 \newcommand{\NewHTMLdescription}[1]{%
929 \end{wrapprint}
```

35 Footnotes

`lwrap` uses native \LaTeX footnote code, although with its own `\box` to avoid the \LaTeX 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.

35.1 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 [35.4](#) for the implementation.

35.2 Minipage footnotes

See section [61.2](#) for minipage footnotes.

35.3 Titlepage thanks

See section [42.6](#) for titlepage footnotes.

35.4 Regular page footnote implementation

for HTML output: 930 \begin{warpHTML}

Patch L^AT_EX footnotes to use a new \box for lwarp footnotes.

```
931 \newbox\LWR@footnotes
932 \setbox\LWR@footnotes=\vbox{}
```

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

```
933 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
934
935 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}%
936
937 \long\def\@footnotetext#1{%
938 \global\setbox\LWR@footnotes=\vbox{\unvbox\LWR@footnotes%
939 \protected@edef\@currentlabel{%
940 \csname p@footnote\endcsname\@thefnmark%
941 }% \@currentlabel
942 \color@begingroup%
943 \@makefntext{#1}%
```

A closing paragraph tag is required here:

```
944 \LWR@htmltagc{/\LWR@tagregularparagraph}%
945 \color@endgroup%
946 }% vbox
947 }%
948
949 \long\def\@mpfootnotetext#1{%
950 \global\setbox\@mpfootins\vbox{%
951 \unvbox\@mpfootins
952 \reset@font\footnotesize
953 \hsize\columnwidth
954 \@parboxrestore
955 \protected@edef\@currentlabel
956 {\csname p@mpfootnote\endcsname\@thefnmark}%
957 \color@begingroup
958 \@makefntext{%
959 \ignorespaces#1%
960 }%
```

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

```
961 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
962 {}%
963 {\LWR@htmltagc{/\LWR@tagregularparagraph}}%
```

```

964 \color@endgroup}%
965 }

```

Enclose the footnotes in a class, print, then clear:

```

966 \newcommand*{\LWR@printpendingfootnotes}{%
967 \begin{BlockClass}{footnotes}
968 \LWR@originedskip
969 \unvbox\LWR@footnotes
970 \setbox\LWR@footnotes=\vbox{}
971 \end{BlockClass}
972 }

973 \end{warpHTML}

```

36 Marginpars

for HTML output: 974 \begin{warpHTML}

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

```

975 \renewcommand{\marginpar}[2] [] {%
976 \LWR@htmlspanclass{marginpar}{#2}%
977 }

978 \end{warpHTML}

```

37 Splitting HTML files

- Files are split according to FileDepth and CombineHigherDepths.
- Filenames are sanitized by \LWR@filenamenoblanks.
- \LWR@newhtmlfile finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT: 979 \begin{warpall}

Ctr FileDepth {*section depth*} determines how deeply to break into new HTML files, similar to tocdepth. The default of -5 produces one large HTML file.

```

980 \newcounter{FileDepth}
981 \setcounter{FileDepth}{-5}

```

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

```
982 \newbool{CombineHigherDepths}
983 \booltrue{CombineHigherDepths}

984 \end{warpall}
```

for HTML output: 985 \begin{warpHTML}

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

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

\LWR@thisnewfilename The filename being sanitized.

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

\LWR@filenamenoblanks $\{\langle filename \rangle\}$

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 L^AT_EX commands which appear in section names and TOC captions.

```
988 \newcommand*{\LWR@filenamenoblanks}[1]{%
989 \begingroup
```

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

```
990 \renewcommand*{\HTMLunicode}[1]{%
991 \renewcommand*{\HTMLentity}[1]{##1}
992 \renewcommand*{\LWR@htmltagc}[1]{%
993 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
994 \renewcommand*{\&}{and}
995 \renewcommand{\textit}[1]{##1}
996 \renewcommand{\textsc}[1]{##1}
997 \renewcommand{\textsl}[1]{##1}
998 \renewcommand{\textbf}[1]{##1}
999 \renewcommand{\texttt}[1]{##1}
1000 \renewcommand{\textsf}[1]{##1}
1001 \renewcommand{\textrm}[1]{##1}
```



```
1002 \renewcommand{\textsuperscript}[1]{##1}
1003 \renewcommand{\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
1004 \edef\LWR@thisnewfilename{#1}
1005 \fullexpandarg
```

Convert spaces into hyphens:

```
1006 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
1007 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
1008 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
1009 \StrSubstitute{\LWR@thisnewfilename}%
1010 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
1011 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
1012 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
1013 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
1014 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
1015 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
1016 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
1017 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
1018 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
1019 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
1020 \StrSubstitute{\LWR@thisnewfilename}%
1021 {\textless}{-}[\LWR@thisnewfilename]
1022 \StrSubstitute{\LWR@thisnewfilename}%
1023 {\textgreater}{-}[\LWR@thisnewfilename]
1024 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
1025 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
1026 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
1027 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
1028 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
1029 \StrSubstitute{\LWR@thisnewfilename}%
1030 {\textbackslash}{-}[\LWR@thisnewfilename]
1031 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
1032 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
1033 %      "~{" for babel
1034 \StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
1035 \StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
1036 \StrSubstitute{\LWR@thisnewfilename}{' }{-}[\LWR@thisnewfilename]
```

Convert short words:

```
1037 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
1038 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
```

```

1039 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
1040 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
1041 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
1042 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
1043 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
1044 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
1045 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
1046 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
1047 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
1048 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
1049 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
1050 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
1051 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
1052 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1053 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
1054 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

1055 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
1056 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
1057 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
1058 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1059 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1060 %      emdash
1061 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1062 %      endash
1063 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
1064 \endgroup
1065 }

```

`\LWR@newhtmlfile` $\{ \langle section\ name \rangle \}$

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.

```

1066 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```

1067 \LWR@htmlElementclassend{section}{textbody}
1068
1069 \LWR@printpendingfootnotes
1070
1071 \LWR@htmlElement{footer}
1072
1073 \LWR@pagebottom

```

```
1074
1075 \LWR@html elementend{footer}
```

No bottom navigation if are finishing the home page: Presumably has a table-of-contents printed.

```
1076 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

End of this HTML file:

```
1077 \LWR@stoppars
1078 \LWR@htmltag{/body}\LWR@orignewline
1079 \LWR@htmltag{/html}\LWR@orignewline
1080
1081 \addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into `\LWR@thisfilename`. If not using a filename, the file number will be used instead.

```
1082 \ifbool{FileSectionNames}%
1083 {\LWR@filenamenoblanks{#1}}
1084 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilenumber}}
```

Include an HTML comment to instruct `lwarpmk` where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
1085 \LWR@htmlblockcomment{%
1086 \LWR@origtexttt{|Start file|}
1087 \LWR@htmlsectionfilename{\LWR@thisfilename}|}%
1088 }
```

At the top of the starting file:

```
1089 \LWR@stoppars
1090
1091 \LWR@filestart{ -- #1}% there is an EMdash in front of the #1
1092
1093 \LWR@topnavigation
1094
1095 \LWR@html element{header}
1096
1097 \LWR@pagetop
1098
1099 \LWR@html elementend{header}
1100
1101 \ifcvoid{thetitle}{ }\LWR@printthetitle}
```

```

1102
1103 \LWR@sidetoc
1104
1105 \LWR@htmlelementclass{section}{textbody}
1106
1107 \LWR@stoppars
1108
1109 \setcounter{LWR@latestautopage}{\value{page}}
1110 }

1111 \end{warpHTML}

```

38 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing L^AT_EX code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

Unicode Section names and the resulting filenames with accented characters are partially supported, depending on the ability of `pdflatex` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that `pdflatex` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.



X_YL^AT_EX and LuaL^AT_EX directly support accented section and file names.

for HTML output: 1112 \begin{warpHTML}

38.1 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.



```

1113 \newbool{LWR@mainmatter}
1114 \DeclareDocumentCommand{\mainmatter}{}{%
1115 \booltrue{LWR@mainmatter}%
1116 }

```

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

```

1117 \DeclareDocumentCommand{\frontmatter}{}{%
1118 \boolfalse{LWR@mainmatter}%
1119 }

```

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

```
1120 \DeclareDocumentCommand{\backmatter}{-}{-}{%
1121 \boolfalse{LWR@mainmatter}
1122 }
```

38.2 Sectioning support macros

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

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

```
1123 \newcommand*{\LWR@sectionnumber}[1]{%
1124 \InlineClass{sectionnumber}{#1} %
1125 }
```

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

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

Create an autosection tag.

```
1126 \newcommand*{\LWR@createautosec}[1]{%
1127 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
1128 }
```

`\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.

```
1129 \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.

```
1130 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1131 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1132 \LWR@pushoneclose{#1}{#2}%
1133 }
```

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.

```
1134 \newcounter{LWR@prevFileDepth}
1135 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\LWR@section` * [*TOC name*] {*name*} {*sectiontype*}

The common actions for the high-level sectioning commands.

```
1136 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
1137 \LWR@traceinfo{LWR@section}%
1138 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
1139 \global\boolfalse{LWR@minipagethispar}%
```

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

```
1140 \LWR@orignewpage%
1141
```

Start a new HTML file if not starred, and is a shallow sectioning depth:

```
1142 \LWR@traceinfo{LWR@section: testing about to start a new HTML file}%
1143 \IfBooleanTF{#1}{\% not starred
1144 \ifthenelse{%
1145 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
1146 \AND%
1147 \(%
1148 \NOT\boolean{CombineHigherDepths}\OR%
1149 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
1150 \)%
1151 }%
```

If so: start a new HTML file:

```
1152 {\% new file
1153 \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
1154 \IfNoValueTF{#2}%
```

If no optional entry

```
1155 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
1156 {\LWR@newhtmlfile{#2}}%
```

```
1157 }% new file
```

Else: No new HTML file:

```
1158 {}% not new file
```

```
1159 }% not starred
```

Remember this section's name for \nameref:

```
1160 \LWR@traceinfo{\LWR@section: about to LWR@setlatestname}%
```

```
1161 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
```

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

```
1162
```

```
1163 \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}%
```

```
1164
```

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

```
1165 \ifthenelse{%
```

```
1166 \cnttest{\csuse{\LWR@depth#4}}{>=}{\LWR@depthparagraph}%
```

```
1167 }%
```

```
1168 {\LWR@startpars}
```

```
1169 {}
```

Create the opening tag with an autosec:

```
1170 \LWR@createautosec{\csuse{\LWR@tag#4}}%
```

If not starred, step counter and add to TOC:

```
1171 \IfBooleanTF{#1}%
```

```
1172 {}% starred
```

```
1173 {}% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```

1174 \ifthenelse{%
1175 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1176 }%
1177 {% if secnumdepth

```

If in the main matter, step the counter and add the TOC entry. For `article` class, `lwarp` assumes that all is mainmatter.

```

1178 \LWR@traceinfo{LWR@section: about to test main matter}%
1179 \ifbool{LWR@mainmatter}%
1180 {%
1181 \LWR@traceinfo{LWR@section: yes mainmatter}%
1182 \refstepcounter{#4}%

```

Add main matter numbered TOC entry with the TOC name or the regular name:

```

1183 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1184 \addcontentsline{toc}{#4}%
1185 {\protect\LWR@sectionnumber{\csuse{the#4}}%
1186 \IfValueTF{#2}{#2}{#3}}%
1187 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1188 }% end of if main matter

```

If not main matter, add unnumbered TOC name or regular name:

```

1189 {% not main matter
1190 \LWR@traceinfo{LWR@section: no main matter}%
1191 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1192 }% end of not main matter
1193 }% end of secnumdepth

```

Deeper than `secnumdepth`, so add an unnumbered TOC entry:

```

1194 {%
1195 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1196 }%

```

For part, print the section type:

```

1197 \ifbool{LWR@mainmatter}%
1198 {%
1199 \ifthenelse{%
1200 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
1201 {\value{secnumdepth}}\) \AND
1202 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
1203 }%
1204 {\csuse{#4name}~{}}%
1205 {}%

```


Print the section number:

```

1206 \LWR@traceinfo{LWR@section: about to print section number}%
1207 \ifthenelse{%
1208 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}}%
1209 }%
1210 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1211 {}%
1212 \LWR@traceinfo{LWR@section: finished print section number}%
1213 }{}%
1214 }% end of not starred

```

Print the section name:

```

1215 #3

```

close the heading tag, such as /H2

```

1216 \LWR@htmltag{\csuse{LWR>tag#4end}}%

```

Generate a L^AT_EX label:

```

1217 \label{autopage-\thepage}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

1218 \ifthenelse{%
1219 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
1220 {\LWR@startpars}%
1221 {}%

```

If not starred, remember the previous depth to possibly trigger a new HTML page.

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.

```

1222 \IfBooleanTF{#1}{\% not starred
1223 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
1224 }% not starred
1225 \LWR@traceinfo{LWR@section: done}%
1226 }

```

38.3 \section and friends

`\part` * [*TOC name*] {*name*}

```

1227 \@ifundefined{chapter}
1228 {}
1229 {%
1230 \DeclareDocumentCommand{\part}{s o m}{%
1231 \LWR@stoppars%
1232
1233 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
1234
1235 \LWR@section{#1}{#2}{#3}{part}%
1236 }
1237 }
```

`\chapter` * [*TOC name*] {*name*}

```

1238 \@ifundefined{chapter}
1239 {}
1240 {%
1241 \DeclareDocumentCommand{\chapter}{s o m}{%
1242 \LWR@traceinfo{chapter #3}%
1243 \LWR@stoppars%
1244 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
1245
1246 \LWR@traceinfo{chapter: about to LWR@section}%
1247 \LWR@section{#1}{#2}{#3}{chapter}%
1248 \LWR@traceinfo{chapter: done}%
1249 }
1250 }
```

`\section` * [*TOC name*] {*name*}

```

1251 \DeclareDocumentCommand{\section}{s o m}{%
1252 \LWR@stoppars%
1253
1254 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
1255
1256 \LWR@section{#1}{#2}{#3}{section}%
1257 }
```

`\subsection` * [*TOC name*] {*name*}

```

1258 \DeclareDocumentCommand{\subsection}{s o m}{%
1259 \LWR@stoppars%
1260 }
```

```

1261 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1262
1263 \LWR@section{#1}{#2}{#3}{subsection}%
1264 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

1265 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1266 \LWR@stoppars%
1267
1268 \LWR@startnewdepth{\LWR@depthsubsubsection}%
1269 {\LWR@printclosesubsubsection}%
1270
1271 \LWR@section{#1}{#2}{#3}{subsubsection}%
1272 }

```

`\paragraph` * [*TOC name*] {*name*}

```

1273 \DeclareDocumentCommand{\paragraph}{s o m}{%
1274 \LWR@stoppars%
1275
1276 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1277
1278 \LWR@section{#1}{#2}{#3}{paragraph}%
1279 }

```

`\subparagraph` * [*TOC name*] {*name*}

```

1280 \DeclareDocumentCommand{\subparagraph}{s o m}{%
1281 \LWR@stoppars%
1282
1283 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
1284
1285 \LWR@section{#1}{#2}{#3}{subparagraph}%
1286 }

1287 \end{warpHTML}

```

39 Starting a new file

for HTML output: 1288 \begin{warpHTML}

`\MetaLanguage` Default language for the HTML lang tag.

```

1289 \newcommand*{\MetaLanguage}{en-US}

```

`\LWR@filestart` $\{\langle title_suffix \rangle\}$

Creates the opening HTML tags.

1290 `\newcommand*\LWR@filestart}[1]{`

Locally temporarily disable direct-formatting commands:

```

1291 \begingroup
1292 \renewcommand{\textit}[1]{##1}% not used in filenames
1293 \renewcommand{\textsc}[1]{##1}
1294 \renewcommand{\textsl}[1]{##1}
1295 \renewcommand{\textbf}[1]{##1}
1296 \renewcommand{\texttt}[1]{##1}
1297 \renewcommand{\textsf}[1]{##1}
1298 \renewcommand{\textrm}[1]{##1}
1299 \renewcommand{\textsuperscript}[1]{##1}
1300 \renewcommand{\textsubscript}[1]{##1}
1301 \renewcommand*{\HTMLunicode}[1]{ }
1302 \renewcommand*{\HTMLentity}[1]{ }
1303 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1304 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}

```

Create the page's HTML header:

1305 `\LWR@htmltag{!DOCTYPE html}\LWR@orignewline`

The language is user-adjustable:

1306 `\LWR@htmltag{html lang="\MetaLanguage"{}\LWR@orignewline`

Start of the meta data:

1307 `\LWR@htmltag{head}\LWR@orignewline`

Charset is fixed at UTF-8:

1308 `\LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline`

lwarp is the generator:

```

1309 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\%
1310     \LWR@orignewline

```

If there is a description, add it now:

```

1311 \ifdefempty{\LWR@currentHTMLdescription}{\%
1312 \LWR@htmltag{\%

```

```

1313 meta name="description" content="\LWR@currentHTMLdescription" /}%
1314     \LWR@orignewline
1315 }%

```

Mobile-friendly viewport:

```

1316 \LWR@htmltag{meta name="viewport" %
1317 content="width=device-width, initial-scale=1.0" /}%
1318     \LWR@orignewline

```

IE patch:

```

1319 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
1320 \LWR@htmltag{%
1321 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
1322 \LWR@htmltag{/script}\LWR@orignewline
1323 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline

```

The page's title:

```

1324 \ifcsvoid{thetitle}{}{%
1325 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
1326 }%

```

The page's stylesheet:

```

1327 \LWR@htmltag{%
1328 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1329 \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.

```

1330 \ifbool{mathjax}%
1331 {%
1332 \boolfalse{LWR@verbtags}
1333 \VerbatimInput{lwarp_mathjax.txt}%
1334 \booltrue{LWR@verbtags}
1335 \LWR@stoppars
1336 }% end of mathjax
1337 {}%

```

End of the header:

```

1338 \LWR@htmltag{/head}\LWR@orignewline

```

Start of the body:

```

1339 \LWR@htmltag{body}\LWR@orignewline
1340 \endgroup
1341 }

1342 \end{warpHTML}

```

40 Starting HTML output

for HTML output: 1343 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```

1344 \newcommand*{\LWR@LwarpStart}
1345 {%

```

Expand and detokenize \HomeHTMLFileName and \HTMLFileName:

```

1346 \edef\LWR@strresult{\HomeHTMLFileName}
1347 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1348 \edef\LWR@strresult{\HTMLFileName}
1349 \edef\HTMLFileName{\detokenize\expandafter{\LWR@strresult}}

```

Force onecolumn:

```

1350 \LWR@origonecolumn%

```

Reduce chance of line overflow in verbatim environments:

```

1351 \LWR@origscriptsize%

```

In PDF output, don't allow line breaks to interfere with HTML tags:

```

1352 \LWR@origraggedright%
1353 \let\\\LWR@endoffline%

```

Spread the lines for pdf_{totext} to read them well:

```

1354 \linespread{1.3}%

```

For pdf_{totext} to reliably identify paragraph splits:

```

1355 \setlength{\parindent}{0pt}
1356 \setlength{\parskip}{2ex}

```

For the `lateximages` record file:

```
1357 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1358 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
1359 \renewcommand{\ps@plain}{} 
```

```

\centering Not used in the HTML environment:
\raggedleft
\raggedright 1360 \renewcommand*{\centering}{}
               1361 \renewcommand*{\raggedleft}{}
               1362 \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:

```

1363 \let\LWR@origtabular\tabular
1364 \let\LWR@origendtabular\endtabular
1365 \let\tabular\LWR@tabular
1366 \let\endtabular\endLWR@tabular

```

Float captions:

```
1367 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by [Label in HTML](#) `cleveref`.

```

1368 \let\LWR@origltx@label\ltx@label
1369 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

1370 \let\TeX\LWR@TeX
1371 \let\LaTeX\LWR@LaTeX
1372 \let\LuaTeX\LWR@LuaTeX
1373 \let\LuaLaTeX\LWR@LuaLaTeX
1374 \let\XeTeX\LWR@XeTeX
1375 \let\XeLaTeX\LWR@XeLaTeX
1376 \let\ConTeXt\LWR@ConTeXt

```

Graphics:

```
1377 \let\rotatebox\LWR@rotatebox
1378 \let\scalebox\LWR@scalebox
1379 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
1380 \global\boolfalse{LWR@doingapar}
1381 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
1382 \LWR@filestart{}
1383 \LWR@htmltag{header}\LWR@orignewline
1384 \LWR@startpars
1385 \LWR@firstpagetop
1386 \LWR@stoppars
1387 \LWR@htmltag{/header}\LWR@orignewline
1388 \LWR@htmltag{section class="textbody"{} }
1389 \LWR@origpagestyle{empty}
```

Document and page settings:

```
1390 \mainmatter
1391 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section [42.6](#).

```
1392 \if@titlepage
1393   \thanksmarkseries{arabic}
1394 \else
1395   \thanksmarkseries{fnsymbol}
1396 \fi
```

Initial default patch for fancyvrb:

```
1397 \fvset{frame=none}%
```

The ampersand is redefined active, and acts depending on whether it is inside a tabular.

```
1398 \catcode'\&=\active
```

Allow HTML paragraphs to begin:

```
1399 \LWR@startpars
1400 }
```



```
1401 \end{warpHTML}
```

41 Ending HTML output

for HTML output: 1402 \begin{warpHTML}

`\LWR@requesttoc` $\{\langle boolean \rangle\}$ $\{\langle suffix \rangle\}$ Requests that a toc, lof, or lot be generated.

```
1403 \newcommand*{\LWR@requesttoc}[2]{%
1404 \ifbool{#1}
1405 {
1406 \expandafter\newwrite\csuse{tf@#2}
1407 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
1408 }{}
1409 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
1410 \newcommand*{\LWR@LwarpEnd}
1411 {
1412 \LWR@stoppars
1413 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
1414 \LWR@html-elementclassend{section}{textbody}
```

Print any pending footnotes:

```
1415 \LWR@printpendingfootnotes
```

Create the footer:

```
1416 \LWR@html-element{footer}
1417
1418 \LWR@pagebottom
1419
1420 \LWR@html-elementend{footer}
```

No bottom navigation if are finishing the home page:
Presumably has a table-of-contents printed.

```
1421 \ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

1422 \LWR@stoppars% final stop of all paragraphs

Finish the HTML file:

1423 \LWR@htmltag{/body}\LWR@orignewline

1424 \LWR@htmltag{/html}\LWR@orignewline

Seems to be required sometimes:

1425 \LWR@orignewpage

For lateximage commands:

1426 \immediate\closeout\LWR@file

1427 }

1428 \end{warpHTML}

42 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:

`\maketitlehookaa` **\maketitlehookaa:** Between “published” and the title.

`\maketitlehookaaa` **\maketitlehookaaa:** Between the title and the subtitle.

`\prepublished` **\prepublished:** Before the “published” field.

`\postpublished` **\postpublished:** After the “published” field.

`\presubtitle` `\presubtitle:` Before the subtitle.

`\postsubtitle` `\postsubtitle:` After the subtitle.

`\printthanks` `\printthanks` has been added to force the printing of thanks inside a `titlingpage` environment when `\maketitle` 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}
...
```

42.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

`\published` `\@title`, `\@subtitle`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.

`\printpublished` `\printtitle`, `\prints subtitle`, etc. are user-level macros intended to be used in `titlepage` and `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, 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.

`\thepublished` `\thetitle`, `\thesubtitle`, `\theauthor`, etc. are user-level sanitized versions which have removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used

in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\SetPageBottom` $\{\langle text \rangle\}$

```
\SetPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

`\author` $\{\langle author \rangle\}$ 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` $\{\langle text \rangle\}$

Adds the affiliation to the author for use in `\maketitle`. Nullified when later used for inline use of `\theauthor`.

```
for HTML output: 1429 \begin{warpHTML}
                  1430 \newrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{#1}}
                  1431 \end{warpHTML}
```

```
for PRINT output: 1432 \begin{warpprint}
                  1433 \newrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
                  1434 \end{warpprint}
```

The following are based on the original titling code:

```
for HTML & PRINT: 1435 \begin{warpall}
```

`\author` $\{\langle text \rangle\}$

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.

```

1436 \renewcommand{\author}[1]{%
1437 \gdef\@author{#1}
1438 \begingroup
1439   \renewcommand{\thanks}[1]{ }
1440   \renewcommand{\and}{\unskip, }
1441   \renewcommand{\thanksmark}[1]{ }
1442   \renewcommand{\thanksgap}[1]{ }
1443   \renewcommand{\affiliation}[1]{ }
1444   \protected@xdef\theauthor{#1}
1445 \endgroup

```

`\published {text}`

```

1446 \newcommand{\published}[1]{%
1447 \gdef\@published{#1}
1448 \begingroup
1449   \renewcommand{\thanks}[1]{ }
1450   \renewcommand{\thanksmark}[1]{ }
1451   \renewcommand{\thanksgap}[1]{ }
1452   \protected@xdef\thepublished{#1}
1453 \endgroup
1454 }
1455 \newcommand{\@published}{}
1456 \newcommand{\thepublished}{}

```

`\subtitle {text}`

```

1457 \newcommand{\subtitle}[1]{%
1458 \gdef\@subtitle{#1}
1459 \begingroup
1460   \renewcommand{\thanks}[1]{ }
1461   \renewcommand{\thanksmark}[1]{ }
1462   \renewcommand{\thanksgap}[1]{ }
1463   \protected@xdef\thesubtitle{#1}
1464 \endgroup
1465 }
1466 \newcommand{\@subtitle}{}
1467 \newcommand{\thesubtitle}{}

1468 \end{warpall}

```

42.2 Changes to HTML titlepage and titlingpage

for HTML output: 1469 `\begin{warpHTML}`

Env `titlepage` Sets up a titlepage div with a L^AT_EX PDF minipage inside.

```
1470 \renewenvironment*{titlepage}
1471   {\BlockClass{titlepage}\LWR@subminipage}
1472   {\LWR@endsubminipage\endBlockClass}
```

Env `titlingpage`

```
1473 \renewenvironment*{titlingpage}
1474 {%
```

Start an HTML titlepage div:

```
1475 \begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
1476 \LWR@maketitlesetup
1477 \let\maketitle\LWR@titlingmaketitle
1478 }
1479 {
```

At the end of the environment, end the HTML titlepage div:

```
1480 \end{titlepage}
```

Reset the footnote counter:

```
1481 \@bscontmark
1482 }
```

```
1483 \end{warpHTML}
```

for HTML & PRINT: 1484 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```
1485 \newcommand*{\printthanks}{\@thanks}
```

Env `titlingpage` At the end of the `titlingpage` for both print and HTML, reset footnote markers to zero.

```
1486 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1487 \end{warpall}
```

42.3 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: 1488 \begin{warpHTML}

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
1489 \newcommand{\prepublished}[1]{%
1490 \def\@bspublished{\BlockClass{published}#1}%
1491 }
1492
1493 \newcommand{\postpublished}[1]{%
1494 \def\@bspublished{\BlockClass{published}#1}%
1495 }
1496
1497 \renewcommand{\pretitled}[1]{%
1498 \def\@bspublished{\BlockClass{published}#1}%
1499 }
1500
1501 \renewcommand{\posttitled}[1]{%
1502 \def\@bspublished{\BlockClass{published}#1}%
1503 }
1504
1505 \newcommand{\presubtitle}[1]{%
1506 \def\@bspublished{\BlockClass{published}#1}%
1507 }
1508
1509 \newcommand{\postsubtitle}[1]{%
1510 \def\@bspublished{\BlockClass{published}#1}%
1511 }
1512
1513 \renewcommand{\preauthor}[1]{%
1514 \def\@bspublished{\BlockClass{published}#1}%
1515 }
1516
1517 \renewcommand{\postauthor}[1]{%
1518 \def\@bspublished{\BlockClass{published}#1}%
1519 }
1520
1521 \renewcommand{\predated}[1]{%
1522 \def\@bspublished{\BlockClass{published}#1}%
```

```

1523 }
1524
1525 \renewcommand{\postdate}[1]{%
1526 \def\@bspostdate{\endBlockClass#1}%
1527 }
1528
1529 \prepublished{\begin{center}}
1530 \postpublished{\par\end{center}}
1531
1532 \pretitle{\begin{center}}
1533 \posttitle{\par\end{center}}
1534
1535 \presubtitle{\begin{center}}
1536 \postsubtitle{\par\end{center}}
1537
1538 \preauthor{\begin{center}%
1539 \begin{tabular}[t]{c}%
1540 }
1541 \postauthor{\end{tabular}\par\end{center}}
1542
1543 \predate{\begin{center}}
1544 \postdate{\par\end{center}}

```

\printpublished

```

1545 \newcommand*{\printpublished}{
1546 \ifthenelse{\equal{\thepublished}{}}{
1547 {}
1548 {
1549 \begin{BlockClass}{published}
1550 \@published
1551 \end{BlockClass}
1552 }
1553 }

```

\printtitle

```

1554 \newcommand*{\printtitle}
1555 {
1556 \LWR@stoppars
1557 \LWR@htmltag{h1}%
1558 \@title%
1559 \LWR@htmltag{/h1}
1560 \LWR@startpars
1561 }

```


`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
1562 \newcommand*{\LWR@printthetitle}
1563 {
1564 \LWR@stoppars
1565 \LWR@htmltag{h1}%
1566 \thetitle%
1567 \LWR@htmltag{/h1}
1568 \LWR@startpars
1569 }
```

`\printsubtitle`

```
1570 \newcommand*{\printsubtitle}{
1571 \ifthenelse{\equal{\thesubtitle}{}}{
1572 {}
1573 {
1574 \begin{BlockClass}{subtitle}
1575 \@subtitle
1576 \end{BlockClass}
1577 }
1578 }
```

`\printauthor`

```
1579 \newcommand*{\printauthor}{
1580 \begin{BlockClass}{author}
1581 \begin{tabular}{c}\@author\end{tabular}
1582 \end{BlockClass}
1583 }
```

`\printdate`

```
1584 \newcommand*{\printdate}{%
1585 \begin{BlockClass}{titledate}
1586 \@date
1587 \end{BlockClass}
1588 }
```

```
1589 \end{warpHTML}
```

42.4 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: 1590 \begin{warpprint}

\printpublished

1591 \newcommand*{\printpublished}{\Large\scshape\@published}}

\printtitle

1592 \newcommand*{\printtitle}{\Huge\@title}}

\printsubtitle

1593 \newcommand*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}}

\printauthor

1594 \newcommand*{\printauthor}

1595 {\large\begin{tabular}[t]{c}\@author\end{tabular}}}

\printdate

1596 \newcommand*{\printdate}{\small\textit{\@date}}}

42.5 \maketitle for print output

\maketitle From the titling package, patched to add the publisher and subtitle.

1597 \providecommand{\maketitle}{}%

1598 \if@titlepage

1599 \renewcommand{\maketitle}{\begin{titlepage}%

1600 \let\footnotesize\small

1601 \let\footnoterule\relax

1602 \let\footnote\thanks

1603 \@bsmarkseries

1604 \def\@makefnmark{\rlap{\@textsuperscript{%

1605 \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%

1606 \long\def\@makefntext##1{\makethanksmark ##1}

1607 \null\vfil

1608 \vskip 60\p@

1609 \vspace*{\droptitle}

1610 \maketitlehooka

1611 \ifcseempty{\@published}

1612 {}%

```

1613 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1614     {\@bsprettitle \@title \@bsposttitle}
1615     \ifcsemtty{\@subtitle}
1616 {}
1617 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubttile}}
1618     \maketitlehookb
1619     {\@bspreauthor \@author \@bspostauthor}
1620     \maketitlehookc
1621     {\@bspredate \@date \@bspostdate}
1622     \maketitlehookd
1623     \par
1624     \@thanks
1625     \vfil\null
1626     \end{titlepage}%
1627     \@bscontmark % \setcounter{footnote}{0}%
1628 %%% \@bsmttitleempty
1629 } % end titlepage defs
1630 \else
1631 \renewcommand{\maketitle}{\par
1632 \begin{group}
1633 \@bsmarkseries
1634 \def\@makefnmark{\rlap{\@textsuperscript{%
1635 \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1636 \long\def\@makefntext##1{\makethanksmark ##1}
1637 \if@twocolumn
1638 \ifnum \col@number=\@ne
1639 \@maketitle
1640 \else
1641 \twocolumn[\@maketitle]%
1642 \fi
1643 \else
1644 \newpage
1645 \global\@topnum\z@
1646 \@maketitle
1647 \fi
1648 \thispagestyle{plain}\@thanks
1649 \end{group}
1650 \@bscontmark % \setcounter{footnote}{0}%
1651 %%% \@bsmttitleempty
1652 } % end non-titlepage
1653
1654 \def\@maketitle{%
1655 \newpage
1656 \null
1657 \vskip 2em%
1658 \vspace*{\droptitle}
1659 \maketitlehooka
1660 \ifcsemtty{\@published}
1661 {}
1662 {{\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}

```

```

1663     {\@bspretitle \@title \@bsposttitle}
1664     \ifcsemtty{@subtitle}
1665 {}
1666 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1667     \maketitlehookb
1668     {\@bspreauthor \@author \@bspostauthor}
1669     \maketitlehookc
1670     {\@bspredate \@date \@bspostdate}
1671     \maketitlehookd
1672     \par
1673     \vskip 1.5em}
1674 \fi
1675
1676 \providecommand{\maketitlehookaa}{}
1677
1678 \providecommand{\maketitlehookaaa}{}
1679
1680 \newcommand{\prepublished}[1]{%
1681 \def\@bsprepublished{#1}%
1682 }
1683
1684 \newcommand{\postpublished}[1]{%
1685 \def\@bspostpublished{#1}%
1686 }
1687
1688 \newcommand{\presubtitle}[1]{%
1689 \def\@bspresubtitle{#1}%
1690 }

```

`\presubtitle` Hook after printing the subtitle.

```

1691 \newcommand{\postsubtitle}[1]{%
1692 \def\@bspostsubtitle{#1}%
1693 }

```

Initial settings:

```

1694 \if@titlepage
1695 \prepublished{
1696 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1697 \begin{center}}
1698 \postpublished{\par\end{center}\vskip 2em}
1699
1700 \presubtitle{\unskip\begin{center}\unskip}
1701 \postsubtitle{\par\end{center}\vskip 2em}
1702 \else
1703 \prepublished{\begin{center}}
1704 \postpublished{\par\end{center}\vskip 0.5em}
1705

```

```

1706 \presubtitle{\begin{center}\unskip}
1707 \postsubtitle{\par\end{center}\vskip 0.5em}
1708 \fi

1709 \end{warpprint}

```

42.6 \maketitle for HTML output

An HTML div of class `titlepage` is created, inside of which a L^AT_EX 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 35 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 1710 `\begin{warppHTML}`

`\LWR@maketitlesetup` Patches `\thanks` macros to use L^AT_EX minipage footnotes.

```
1711 \newcommand*\LWR@maketitlesetup{%
```

Select which kind of footnote marks to use:

```

1712 \@bsmarkseries
1713 \@mpbsmarkseries

```

Redefine the footnote mark:

```
1714 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

```

Redefine the footnote text:

```
1715 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
1716 \makethanksmark \LWR@origspace{1in}
```

`\makethanksmark` \Rightarrow `\thanksfootmark` \Rightarrow `\tamark` \Rightarrow
`\@thefnmark` \Rightarrow `\itshape` a (or similar)

Print the text:

```
1717 ##1%
1718 }%
1719 }
```

`\@fnsymbol` $\{ \langle counter \rangle \}$

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being found by `pdftotext`.

```
1720 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
1721 \mathsection\or \mathparagraph\or \text{\HTMLunicode{2016}}\or
1722 **\or \dagger\dagger \or \ddagger\ddagger \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.

```
1723 \renewcommand*\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
1724 \begin{titlepage}
```

Set up special patches:

```
1725 \LWR@maketitlesetup
```

Typeset the title, etc:

```
1726 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
1727 \@thanks
```

Close the HTML titlepage div:

```
1728 \end{titlepage}
```

Reset the footnote counter:

```
1729 \@bscontmark
1730 }
```

`\@maketitle` Typesets the title, etc. for HTML:

```

1731 \DeclareDocumentCommand{\@maketitle}{-}{%
1732 \maketitlehooka
1733 \ifcempty{@published}
1734 {}
1735 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1736 {\@bsprettitle \@title \@bsposttitle}
1737 \ifcempty{@subtitle}
1738 {}
1739 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1740 \maketitlehookb
1741 {\@bspreauthor \@author \@bspostauthor}
1742 \maketitlehookc
1743 {\@bspredate \@date \@bspostdate}
1744 \maketitlehookd
1745 }

1746 \providecommand{\maketitlehookaa}{}
1747 \providecommand{\maketitlehookaaa}{}

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```

1748 \newcommand*{\LWR@titlingmaketitle}{%

```

Typeset the title, etc:

```

1749 \@maketitle

```

Immediately generate any `\thanks` footnotes:

```

1750 \@thanks
1751 }

```

`\thanksmarkseries` `{\series}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc. Modified to use the L^AT_EX PDF minipage which is included with the title page.

```

1752 \renewcommand{\thanksmarkseries}[1]{%
1753 \def\@mpbsmarkseries{%
1754 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}%
1755 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
1756 }

1757 \end{warpHTML}

```

43 Abstract

The following code replaces the L^AT_EX default, and will itself be replaced later if the `abstract` package is loaded.

```

for HTML output: 1758 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

                Also over-written by the babel package.

1759 \providecommand*{\abstractname}{Abstract}

Env abstract

1760 \DeclareDocumentEnvironment{abstract}{}
1761 {
1762 \BlockClass{abstract}
1763 \BlockClassSingle{abstracttitle}{\abstractname}
1764 }
1765 {
1766 \endBlockClass
1767 }

1768 \end{warpHTML}

```

44 Quote and verse

44.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

```

for HTML output: 1769 \begin{warpHTML}
1770 \newcommand{\attribution}[1]{%
1771 \InlineClass{attribution}{--\,#1}}% emdash
1772 \end{warpHTML}

```

```

for PRINT output: 1773 \begin{warpprint}
1774 \newcommand{\attribution}[1]{\textsc{--\,#1}}
1775 \end{warpprint}

```

`\citetitle` for use inside quote, quotation, verse:


```

for HTML output: 1776 \begin{warpHTML}
                  1777 \newcommand{\citetitle}[1]{%
                  1778 \InlineClass{citetitle}{--\, #1}}% emdash
                  1779 \end{warpHTML}

for PRINT output: 1780 \begin{warpprint}
                  1781 \newcommand{\citetitle}[1]{\textsl{---\, #1}}
                  1782 \end{warpprint}

```

44.2 Quotes, quotations

```

for HTML output: 1783 \begin{warpHTML}

Env  quote

                  1784 \renewenvironment*{quote}
                  1785 {\LWR@htmlblocktag{blockquote}}
                  1786 {\LWR@htmlblocktag{/blockquote}}
                  1787
                  1788 \renewenvironment*{quotation}
                  1789 {\LWR@htmlblocktag{blockquotation}}
                  1790 {\LWR@htmlblocktag{/blockquotation}}

                  1791 \end{warpHTML}

```

44.3 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}

```

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\TMLvleftskip</code></p> <p>Len <code>\TMLleftmargini</code></p>	<p>These lengths are used by <code>verse</code> and <code>memoir</code> to control the left margin, and they may already be set by the user for print output. New lengths <code>\HTMLvleftskip</code> and <code>\HTMLleftmargini</code> are provided to control the margins in HTML output. These new lengths may be set by the user before any <code>verse</code> environment, and persist until they are manually changed again. One reason to change <code>\HTMLleftmargini</code></p>
--	---

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.

Horizontal spacing relies on `pdftotext`’s ability to discern the `-layout` 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: 1792 `\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.

```
1793 \newlength{\TMLvleftskip}
1794 \setlength{\TMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
1795 \newlength{\TMLleftmargini}
1796 \setlength{\TMLleftmargini}{4.5em}

1797 \end{warpall}
```

45 Verbatim

for HTML output: 1798 `\begin{warpHTML}`

Env `verbatim`

```
1799 \AfterEndPreamble{
1800 \AtBeginEnvironment{verbatim}{\LWR@atbeginverbatim{verbatim}\unskip\vspace*{--\baselineskip}}
1801 \AfterEndEnvironment{verbatim}{\unskip\vspace*{--\baselineskip}\LWR@afterendverbatim}
1802 }

1803 \end{warpHTML}
```

46 Fancyvrb

for HTML & PRINT: 1804 `\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.

```
1805 \newlength{\VerbatimHTMLWidth}
1806 \setlength{\VerbatimHTMLWidth}{4in}
1807 \end{warpall}
```

for HTML output: 1808 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing VerbatimInput in the HTML head.

```
1809 \newbool{LWR@verbtags}
1810 \booltrue{LWR@verbtags}
```

For \VerbatimFootnotes:

```
1811 \renewcommand{\VerbatimFootnotes}{
1812 \PackageError{lwarp}
1813 {Verbatim footnotes are not yet supported by lwarp.}
1814 {This may be improved some day.}
1815 }
```

\LWR@atbeginverbatim {\class}

Encloses a verbatim environment with the given CSS class.

```
1816 \newcommand*{\LWR@atbeginverbatim}[1]
1817 {%
```

Avoid excessive space between lines:

```
1818 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1819 \LWR@stoppars%
```

Create a new `pre` of the given class:

```
1820 \ifbool{LWR@verbtags}{\LWR@htmltag{pre class="#1"{}{}{}}{}}%
```

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.

```
1821 \LWR@origttfamily%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain L^AT_EX `\hspace` functionality:

```
1822 \let\hspace\LWR@orighspace%
1823 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
1824 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
1825 \unskip%
```

At the end of the environment, close the `pre`:

```
1826 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
1827
1828 }{}}%
```

Resume regular paragraph handling:

```
1829 \LWR@startpars%
1830 }
```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```
1831 \newcommand*{\LWR@Verbatimclass}{fancyvrb}
```

`Env VerbatimClass` `{\class}` [*Verbatim options*]

Creates a `Verbatim` enclosed in a `DIV` of the given class.

```
1832 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1833 {%
1834 \renewcommand*{\LWR@Verbatimclass}{#1}%
1835 \LWR@origVerbatim[#2]%
1836 }
1837 {\endVerbatim}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
1838 \AfterEndPreamble{
```

Remember the original definition of `Verbatim`:

```
1839 \let\LWR@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.

```
1840 \renewcommand*{\Verbatim}{%
1841 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
1842 \LWR@origVerbatim%
1843 }
```

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.

```
1844 \newcommand*{\LWR@fvstartnone}{%
1845 \LWR@traceinfo{fvstartnone}%
1846 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
1847 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
1848 }
1849
1850 \newcommand*{\LWR@fvendnone}{%
1851 \LWR@traceinfo{fvendnone}%
1852 \hbox to\z@{\LWR@afterendverbatim}%
1853 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
1854 }
1855
1856 \newcommand*{\LWR@fvstartsingle}{%
1857 \LWR@traceinfo{fvstartsingle}%
1858 \LWR@fvstartnone%
1859 \FV@BeginListFrame@Single%
1860 }
1861
1862 \newcommand*{\LWR@fvendsingle}{%
1863 \LWR@traceinfo{fvendsingle}%
1864 \FV@endListFrame@Single%
1865 \LWR@fvendnone%
1866 }
1867
1868 \newcommand*{\LWR@fvstartline}{%
1869 \LWR@traceinfo{fvstartline}%
1870 \LWR@fvstartnone%
1871 \FV@BeginListFrame@Lines%
1872 }
1873
1874 \newcommand*{\LWR@fvendline}{%
1875 \LWR@traceinfo{fvendline}%
1876 \FV@endListFrame@Lines%
1877 \LWR@fvendnone%
1878 }
```

The following patches select the start/left/right/end behaviors depending on **frame**.
Original code is from the fancyvrb package.

```

1879 \def\FV@Frame@none{%
1880 \let\FV@BeginListFrame\LWR@fvstartnone%
1881 \let\FV@LeftListFrame\relax%
1882 \let\FV@RightListFrame\relax%
1883 \let\FV@EndListFrame\LWR@fvendnone}
1884
1885 \def\FV@Frame@single{%
1886 \let\FV@BeginListFrame\LWR@fvstartsingle%
1887 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1888 \let\FV@RightListFrame\FV@RightListFrame@Single%
1889 \let\FV@EndListFrame\LWR@fvendsingle}
1890
1891 \def\FV@Frame@lines{%
1892 \let\FV@BeginListFrame\LWR@fvstartline%
1893 \let\FV@LeftListFrame\relax%
1894 \let\FV@RightListFrame\relax%
1895 \let\FV@EndListFrame\LWR@fvendline}
1896
1897 \def\FV@Frame@topline{%
1898 \let\FV@BeginListFrame\LWR@fvstartline%
1899 \let\FV@LeftListFrame\relax%
1900 \let\FV@RightListFrame\relax%
1901 \let\FV@EndListFrame\LWR@fvendnone}
1902
1903 \def\FV@Frame@bottomline{%
1904 \let\FV@BeginListFrame\LWR@fvstartnone%
1905 \let\FV@LeftListFrame\relax%
1906 \let\FV@RightListFrame\relax%
1907 \let\FV@EndListFrame\LWR@fvendline}
1908
1909 \def\FV@Frame@leftline{%
1910 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
1911 \ifx\FancyVerbFillColor\relax%
1912 \let\FV@FrameFillLine\relax%
1913 \else%
1914 \@tempdima\FV@FrameRule\relax%
1915 \multiply\@tempdima-\tw@%
1916 \edef\FV@FrameFillLine{%
1917 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
1918 \kern-\number\@tempdima sp}}%
1919 \fi%
1920 \let\FV@BeginListFrame\LWR@fvstartnone%
1921 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1922 \let\FV@RightListFrame\relax%
1923 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the `fancyvrb` package.

```

1924 \def\FV@SingleFrameLine#1{%
1925   \hbox to\z@{%
1926     %      \kern\leftmargin
1927     \ifnum#1=\z@\relax
1928       \let\FV@Label\FV@LabelBegin
1929     \else
1930       \let\FV@Label\FV@LabelEnd
1931     \fi
1932     \ifx\FV@Label\relax
1933 %       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
1934     \else
1935       \ifnum#1=\z@
1936 %       \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
1937       \ifx\FV@LabelPositionTopLine\relax
1938 \else
1939       \LWR@htmltagc{div class="fancyvrblabel"}
1940 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
1941 \LWR@htmltagc{/div}
1942 \fi
1943     \else
1944 %       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
1945       \ifx\FV@LabelPositionBottomLine\relax
1946 \else
1947       \LWR@htmltagc{div class="fancyvrblabel"}
1948 \LWR@origtextrm{\FV@LabelEnd}
1949 \LWR@htmltagc{/div}
1950 \fi
1951     \fi
1952   }
1953   \fi
1954   \hss
1955 }
1956 }

```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

1957 \def\FV@ListProcessLine#1{%
1958   \hbox to \hsize{%
1959     %      \kern\leftmargin
1960     \hbox to \VerbatimHTMLWidth {%
1961       \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
1962 \FV@LeftListNumber%
1963 %      \FV@LeftListFrame
1964     \FancyVerbFormatLine{#1}%
1965     \hss%

```

```

1966 %          \FV@RightListFrame
1967          \FV@RightListNumber%
1968 }%
1969          \hss% required to avoid underfull hboxes
1970 }
1971 }

```

Env **BVerbatim**

```

1972 \AtBeginEnvironment{BVerbatim}
1973 {
1974 \LWR@atbeginverbatim{bverbatim}
1975
1976 }
1977
1978 \AfterEndEnvironment{BVerbatim}
1979 {
1980 \leavevmode\par\vspace{-\baselineskip}
1981 \LWR@afterendverbatim
1982 }

```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```

1983 } % \AfterEndPreamble

```

\UseVerbatim $\{\langle text \rangle\}$

No changes required.

```

1984 \end{warpHTML}

```

47 Theorems

\newtheorem $\{\langle text \rangle\}$ [$\langle counter \rangle$] -or- [$\langle oldname \rangle$] $\{\langle text \rangle\}$

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a div of class **theorem**.
- 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: 1985 \begin{warpHTML}

\@begintheorem {<name>} {<number>}

```
1986 \renewcommand{\@begintheorem}[2]{%
1987 \BlockClass{theoremcontents}
1988 \InlineClass{theoremlabel}{#1\ #2\ }
1989 }
```

\@opargbegintheorem {<name>} {<number>} {<oparg>}

```
1990 \renewcommand{\@opargbegintheorem}[3]{%
1991 \BlockClass{theoremcontents}
1992 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
1993 }
```

\@endtheorem

```
1994 \renewcommand*\@endtheorem{%
1995 \endBlockClass% theoremcontents
1996 }
```

```
1997 \end{warpHTML}
```

48 Lists

If using babel with French, use

```
\frenchbsetup{StandardLists=true}
```

 **French** to preserve the special HTML and enumitem list handling.

enumitem enumitem is pre-loaded during HTML output. Many of the spacing options are rendered irrelevant by pdftotext and HTML. Numbering, labels, and \newlist function correctly.

48.1 Itemize

for HTML output: 1998 \begin{warpHTML}

```
1999 \let\LWR@origitem\item
```

\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.

```

2000 \newcommand*{\LWR@itemizeitem}{%
2001 \LWR@stoppars%
2002 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem{}}%
2003 \LWR@htmltag{li}%
2004 \LWR@startpars%
2005 \LWR@origitem%
2006 }

```

To have a blank item, use `\mbox{}`. 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}
...

```

Env `itemize` [*(enumitem options)*]

```

2007 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
2008
2009 \newcommand*{\LWR@itemizestart}{%
2010 \LWR@stoppars%
2011 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
2012 \LWR@htmltag{ul style="list-style-type:none">{}}%
2013 \LWR@startpars%
2014 \let\item\LWR@itemizeitem%
2015 }
2016
2017 \AtEndEnvironment{itemize}{\LWR@itemizeend}
2018
2019 \newcommand*{\LWR@itemizeend}{%
2020 \LWR@stoppars%
2021 \LWR@closeprevious{\LWR@depthlistitem}%
2022 \LWR@closeoneprevious{}}%
2023 \LWR@startpars%
2024 }

```

48.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env **enumerate** [*enumitem options*]

```

2025 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
2026
2027 \newcommand*{\LWR@enumeratestart}{%
2028 \LWR@stoppars%
2029 \LWR@pushhonestclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
2030 \LWR@htmltag{ul style="list-style-type:none">{}}%
2031 \LWR@startpars%
2032 \let\item\LWR@itemizeitem%
2033 }
2034
2035
2036 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
2037
2038 \newcommand*{\LWR@enumerateend}{%
2039 \LWR@stoppars%
2040 \LWR@closeprevious{\LWR@depthlistitem}%
2041 \LWR@closeoneprevious{}}%
2042 \LWR@startpars%
2043 }
```

48.3 Description

\LWR@descitem [*label*] Handles an `\item` inside a description.

```

2044 \newcommand*{\LWR@descitem}[1][]%
2045 {%
2046 \LWR@stoppars%
2047 \LWR@setlatestname{#1}%
2048 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
2049 \LWR@origitem[]%
```

Be sure the label doesn't print to the left of the rest of the file:

```

2050 \LWR@origspace{1in}
2051 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
2052 \LWR@orignewline%
2053 \LWR@htmltag{dd}%
2054 \LWR@startpars%
2055 }
```

Env **description** [*enumitem options*]

```

2056 \AtBeginEnvironment{description}{\LWR@descriptionstart}
2057
```

```

2058 \newcommand*{\LWR@descriptionstart}{%
2059 \LWR@stoppars%
2060 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
2061 \LWR@htmltag{dl}%
2062 \LWR@startpars%
2063 \let\item\LWR@descitem%
2064 }
2065
2066 \AtEndEnvironment{description}{\LWR@descriptionend}
2067
2068 \newcommand*{\LWR@descriptionend}{%
2069 \LWR@stoppars%
2070 \LWR@closeprevious{\LWR@depthlistitem}%
2071 \LWR@closeoneprevious{}}%
2072 \LWR@startpars%
2073 }

```

`\newlist` $\{\langle name \rangle\} \{\langle type \rangle\} \{\langle maxdepth \rangle\}$

`\renewlist` $\{\langle name \rangle\} \{\langle type \rangle\} \{\langle maxdepth \rangle\}$

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.

```

2074 \let\LWR@orignewlist\newlist
2075
2076 \renewcommand*{\newlist}[3]{%
2077 \LWR@orignewlist{#1}{#2}{#3}%
2078 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
2079 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
2080 }
2081 \end{warpHTML}

```

49 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

Limitations:

- [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.
- \newcolumnntype is ignored; unknown column types are set to 1.
- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

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

- 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.

- 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.

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

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

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

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

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a “misplaced \noalign” error occurs, change
This & That \endhead
to
\warpprintonly{This & That \endhead}
and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevant to HTML output.

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

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

⚠ \multirow &
\multicolumn

⚠ \multirow

\multirow with rules

rule at last row

⚠ paragraphs

\cmidrule width, trim

longtable headings

⚠ \warpprintonly

⚠ S columns

49.1 Token lookahead

Used by `\LWR@futurenonpacelet` to look at the next token.

for HTML output: 2082 `\begin{warpHTML}`

`\LWR@mynexttoken`

2083 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonpacelet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonpacelet`

```
2084 \def\LWR@futurenonpacelet#1{\def\LWR@cs{#1}%
2085 \afterassignment\LWR@fns lone\let\nexttoken= }
2086 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
2087 \def\LWR@fns ltwo{%
2088 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfns lthree%
2089 \else\let\next=\nexttoken\fi\next}
2090 \def\@BTfns lthree{\afterassignment\LWR@fns lone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
2091 \newcommand*{\LWR@getmynexttoken}{%
2092 % nothing must follow this next line
2093 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumn tag
2094 }
```

49.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
2095 \newbool{LWR@startedrow}
2096 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
2097 \newbool{LWR@doinghline}
2098 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
2099 \newbool{LWR@doingtbrule}
2100 \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.

```
2101 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
2102 \newbool{LWR@skippingmrowcell}
```

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.

```
2103 \newbool{LWR@intabularmetadata}
2104 \boolfalse{LWR@intabularmetadata}
```

49.3 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>

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
2105 \newcommand*{\LWR@closetabledatacell}{%
2106 \global\booltrue{LWR@intabularmetadata}%
2107 \ifbool{LWR@exitingtabular}{}%
2108 {% not exiting tabular
2109 \ifbool{LWR@skippingmrowcell}{}%
2110 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
2111 \unskip%
2112 \LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}%
2113 \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
2114 \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
2115 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
2116 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
2117 \unskip\LWR@htmltag{/td}\LWR@orignewline%
2118 }% not skipping mrowcell
2119 }% not exiting tabular
2120 \global\boolfalse{LWR@skippingmrowcell}%
2121 }
```

`LWR@tabulardepth` tracks whether `&` is being used inside a `tabular`.

```
2122 \newcounter{LWR@tabulardepth}
2123 \setcounter{LWR@tabulardepth}{0}
2124
```

When not used inside a `tabular`, `&` performs its original function as recorded here (with catcode 4).

```
2125 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2126 \begingroup
```

`&` 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.

The `\catcode` allows the `&` character to be redefined.

```
2127 \catcode'\&=\active
2128
2129 \gdef&{%
2130 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2131 {%
```

If not skipping a multirow cell, close the current data cell.

```
2132 \unskip%
2133 \LWR@closetabledatacell%
```

Move to the next column.

```
2134 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.


```
2135 \LWR@getmynexttoken%
2136 }%
```

If not inside a tabular, performs the original action:

```
2137 {\LWR@origampmacro}%
2138 }
2139 \endgroup
```

Outside the group, & is left its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded.

\LWR@lwarpStart finally makes & active at the beginning of the HTML conversion.

49.4 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendoffline

Throws away options \\[dim] or *

\LWR@tabularendoffline

```
2140 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
2141 {%
2142 \LWR@closetabledatacell%
```

Finish the previous row:

```
2143 \LWR@htmltag{/tr}\LWR@orignewline
2144 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
2145 \global\boolfalse{\LWR@startedrow}
```

Additional setup:

```
2146 \global\boolfalse{\LWR@doinghline}%
2147 \global\boolfalse{\LWR@doingtbrule}%
2148 \LWR@clearmidrules%
```

Start at first column:

```
2149 \setcounter{\LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
2150 \LWR@getmynexttoken%
2151 }
```

49.5 Variables

```
2152 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2153 \newcommand*{\LWR@pposition}{}
2154 \newcommand*{\LWR@pleft}{}
2155 \newcommand*{\LWR@pright}{}

```

`\LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalcols`.

Will contain a string such as `llrrccpc`, exactly one letter per column, without `@`, `>`, `<`, or the vertical pipe.

```
2156 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
2157 \newcommand*{\LWR@strresult}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
2158 \newcommand*{\LWR@origcolspec}{}

```

`Ctr` `LWR@tablecolwidth` Holds the width of the table specification.

(This is not the total # columns.)

```
2159 \newcounter{LWR@tablecolwidth}

```

`Ctr` `LWR@tablecolspos` Where are currently looking into the table column specification.

```
2160 \newcounter{LWR@tablecolspos}

```

`Ctr` `LWR@tabletotalcols` Holds the final number of table columns.

```
2161 \newcounter{LWR@tabletotalcols}

```

`Ctr` `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
2162 \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 23.
- LWR@colbangspec** A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 23.
- LWR@colbeforespec** A data array of specifications for > columns.
- LWR@colafterspec** A data array of specifications for < columns.

49.6 Parsing @, >, <, ! columns

`\LWR@parseatcolumn` Handles @{text} columns.

```
2163 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2164 \LWR@traceinfo{at column}%
2165 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
2166 \LWR@traceinfo{about to read the next token:}%
2167 \expandarg%
2168 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2169 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
2170 \LWR@traceinfo{have now read the next token}%
2171 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
2172 {% left edge of the table:
2173 \LWR@traceinfo{at the left edge}%
2174 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
2175 }%
2176 {% not at the left edge:
2177 \LWR@traceinfo{not at the left edge}%
2178 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2179 \LWR@traceinfo{at \theLWR@tabletotalcols: %
2180 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}!}%
2181 \let\LWR@strresult\relax%
2182 \booltrue{LWR@validtablecol}%
2183 }%
2184 }
```

\LWR@parsebangcolumn

```
2185 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
2186 \LWR@traceinfo{bang column}%
```

```
2187 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
2188 \LWR@traceinfo{about to read the next token:}%
```

```
2189 \expandarg%
```

```
2190 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
```

```
2191 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
2192 \LWR@traceinfo{have now read the next token}%
```

```
2193 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
```

```
2194 {% left edge of the table:
```

```
2195 \LWR@traceinfo{at the left edge}%
```

```
2196 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@strresult}%
```

```
2197 }%
```

```
2198 {% not at the left edge:
```

```
2199 \LWR@traceinfo{not at the left edge}%
```

```
2200 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
```

```
2201 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%
```

```
2202 }%
```

```
2203 \let\LWR@strresult\relax%
```

```
2204 \booltrue{LWR@validtablecol}%
```

```
2205 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
2206 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
2207 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
2208 \expandarg%
```

```
2209 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
```

```
2210 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2211 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2212 \let\LWR@strresult\relax%
2213 \booltrue{LWR@validtablecol}%
2214 }

```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```

2215 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '`<`':

```

2216 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into `\LWR@strresult`:

```

2217 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
2218 \expandarg%
2219 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2220 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

2221 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2222 \let\LWR@strresult\relax%
2223 \booltrue{LWR@validtablecol}%
2224 }

```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```

2225 \newcommand*{\LWR@parseskipcolumn}{%
2226 \booltrue{LWR@validtablecol}%
2227 }

```

49.7 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.

```

2228 \newcommand*{\LWR@parsenormalcolumn}[1]{%
2229 \appto\LWR@tablecolspec{#1}%
2230 \addtocounter{LWR@tabletotalcols}{1}%
2231 \addtocounter{LWR@tabletotalcolsnext}{1}%
2232 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%

```

```

2233 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
2234 \LWR@setexparray{LWR@colbefore-spec}{\theLWR@tabletotalcolsnext}{\relax}%
2235 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
2236 \booltrue{LWR@validtablecol}%
2237 }

```

49.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```
2238 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
2239 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```
2240 \addtocounter{LWR@tablecolspos}{1}%
2241 }
```

49.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```
2242 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
2243 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
2244 \addtocounter{LWR@tablecolspos}{3}%
2245 }
```

49.10 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

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

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.

```
2246 \newbool{LWR@validtablecol}
```

`\LWR@parsetablecols` $\{\langle colspecs \rangle\}$

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

```
2247 \newcommand*{\LWR@parsetablecols}[1]{%
2248 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
2249 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
2250 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
2251 \setcounter{LWR@tabletotalcols}{0}%
2252 \setcounter{LWR@tabletotalcolsnext}{1}%
2253 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%
2254 \LWR@setexparray{LWR@colatspec}{1}{\relax}%
2255 \LWR@setexparray{LWR@colatspec}{2}{\relax}%
2256 \LWR@setexparray{LWR@colatspec}{3}{\relax}%
```

Table 7: Tabular HTML column conversions

l, r, c:	Converted to table cells without paragraph tags. Uses CSS <code>vertical-align:middle</code> 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 <code>vertical-align:bottom</code> 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 <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
m:	With paragraph tags, CSS <code>vertical-align:middle</code> .
b:	With paragraph tags, CSS <code>vertical-align:top</code> 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.

```

2257 \LWR@setexparray{LWR@colbangspec}{\leftedge}{\relax}%
2258 \LWR@setexparray{LWR@colbangspec}{1}{\relax}%
2259 \LWR@setexparray{LWR@colbangspec}{2}{\relax}%
2260 \LWR@setexparray{LWR@colbangspec}{3}{\relax}%
2261 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
2262 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
2263 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
2264 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
2265 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
2266 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%

```

Starting at the first column specification:

```

2267 \setcounter{LWR@tablecolspos}{1}%

```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```

2268 \LWR@traceinfo{about to StrLen}%
2269 \noexpandarg%
2270 \StrLen{#1}[\LWR@strresult]%
2271 \fullexpandarg%
2272 \LWR@traceinfo{finished StrLen}%
2273 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%

```

Scan through the column specifications:

```

2274 \whileof{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%

```

Place the next single-character column type into `\LWR@strresult`:

```

2275 \noexpandarg%
2276 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
2277 \fullexpandarg%

```

Not yet found a valid column type

```

2278 \boolfalse{LWR@validtablecol}%

```



Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.

```

2279 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{1}}{%
2280 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
2281 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
2282 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
2283 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
2284 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
2285 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%

```

```

2286 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
2287 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
2288 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
2289 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
2290 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%

```

From the dcolumn package:

```

2291 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

2292 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%

```

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumnntype{P}[1]{>{\centering\arraybackslash}p{#1}}

```

```

2293 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{%
2294 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{%
2295 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{%

```

If this column was an invalid column type, convert it to a p column:

```

2296 \ifbool{LWR@validtablecol}{}%
2297 \LWR@parsenormalcolumn{1}%
2298 }%
2299 \addtocounter{LWR@tablecolspos}{1}%
2300 }%
2301 }%

```

49.11 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.

```

2302 \newcommand*{\LWR@maybenewtablerow}
2303 {%
2304 \ifbool{LWR@startedrow}%
2305 {}% started the row
2306 {% not started the row

```

Remember that now have started the row:

```

2307 \global\booltrue{LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

2308 \global\booltrue{LWR@intabularmetadata}%
2309 \ifbool{LWR@doinghline}%
2310 {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
2311 {% not doing hline
2312 \ifbool{LWR@doingtbrule}%
2313 {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
2314 {\LWR@htmltag{tr}\LWR@orignewline}%
2315 }% end of not doing hline
2316 }% end of not started the row
2317 }

```

49.12 Data opening tag

`\LWR@tabledatasinglecolumntag` Print a table data opening tag with style for alignment

```

2318 \newcommand*{\LWR@tabledatasinglecolumntag}%
2319 {%
2320 \LWR@maybe newtable row%

```

If have found the end of tabular command, do not create the next data cell:

```

2321 \ifbool{LWR@exitingtabular}{}%
2322 {% not exiting tabular

```

Fetch the current column's alignment character into `\LWR@strresult`:

```

2323 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

2324 \LWR@htmltag{td class="td%

```

append this column's spec:

```

2325 \LWR@strresult%

```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```

2326 \ifthenelse{\equal{\LWR@gettexpparray{LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
2327 "{}}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

2328 \ifthenelse{%

```

```

2329 \equal{\LWR@strresult}{p}\OR%
2330 \equal{\LWR@strresult}{m}\OR%
2331 \equal{\LWR@strresult}{b}\OR%
2332 \equal{\LWR@strresult}{P}\OR%
2333 \equal{\LWR@strresult}{M}\OR%
2334 \equal{\LWR@strresult}{B}\OR%
2335 \equal{\LWR@strresult}{X}%
2336 }%
2337 {% allow pars
2338 \LWR@startpars%
2339 \global\booltrue{\LWR@tableparcell}%
2340 }% allow pars
2341 {}% no pars

```

Print the @ and ! contents before first column, and then the > contents:

```

2342 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
2343 {%
2344 \LWR@getexparray{\LWR@colatspec}{\leftedge}%
2345 \LWR@getexparray{\LWR@colbangspec}{\leftedge}%
2346 }% left edge
2347 {}% not left edge
2348 \LWR@getexparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}%
2349 \global\boolfalse{\LWR@intabularmetadata}%
2350 }% not exiting tabular
2351 }%

```

49.13 Midrules

LWR@midrules LWR@midrules is a data array (section 23) of columns containing Y if a midrule should be created for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules data array.

```

2352 \newcounter{\LWR@midrulecounter}

```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```

2353 \newcommand*{\LWR@clearmidrules}
2354 {%
2355 \setcounter{\LWR@midrulecounter}{1}%
2356 \whiledo{%
2357 \cnttest{\value{\LWR@midrulecounter}}{<=}{\value{\LWR@tablecolwidth}}%
2358 }%
2359 {%

```

```

2360 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
2361 \addtocounter{LWR@midrulecounter}{1}%
2362 }%
2363 }

```

`\LWR@subcmidrule` [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

2364 \newcommand*{\LWR@subcmidrule}[4]{%
2365 \setcounter{LWR@midrulecounter}{#3}%
2366 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
2367 {%
2368 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2369 \addtocounter{LWR@midrulecounter}{1}%
2370 }% end of the whiledo
2371 }

```

`\LWR@docmidrule` [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns.

```

2372 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
2373 {\LWR@subcmidrule{#1}{#2}{#3}}

```

49.14 Multicolumns

49.14.1 Parsing multicolumns

```

2374 \newcounter{LWR@tablemulticolwidth}
2375 \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`.

```

2376 \newcommand*{\LWR@printmccoltype}[1]{%
2377 \LWR@traceinfo{lw@printmccoltype -#1-%}

```

Get one token of the column spec:

```

2378 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Add to the HTML tag depending on which column type is found:

```

2379 \IfStrEq{\LWR@strresult}{l}{l}{}%
2380 \IfStrEq{\LWR@strresult}{c}{c}{}%
2381 \IfStrEq{\LWR@strresult}{r}{r}{}%
2382 \IfStrEq{\LWR@strresult}{p}{p}{}%
2383 \IfStrEq{\LWR@strresult}{m}{m}{}%
2384 \IfStrEq{\LWR@strresult}{b}{b}{}%
2385 \IfStrEq{\LWR@strresult}{P}{P}{}%
2386 \IfStrEq{\LWR@strresult}{M}{M}{}%
2387 \IfStrEq{\LWR@strresult}{B}{B}{}%
2388 \IfStrEq{\LWR@strresult}{S}{r}{}%
2389 \IfStrEq{\LWR@strresult}{X}{p}{}%
2390 \LWR@traceinfo{lw@printmccoltype done}%
2391 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

2392 \newcommand*{\LWR@multicolpartext}{%
2393 \LWR@startpars%
2394 \LWR@multicoltext%
2395 \LWR@stoppars%
2396 }

```

`\LWR@multicolother` $\{\langle colspec \rangle\}$ For @, >, <, print the next token without paragraph tags:

```

2397 \newcommand*{\LWR@multicolother}[1]{%
2398 \addtocounter{LWR@tablemulticolspos}{1}%
2399 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
2400 \LWR@strresult%

```

A valid column data type was found:

```

2401 \booltrue{LWR@validtablecol}%
2402 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

2403 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```

2404 \booltrue{LWR@validtablecol}%
2405 }

```

`\LWR@printmccoldata` $\{\langle colspec \rangle\}$ Print the data for any valid column type found.

```

2406 \newcommand*{\LWR@printmccoldata}[1]{%
2407 \LWR@traceinfo{lw@printmccoldata -#1}%

```

Not yet found a valid column type:

```
2408 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
2409 \StrChar{#1}{\theLWR@tablemulticolspos}[LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
2410 \IfStrEq{LWR@strresult}{l}{LWR@multicoltext}{}%
2411 \IfStrEq{LWR@strresult}{c}{LWR@multicoltext}{}%
2412 \IfStrEq{LWR@strresult}{r}{LWR@multicoltext}{}%
2413 \IfStrEq{LWR@strresult}{D}{}%
2414 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
2415 LWR@multicoltext%
2416 }{}%
2417 \IfStrEq{LWR@strresult}{p}{LWR@multicolparttext}{}%
2418 \IfStrEq{LWR@strresult}{m}{LWR@multicolparttext}{}%
2419 \IfStrEq{LWR@strresult}{b}{LWR@multicolparttext}{}%
2420 \IfStrEq{LWR@strresult}{P}{LWR@multicolparttext}{}%
2421 \IfStrEq{LWR@strresult}{M}{LWR@multicolparttext}{}%
2422 \IfStrEq{LWR@strresult}{B}{LWR@multicolparttext}{}%
2423 \IfStrEq{LWR@strresult}{S}{LWR@multicolparttext}{}%
2424 \IfStrEq{LWR@strresult}{X}{LWR@multicolparttext}{}%
2425 \IfStrEq{LWR@strresult}{|}{LWR@multicolskip}{}%
2426 \IfStrEq{LWR@strresult}{\detokenize@}{LWR@multicolother{#1}}{%
2427 \IfStrEq{LWR@strresult}{\detokenize!}{LWR@multicolother{#1}}{%
2428 \IfStrEq{LWR@strresult}{\detokenize>}{LWR@multicolother{#1}}{%
2429 \IfStrEq{LWR@strresult}{\detokenize<}{LWR@multicolother{#1}}{%
```

If an invalid column type:

```
2430 \ifbool{LWR@validtablecol}{LWR@multicoltext}%
```

Tracing:

```
2431 LWR@traceinfo{lwarp@printmccoldata done}%
2432 }
```

```
\parsemulticolumnalignment {<1: colspec>} {<2: printresults>}
```

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a p{spec} column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

2433 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2434 \setcounter{LWR@tablemulticolspos}{1}%
2435 \StrLen{#1}[\LWR@strresult]%
2436 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

2437 \whiledo{%
2438 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
2439 }%
2440 {%

```

Execute the assigned print function for each token in the column spec:

```

2441 #2{#1}%

```

Move to the next token in the column spec:

```

2442 \addtocounter{LWR@tablemulticolspos}{1}%
2443 }%
2444 }

```

49.14.2 High-level multicolumn interface

```
\LWR@domulticolumn {<1: numcols>} {<2: colspec>} {<3: text>}
```

```

2445 \newcommand{\LWR@multicoltext}{}
2446
2447 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2448 \LWR@traceinfo{lwrdomulticolumn -#1- -#2-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```

2449 \renewcommand{\LWR@multicoltext}{%
2450 #3%
2451 \booltrue{LWR@validtablecol}%
2452 }%

```

Row processing:

```

2453 \LWR@maybe newtable row%

```

Begin the opening table data tag:

```

2454 \LWR@htmltag{td colspan="#1"
2455 class="td%

```


Print the column type:

```
2456 \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”.

```
2457 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
2458 "%
2459 }% end of the opening table data tag
2460 \global\boolfalse{\LWR@intabularmetadata}%
2461 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2462 }
```

49.14.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
2463 \newbool{\LWR@starredlongtable}
2464 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
2465 \providecommand*{\LTcapttype}{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] {*caption*}

```
2466 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2467 {%
```

Remember the latest name for `\nameref`:

```
2468 \IfValueTF{#2}{% optional given?
2469 \ifthenelse{\equal{#2}{}}{% optional empty?
2470 {\LWR@setlatestname{#3}}{% empty
2471 {\LWR@setlatestname{#2}}{% given and non-empty
2472 }% optional given
2473 {\LWR@setlatestname{#3}}{% no optional
```

create a multicolumn across all the columns

```
2474 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
```

```

2475 % \IfBooleanTF{#1}% star?
2476 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
2477 % {\IfValueTF{#2}{\LWR@origcaption[2]{#3}}{\LWR@origcaption{#3}}}
2478 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

2479 {% yes star
2480 \LWR@htmlblocktag{figcaption}%
2481 #3%
2482 \LWR@htmlblocktag{/figcaption}%
2483 }%
2484 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

2485 \ifbool{\LWR@starredlongtable}%
2486 {%
2487 \ifthenelse{\equal{#2}{}}% TOC entry
2488 {}%
2489 {%
2490 \refstepcounter{\LTcapttype}%
2491 \protected@edef\@currentlabel{%
2492 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2493 }%
2494 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

2495 \LWR@htmlblocktag{figcaption}%
2496 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
2497 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

2498 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

2499 {}%

```

If the optional caption was given, but might only be []:

```

2500 {% TOC entry not empty
2501 \IfNoValueTF{#2}% No TOC entry?

```

The optional caption is []:

```

2502 {% No TOC entry
2503 \addcontentsline%
2504 {\csuse{ext@LTcapttype}}%
2505 {\LTcapttype}%
2506 {%
2507 \protect\numberline%
2508 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2509 {\ignorespaces #3\protect\relax}%
2510 }%
2511 }% end of No TOC entry

```

The optional caption has text enclosed:

```

2512 {% yes TOC entry
2513 \addcontentsline%
2514 {\csuse{ext@LTcapttype}}%
2515 {\LTcapttype}%
2516 {%
2517 \protect\numberline%
2518 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2519 {\ignorespaces #2\protect\relax}%
2520 }%
2521 }% end of yes TOC entry
2522 }% end of TOC entry not empty
2523 }% end of no star
2524 }% end of \LWR@domulticolumn
2525
2526 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
2527 \addtocounter{LWR@tablecolspos}{-1}
2528
2529 }

```

49.14.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` $\{ \langle numcols \rangle \} \{ \langle alignment \rangle \} \{ \langle text \rangle \}$

```

2530 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
2531 {%
2532 \LWR@domulticolumn{#1}{#2}{#3}%
2533 \addtocounter{LWR@tablecolspos}{#1}%
2534 \addtocounter{LWR@tablecolspos}{-1}%
2535 }

```

49.15 Multirow

Pkg multirow

```
\LWR@tabledatamultirowtag {<numrows>} [<bigstruts>] {<width>} [<fixup>] {<text>}
```

```
2536 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
```

```
2537 {%
```

```
2538 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
2539 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
2540 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
2541 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
```

```
2542 "{}}
```

While printing the text, redefine \\\ to generate a new line

```
2543 \begingroup \let\\\LWR@endoffline #5 \endgroup
```

```
2544 \LWR@stoppars%
```

```
2545 \global\boolfalse{\LWR@intabularmetadata}%
```

```
2546 }%
```

49.16 Utility macros inside a table

```
2547 \newcommand*{\LWR@donothing}{}%
```

```
2548 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}%
```

```
2549 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}%
```

49.17 Checking for a new table cell

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
2550 \newbool{\LWR@exitingtabular}
```

```
2551 \newcommand*{\LWR@tabledatacolumnntag}%
```

```
2552 {%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
2553 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
2554 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2555 {\booltrue\LWR@exitingtabular}}{ }%
```

`longtable` can have a caption in a cell

```
2556 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2557 {\let\mynext\LWR@donothing}}{ }%
```

Look for other things which would not start a table cell:

```
2558 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2559 {\let\mynext\LWR@donothing}}{ }%
2560 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2561 {\let\mynext\LWR@donothing}}{ }%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
2562 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2563 {\let\mynext\LWR@donothing}}{ }%
2564 %
2565 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2566 {\let\mynext\LWR@donothing}}{ }%
2567 %
2568 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2569 { }%
2570 \let\mynext\LWR@donothing}}{ }%
2571 %
2572 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2573 {\let\mynext\LWR@donothing}}{ }%
2574 %
2575 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2576 {\let\mynext\LWR@donothing}}{ }%
2577 %
2578 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2579 {\let\mynext\LWR@donothing}}{ }%
2580 %
2581 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2582 {\let\mynext\LWR@donothing}}{ }%
2583 %
2584 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}
```

```

2585 {\let\mynext\LWR@donothing}{}%
2586 %
2587 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}
2588 {\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:


```

2589 \mynext%
2590 }

2591 \end{warpHTML}

```

49.18 `\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:

```

2592 \begin{warpall}
2593 \newcommand*{\mrowcell}{}
2594 \end{warpall}

```

49.19 New `\tabular` definition

for HTML output:

```

2595 \begin{warpHTML}

```

Env `LWR@tabular` [*⟨verticalposition⟩*] {*⟨colspecs⟩*}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

2596 \newenvironment*{LWR@tabular}[2]{}
2597 {}
2598 \LWR@traceinfo{tabular started}%
2599 \begingroup%
2600 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

2601 \global\boolfalse{LWR@startedrow}%

```

Not yet doing an hline:

```
2602 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2603 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2604 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```
2605 \global\booltrue{LWR@intabularmetadata}%
2606 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2607 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
2608 \LWR@stoppars%
```

Track column # for setting text-align:

```
2609 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2610 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
2611 \let\\\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:

```
2612 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
2613 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
2614 \NewDocumentCommand{\cmidrule}{o d() m}%
2615 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
2616 \RenewDocumentCommand{\cline}{m}%
```

```

2617 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2618 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2619 \newcommand*{\bottomrule}{\LWR@dotbrule\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`.

```

2620 \let\multicolumn\LWR@tabledatamulticolumntag%
2621 \let\multirow\LWR@tabledatamultirowtag%
2622 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
2623 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

2624 \global\boolfalse{\LWR@tableparcell}%
2625 \global\boolfalse{\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```

2626 \LWR@getmynexttoken%
2627 }%

```

Ending the environment:

```

2628 {%
2629 \LWR@closetabledatacell%
2630 \LWR@htmlblocktag{/tr}%
2631 \LWR@htmlblocktag{/table}%
2632 \global\boolfalse{\LWR@intabularmetadata}%

2633 \addtocounter{\LWR@tabulardepth}{-1}%
2634 \endgroup%
2635 }

2636 \end{warpHTML}

```

49.20 Array

Pkg `array`

`array` is also automatically loaded by `siunitx`.

50 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: 2637 `\begin{warpHTML}`

50.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

2638 `\newcommand*{\@currentlabelname}{}`

`\LWR@stripperperiod` `{\text}` [`\langle`.`\rangle`]

Removes a trailing period.

2639 `\def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%`

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

2640 `\newcommand*{\LWR@setlatestname}[1]{%`

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `getttitlestring`.

2641 `\GetTitleStringExpand{#1}%`

2642 `\edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%`

2643 `\edef\@currentlabelname{%`

2644 `\expandafter\LWR@stripperperiod\@currentlabelname%`

2645 `\ltx@empty.\ltx@empty\@nil%`

2646 `}%`

2647 `}`

Table 8: Cross-referencing data structures

Original L^AT_EX:	(print and HTML)
<code>\refstepcounter:</code> Steps the counter and sets <code>\@currentlabel</code> . <code>\@currentlabel:</code> <code>\p@<ctr>\the<ctr></code> Updated by <code>\refstepcounter</code> . <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code> <code>\newlabel:</code> When the .aux file is read, sets <code>\r@<label></code> . <code>\r@<label>:</code> Set to: <code>{\@currentlabel}{\thepage}}</code> <code>\ref:</code> Returns the first part of <code>\r@<label></code> . <code>\pageref:</code> Returns the second part of <code>\r@<label></code> .	
Added by lwarp:	(HTML only)
<code>\label:</code> Adds HTML tags (section 50.3), plus <code>\slabel</code> data (section 50.2): <code>zLWR@name:</code> The section name for this label. <code>zLWR@htmlfilenumber:</code> The filename or name for this label. <code>zLWR@lateximagedepth:</code> The <code>lateximagedepth</code> for this label. <code>zLWR@lateximagenumber:</code> The <code>lateximagenumber</code> for this label. <code>\nameref:</code> Emulated from <code>hyperref</code> for <code>lwarp</code> . See section 50.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 50.4.	
Added by amsmath:	(print and HTML)
<code>\label:</code> Execution is delayed until the math environment is completed. <code>\ltx@label:</code> L ^A T _E X <code>\label</code> , (HTML:patched by <code>lwarp</code>), later patched by <code>cleveref</code> .	
Added by cleveref:	(print and HTML)
<code>\refstepcounter:</code> Added: sets <code>\cref@currentlabel</code> . <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 35.4 for use with footnotes. <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code> <code>\newlabel:</code> (Unchanged.) When the .aux file is read, sets <code>\r@<label>@cref</code> . <code>\r@<label>@cref:</code> Set to: <code>{\cref@currentlabel}{\thepage}}</code> Utility functions: See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> . Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
<code>\cref</code> , etc.: Modified for <code>lwarp</code> . See section 59. <code>\label</code> inside <code>math</code> : See section 54.4.1.	
Footnotes: See <code>\noteentry</code> in section 35.4.	

50.2 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:

```
2648 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2649 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2650 \zref@newprop{zLWR@htmlfilenumber}{%
2651 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
2652 }%
```

Additional properties for lateximages:

```
2653 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2654 \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:

```
2655 \zref@addprop{special}{zLWR@name}
2656 \zref@addprop{special}{zLWR@htmlfilenumber}
2657 \zref@addprop{special}{zLWR@lateximagedepth}
2658 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
2659 \newcommand*{\LWR@spreff}[2]{%
2660 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```
2661 \newcommand*{\LWR@nameref}[1]{%
2662 \LWR@spreff{#1}{zLWR@name}%
2663 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```

2664 \newcommand*{\LWR@htmlfileref}[1]{%
2665 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
2666 \LWR@sprep{#1}{z\LWR@htmlfilenumber}%
2667 }

```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the `lateximagedepth` for this label:

```

2668 \newcommand*{\LWR@lateximagedepthref}[1]{%
2669 \LWR@sprep{#1}{z\LWR@lateximagedepth}%
2670 }

```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the `lateximagenumber` for this label:

```

2671 \newcommand*{\LWR@lateximagenumberref}[1]{%
2672 \LWR@sprep{#1}{z\LWR@lateximagenumber}%
2673 }

```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```

2674 \newcommand*{\LWR@splabel}[1]{%
2675 \LWR@setlatestname{\@currentlabelname}%
2676 \zref@labelbylist{#1}{special}}

```

50.3 Labels

`\LWR@subsublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

```

2677 \newcommand*{\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```

2678 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
2679 {}%
2680 {% not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

2681 \ifbool{\LWR@doingstartpars}%
2682 {% pars allowed
2683 \ifbool{\LWR@doingapar}
2684 {% par started
2685 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2686 }% par started
2687 {% par not started

```

```

2688 \LWR@stoppars%
2689 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2690 \LWR@startpars%
2691 }% par not started
2692 }% pars allowed
2693 {% pars not allowed
2694 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2695 }% pars not allowed
2696 }% not lateximage
2697 }

```

`\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 L^AT_EX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```

2698 \NewDocumentCommand{\LWR@newlabel}{m o}{%
2699 \LWR@traceinfo{\LWR@newlabel: starting}%
2700 \LWR@traceinfo{\LWR@newlabel: !#1!}%
2701 % \@bsphack%

```

Create a traditional L^AT_EX label, as modified by `cleveref`:

```

2702 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

2703 \LWR@traceinfo{\LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2704 \LWR@traceinfo{\LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
2705 \LWR@traceinfo{\LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
2706 \LWR@splabel{#1}%
2707 \LWR@subsublabel{#1}%
2708 % \@esphack%
2709 \LWR@traceinfo{\LWR@newlabel: done}%
2710 }

```

50.4 References

`\LWR@startref` $\{\langle label \rangle\}$ (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, `#` character, and a label.

```

2711 \newcommand*{\LWR@startref}[1]
2712 {%
2713 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}
2714 \LWR@traceinfo{\LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

2715 \LWR@htmltag{a href="%
2716 \LWR@traceinfo{\LWR@startref B}%
2717 \LWR@htmlrefsectionfilename{#1}%
2718 \LWR@traceinfo{\LWR@startref C}%
2719 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

2720 \LWR@traceinfo{\LWR@startref D: !#1!}%
2721 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

2722 {%
2723 \LWR@traceinfo{\LWR@startref D0: ??}
2724 ??}%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

2725 {%
2726 \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
2727 \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
2728 {%
2729 \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
2730 lateximage\LWR@lateximagenumberref{#1}%
2731 }%
2732 {%
2733 \LWR@traceinfo{\LWR@startref D3}%
2734 #1%
2735 }%
2736 }%
2737 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

2738 "}}%
2739 \LWR@traceinfo{\LWR@startref F}%
2740 }

```

`\LWR@subnewref` $\{\langle label \rangle\}$ $\{\langle label \text{ or } sub@label \rangle\}$

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```
2741 \NewDocumentCommand{\LWR@subnewref}{m m}{%
2742 \LWR@traceinfo{\LWR@subnewref #1 #2}%
2743 \LWR@startref{#1}%
2744 \LWR@origref{#2}%
2745 \LWR@htmltag{/a}%
2746 }
```

`\ref` * $\{\langle label \rangle\}$ `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * $\{\langle label \rangle\}$ Create an internal document reference link, or without a link if starred per hyperref.

```
2747 \NewDocumentCommand{\LWR@newref}{s m}{%
2748 \LWR@traceinfo{\LWR@newref #2}%
2749 \IfBooleanTF{#1}%
2750 {\LWR@origref{#2}}%
2751 {\LWR@subnewref{#2}{#2}}%
2752 }
```

`\pagerefPageFor` Text for starred page references.

```
2753 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` * $\{\langle label \rangle\}$ Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
2754 \NewDocumentCommand{\LWR@newpageref}{s m}{%
2755 \IfBooleanTF{#1}%
2756 {(\pagerefPageFor\LWR@origref{#2})}%
2757 {(\cpageref{#2})}%
2758 }
```

`\nameref` $\{\langle label \rangle\}$

```
2759 \newcommand*{\nameref}[1]{%
2760 \LWR@traceinfo{\nameref A}%
2761 \LWR@startref{#1}%
2762 \LWR@traceinfo{\nameref B}%
2763 \LWR@nameref{#1}%
2764 \LWR@traceinfo{\nameref C}%
2765 \LWR@htmltag{/a}%
```

```
2766 \LWR@traceinfo{nameref D}%
2767 }
```

`\Nameref {⟨label⟩}` In print, adds the page number. In HTML, does not.

```
2768 \let\Nameref\nameref
```

50.5 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.

```
2769 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2770 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref {⟨URL⟩} {⟨text⟩}`

```
2771 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2772 \LWR@htmltag{a href="#1" target="_{blank}\LWR@orignewline}#2\LWR@htmltag{/a}%
2773 \LWR@ensuredoingapar%
2774 }
```

`\LWR@subhyperrefclass {⟨URL⟩} {⟨text⟩} {⟨htmlclass⟩}`

```
2775 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2776 \LWR@htmltag{a href="#1"
2777 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
2778 \LWR@ensuredoingapar%
2779 }
```

`\href [⟨options⟩] {⟨URL⟩} {⟨text⟩}`

Create a link with accompanying text:

```

2780 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
2781 \LWR@subhyperref{#2}{#3}%
2782 \endgroup%
2783 \LWR@ensuredoingapar%
2784 }
2785
2786 \newcommand{\href}{%
2787 \LWR@ensuredoingapar%
2788 \begingroup%
2789 \catcode'\_ =12
2790 \LWR@hrefb%
2791 }

```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```

2792 \newcommand*\LWR@nolinkurlb{[1]{#1\endgroup\LWR@ensuredoingapar}
2793
2794 \newcommand{\nolinkurl}{%
2795 \LWR@ensuredoingapar%
2796 \begingroup\catcode'\_ =12
2797 \LWR@nolinkurlb%
2798 }

```

`\url` $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link:

```

2799 \newcommand*\LWR@urlb{[1]{%
2800 \href{#1}{#1}%
2801 \endgroup%
2802 \LWR@ensuredoingapar%
2803 }
2804
2805 \newcommand{\url}{%
2806 \LWR@ensuredoingapar%
2807 \begingroup\catcode'\_ =12
2808 \LWR@urlb%
2809 }

```

`\LWR@subinlineimage` $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

2810 \newcommand*\LWR@subinlineimage{[5] [] {%
2811 \ifthenelse{\equal{#1}{}}%
2812 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}}%

```

```
2813 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}%  
2814 }  
  
2815 \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 L^AT_EX 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.

51 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”.

51.1 Float captions

for HTML output: 2816 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
2817 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2818 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
2819 \addtocounter{LWR@thisfloat}{1}%
2820 \booltrue{LWR@freezethisfloat}%
```

```
2821 \begingroup
```

Settings while inside the environment:

```
2822 \LWR@origraggedright
```

Open an HTML figure tag:

```
2823 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
2824 \renewcommand*{\@capttype}{#1}
2825 \caption@settype{#1}
2826 \LWR@startpars
2827 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
2828 \let\@float\LWR@floatbegin
2829 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
2830 \newcommand*{\LWR@floatend}{%
2831 \LWR@stoppars%
2832 \LWR@htmllementend{figure}%
2833 \endgroup%
2834 \boolfalse{LWR@freezethisfloat}%
2835 \LWR@startpars%
2836 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```
2837 \let\end@float\LWR@floatend
2838 \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.

```
2839 \newcounter{LWR@thisfloat}
```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```
2840 \newbool{LWR@freezethisfloat}
2841 \boolfalse{LWR@freezethisfloat}
```

`\LWR@maybeinthisfloat`

```
2842 \newcommand*{\LWR@maybeinthisfloat}{%
2843 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
2844 }
```

`\@capttype` Remembers which float type is in use.

```
2845 \newcommand*{\@capttype}{}
```

51.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
2846 \AtBeginDocument{\providecommand*\CaptionSeparator{::~}}
```

`\@makecaption` $\{\langle name \text{ and } num \rangle\} \{\langle text \rangle\}$

Prints the float type and number, the caption separator, and the caption text.

```
2847 \AtBeginDocument{\renewcommand{\@makecaption}[2]{\#1\CaptionSeparator\#2}}
```

51.1.2 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` IDs 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.

Ctr `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
2848 \newcounter{LWR@nextautofloat}
2849 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` $\{\langle autopage \rangle\} \{\langle autofloat \rangle\}$

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.

```
2850 \newcommand*{\LWRsetnextfloat}[2]{%
2851 \setcounter{LWR@nextautopage}{#1}%
2852 \setcounter{LWR@nextautofloat}{#2}%
2853 }
```

Ctr `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.

```
2854 \newcounter{LWR@latestautopage}% updated each new HTML file
2855 \setcounter{LWR@latestautopage}{1}
```

```
2856 \let\LWR@origcaption@begin\caption@begin
2857 \let\LWR@origcaption@end\caption@end
2858 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2859 \newcommand{\LWR@caption@begin}
2860 {
2861 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
2862 \begingroup%
```

The `caption` code was not allowing the closing `par` tag:

```
2863 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
2864 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
2865 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
2866 \LWR@htmlblocktag{figcaption}%
2867 \LWR@origcaption@begin%
2868 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
2869 \newcommand{\LWR@caption@end}
2870 {%
2871 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
2872 \vspace*{\baselineskip}%
```

Closing tag:

```
2873 \LWR@htmlblocktag{/figcaption}%
2874 \endgroup%
2875 % \leavevmode% avoid bad space factor (0) error
2876 \LWR@traceinfo{\LWR@caption@end: done}%
2877 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
2878 \AtBeginDocument{
2879 \let\caption@begin\LWR@caption@begin
2880 \let\caption@end\LWR@caption@end
2881 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
2882 \let\LWR@origcaptionlistentry\captionlistentry
2883
2884 \renewcommand*{\captionlistentry}{%
2885 \LWR@maybeinthisfloat%
2886 \LWR@ensuredoingapar%
2887 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2888 \LWR@origcaptionlistentry%
2889 }
```

```

2890
2891 \def\LWR@LTcaptionlistentry{%
2892 \LWR@ensuredoingapar%
2893 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2894 \bgroup
2895 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
2896 {\egroup\LWR@LT@captionlistentry}}%
2897 \def\LWR@LT@captionlistentry#1{%
2898 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autpage 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.

```

2899 \let\LWR@origaddcontentsline\addcontentsline
2900
2901 \renewcommand*{\addcontentsline}[3]{%
2902 \ifthenelse{\equal{#1}{toc}}{ }{%
2903 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
2904 \addtocontents{\@nameuse{ext@#2}}{%
2905 \protect\LWRsetnextfloat%
2906 {\arabic{LWR@latestautopage}}}%
2907 {\arabic{LWR@thisfloat}}}%
2908 }% addtocontents
2909 }% not toc
2910 \LWR@origaddcontentsline{#1}{#2}{#3}%
2911 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

2912 \AtBeginDocument{
2913 \let\LWR@origcaptionof\captionof
2914
2915 \renewcommand*{\captionof}{%
2916 \LWR@maybeinthisfloat%
2917 \LWR@stoppars
2918 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2919 \LWR@origcaptionof%
2920 }
2921 }

2922 \end{warpHTML}

```


52 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 L^AT_EX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 2923 `\begin{warpHTML}`

52.1 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 L^AT_EX, 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 sideTOC.

```
2924 \newcommand*{\LWR@myshorttoc}[1]{
2925 \LWR@ensuredoingapar
```

Only if the file exists:

```
2926 \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.



disabled

For pdf_lat_ex, 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.

```
2927 \begingroup
2928 % \ifxetexorluatex%
2929 % \else
2930 % \inputencoding{latin1}% currently disabled
2931 % \fi
2932 \makeatletter
```

Read in the TOC file:

```
2933 \@input{\jobname.#1}
2934 % \makeatother
2935 \endgroup
2936 }%
2937 {}%
2938 }
```

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
2939 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
2940 \@ifundefined{chapter}
2941 {\LWR@closeprevious{\LWR@depthsection}}
2942 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
2943 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
2944 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
2945 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
2946 \LWR@myshorttoc{#1}
```

Close the nav:

```
2947 \LWR@htmlelementclassend{nav}{#1}
2948 }
```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
2949 \let\LWR@orig@starttoc\@starttoc
2950
2951 \renewcommand{\@starttoc}[1]{
```

```

2952 \LWR@html elementclass{nav}{#1}
2953 \LWR@orig@starttoc{#1}
2954 \LWR@html elementclassend{nav}{#1}
2955 }

```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

2956 \let\LWR@origtableofcontents\tableofcontents
2957 \let\LWR@origlistoffigures\listoffigures
2958 \let\LWR@origlistoftables\listoftables
2959 \renewcommand*{\tableofcontents}{%

```

Copy the `.toc` file to `.sidetoc` for printing the `sidetoc`. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

2960 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
2961 \LWR@printpendingfootnotes
2962 \LWR@origtableofcontents%
2963 }
2964 \renewcommand*{\listoffigures}{
2965 \LWR@printpendingfootnotes
2966 \LWR@origlistoffigures
2967 }
2968
2969 \renewcommand*{\listoftables}{
2970 \LWR@printpendingfootnotes
2971 \LWR@origlistoftables
2972 }

```

52.2 High-level TOC commands

`\listof` $\{\langle type \rangle\} \{\langle title \rangle\}$

Emulate the `\listof` command from the `float` package (section 91). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```

2973 \NewDocumentCommand{\listof}{m +m}{%
2974 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
2975 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
2976 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
2977 \jobname.\csuse{ext@#1}\relax
2978 }

```

52.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

CSS related to sideTOC:

nav.sidetoc: The entire sideTOC.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```
2979 \end{warpHTML}
```

for HTML & PRINT: 2980 `\begin{warpall}`

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

```
2981 \newcounter{SideTOCDepth}
2982 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sidetoc.

```
2983 \newcommand{\sidetocname}{Contents}
```

```
2984 \end{warpall}
```

for HTML output: 2985 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

```
2986 \newcommand*{\LWR@sidetoc}{
2987 \LWR@stoppars
2988
```

The entire sideTOC is placed into a `nav` of class `sidetoc`.

```
2989 \LWR@htmlclass{nav}{sidetoc}
2990
2991 \setcounter{tocdepth}{\value{SideTOCDepth}}
2992
```

The title is placed into a div of class `sidetoctitle`, and may contain paragraphs.

```
2993 \begin{BlockClass}{sidetoctitle}
2994 \sidetocname
2995 \end{BlockClass}
```

The table of contents is placed into a div of class `sidetoccontents`.

```
2996 \begin{BlockClass}{sidetoccontents}
2997 \LinkHome
2998
2999 \LWR@myshorttoc{sidetoc}
3000 \end{BlockClass}
3001 \LWR@html@elementclassend{nav}{sidetoc}
3002 }
```

52.4 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:

```
3003 \renewcommand*{\numberline}[1]{%
3004 \LWR@sectionnumber{#1}%
3005 }
```

`\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

```
3006 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
3007 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
3008 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```

3009 \LWR@subhyperrefclass{%
3010 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
3011 \LWR@stoppars%
3012 }
3013 {}
3014 }

```

Ctrl **lofdepth** TOC depth for figures.

```

3015 \newcounter{lofdepth}
3016 \setcounter{lofdepth}{1}

```

Ctrl **lotdepth** TOC depth for tables.

```

3017 \newcounter{lotdepth}
3018 \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

```

3019 \newcommand{\hypertocfloat}[5]{%
3020 \LWR@startpars

```

If some float-creation package has not yet defined the float type's **lofdepth** counter, etc, define it here:

```

3021 \@ifundefined{c@#3depth}{%
3022 \newcounter{#3depth}%
3023 \setcounter{#3depth}{1}%
3024 }{}%

```

Respond to **lofdepth**, etc.:

```

3025 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
3026 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{%
3027 \LWR@startpars%

```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```

3028 \LWR@subhyperrefclass{%
3029 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
3030 \#autofloat-\arabic{LWR@nextautofloat}}%
3031 {#4}{toc#2}%
3032 \LWR@stoppars%
3033 }{}%
3034 }

```

Automatically called by \contentsline:

```

3035 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
3036 \DeclareDocumentCommand{\l@chapter}{m m}
3037 {\hypertoc{0}{chapter}{#1}{#2}}
3038 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
3039 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
3040 \renewcommand{\l@subsubsection}[2]
3041 {\hypertoc{3}{subsubsection}{#1}{#2}}
3042 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
3043 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
3044 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
3045 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

3046 \end{warpHTML}

```

53 Index

See:

[http://tex.stackexchange.com/questions/187038/
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

for HTML output: 3047 \begin{warpHTML}

\LWR@indexsection Controls whether the index will be in a section or a chapter, depending on the documentclass.

```

3048 \@ifundefined{chapter}
3049 {\newcommand*\LWR@indexsection{\section{\indexname}}}
3050 {\newcommand*\LWR@indexsection{\chapter{\indexname}}}

```

\printindex

```

3051 \let\LWR@origprintindex\printindex
3052
3053 \renewcommand*{\printindex}
3054 {
3055 \LWR@indexsection
3056 \LWR@startpars
3057 \LWR@origprintindex
3058 }

```

Env **theindex**

```

3059 \renewenvironment*{theindex}{%
3060 \let\item\LWR@indexitem%
3061 \let\subitem\LWR@indexsubitem%
3062 \let\subsubitem\LWR@indexsubsubitem%
3063 }{}

```

\LWR@indexitem

```

3064 \newcommand{\LWR@indexitem}{
3065
3066 \InlineClass{indexitem}{}
3067 }

```

\LWR@indexitem

```

3068 \newcommand{\LWR@indexsubitem}{
3069
3070 \InlineClass{indexsubitem}{}
3071 }

```

\LWR@indexitem

```

3072 \newcommand{\LWR@indexsubsubitem}{
3073
3074 \InlineClass{indexsubsubitem}{}
3075 }

```

\hyperindexref $\{\langle autosecnumber \rangle\}$

\hyperindexref{web address} is inserted into *.ind by the xindy style file
lwarp_html.xdy

```

3076 \newcommand*{\hyperindexref}[1]{
3077 \LWR@htmltag{a href="%

```


Create an HTML reference to the autosection:

```
3078 \LWR@htmlrefsectionfilename{autopage-#1}\#autosec-#1"{}%}
```

Add the section number and section name, then close the tag:

```
3079 \LWR@origref{autopage-#1}\enskip\LWR@nameref{autopage-#1}%
3080 \LWR@htmltag{/a}%
3081 }
```

```
3082 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```
3083 \begin{warpprint}
3084 \newcommand{\hyperindexref}[1]{#1}
3085 \end{warpprint}
```

54 Math

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 font⁷, 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 tag 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.

⁷See section 156 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.

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**

L^AT_EX macros

- MathJax does not automatically support custom L^AT_EX macros, but they may be set up by the user.

custom MathJax macros

As an example of using custom L^AT_EX 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: 3086 `\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.

```
3087 \let\LWR@origtextdollar\$
3088
3089 \renewcommand*{\$}{%
3090 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{
3091 {\LWR@origtextdollar}%
3092 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
3093 }
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
3094 \newcounter{LWR@externalfilecnt}
```

54.1 Inline and display math

```
3095 \let\LWR@origdollar=$
3096 \let\secondorigdollar=$% balance for editor syntax highlighting

3097 \let\LWR@origopenparen\(\
3098 \let\LWR@origcloseparen\)
```

$\$$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
 $\$$

```
3099 \begingroup
3100 \catcode'\$=\active%
3101 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar%
```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
3102 \gdef\LWR@doubledollar$#1$${
3103 \ifbool{mathjax}%
```

For MathJax, print the math between `\[` and `\]`:

```
3104 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```
3105 {% not mathjax
3106
3107 \begin{lateximage}%
3108 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
3109 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
3110 \end{lateximage}
3111
3112 }
3113 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
3114 \gdef\LWR@singledollar#1${%
3115 \ifbool{mathjax}%
```

For MathJax, print the math between `\(` and `\)`:

```
3116 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```
3117 {% not mathjax
3118 \begin{lateximage}%
3119 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
3120 \LWR@origdollar#1\LWR@origdollar%
3121 \end{lateximage}%
3122 }%
3123 }%
```

`\(` Redefine to the above dollar macros.
`\)`

```

3124 \gdef\(#1\){$#1$}
3125 \gdef\[#1\]{$$#1$$}
3126
3127 \endgroup

```

Remove the old `math` and `displaymath` environments:

```

3128 \let\math\relax
3129 \let\endmath\relax
3130 \let\displaymath\relax
3131 \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`.

```

3132 \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`.

```

3133 \NewEnviron{displaymath}{\expandafter[\BODY\]\@ignoretrue}

```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```

3134 \AtBeginDocument{\catcode'\$=\active}

```

54.2 MathJax support

Ctr `LWR@nextequation` Used to add one to compute the next equation number.

```

3135 \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 .

```

3136 \newcommand*{\LWR@syncmathjax}{%

```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```

3137 \ifcsdef{thechapter}{

```

```

3138 \BlockClassSingle{hidden}{
3139 \textbackslash(
3140 \textbackslash{}seteqsection \{\thechapter\}
3141 \textbackslash)
3142 }
3143 }
3144 {}% not using chapters
3145

```

MathJax doesn't allow setting the equation number to 1:

```

3146 \ifthenelse{\cnttest{\value{equation}}>0}
3147 {

```

Tell MathJax that the next set of equations begins with the current \LaTeX equation number, plus one.

```

3148 \setcounter{LWR@nextequation}{\value{equation}}
3149 \addtocounter{LWR@nextequation}{1}

```

Place the MathJax command inside “\(\” and “\)” characters, to be printed to HTML, not interpreted by \LaTeX .

```

3150 \BlockClassSingle{hidden}{
3151 \textbackslash(
3152 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
3153 \textbackslash)
3154 }
3155 {}% not eq > 1
3156 }

```

\LaTeX @restoremathlatexformatting While producing math, use regular \LaTeX formatting instead of HTML tags.

```

3157 \newcommand*{\LWR@restoremathlatexformatting}{%
3158 \let\hspace\LWR@origspace%
3159 \let\rule\LWR@origrule%
3160 \let\,\LWR@origcomma% disable HTML short unbreakable space
3161 \let\textit\LWR@origtextit%
3162 \let\textbf\LWR@origtextbf%
3163 \let\texttt\LWR@origtexttt%
3164 \let\textsc\LWR@origtextsc%
3165 \let\textsf\LWR@origtextsf%
3166 \let\textrm\LWR@origtextrm%
3167 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3168 \let\textsuperscript\LWR@origtextsuperscript%
3169 \let\textsubscript\LWR@origtextsubscript%
3170 \let~\LWR@origtilde%
3171 \let\enskip\LWR@origenskip%
3172 \let\quad\LWR@origquad%

```

```

3173 \let\qqquad\LWR@origqqquad%
3174 }

```

`\LWR@hidelatexequation` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the \LaTeX version of the equation inside an HTML comment.

```

3175 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%

```

Stop HTML paragraph handling and open an HTML comment:

```

3176 \LWR@stoppars
3177 \LWR@htmlopencomment
3178

```

Start the LaTeX math environment inside the HTML comment:

```

3179 \begingroup
3180 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their \LaTeX meanings.

```

3181 \LWR@restoremathlatexformatting

```

See `\LWR@htmlmathlabel` in section [54.4.1](#).

Print the contents of the equation:

```

3182 #2

```

End the \LaTeX math environment inside the HTML comment:

```

3183 \csuse{\LWR@origend#1}
3184 \endgroup
3185

```

Close the HTML comment and resume HTML paragraph handling:

```

3186 \LWR@htmlclosecomment
3187 \LWR@startpars
3188 }

```

`\LWR@addmathjax` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

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 .

```

3189 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3190

```

Enclose the MathJax environment inside printed “\(" and “\)” characters.

```
3191 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
3192 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
3193 \textbackslash{}end\{#1\}
3194
3195 }
```

54.3 Equation environment

Remember existing `equation` environment:

```
3196 \let\LWR@origequation\equation
3197 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
3198 \let\equation\relax
3199 \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 L^AT_EX 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.

```
3200 \NewEnviron{equation}{%
3201
3202 \ifbool{mathjax}
```

MathJax output:

```
3203 {
```


Print commands to synchronize MathJax's equation number and format to the current \LaTeX chapter/section and equation number:

```
3204 \LWR@syncmathjax
```

Print the \LaTeX math inside an HTML comment:

```
3205 \LWR@hidelatexequation{equation}{\BODY}
3206 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the \LaTeX equation environment commands, and the contents of the environment's `\BODY`.

```
3207 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3208 \begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}]%
3209 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
3210 \textbackslash{end\{equation\}}]% alt tag
```

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
3211 \LWR@origequation
3212 \BODY% contents collected by NewEnviron
3213 \LWR@origendequation
3214 \end{lateximage}%
3215 }
3216
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3217 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

54.4 AMS Math environments

54.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for `multline`-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multline` environment and not an `equation`.

```
3218 \newbool{LWR@amsmultline}
3219 \boolfalse{LWR@amsmultline}
```

```
\LWR@htmlmathlabel {(\label)}
```

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \LaTeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
3220 \newcommand*{\LWR@htmlmathlabel}[1]{%
3221 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3222 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
3223 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3224 \ifbool{LWR@amsmultline}{\hspace*{\totwidth0}}
```

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.)

```
3225 \LWR@htmlclosecomment%
3226 \LWR@origltx@label{#1}%
3227 \LWR@htmlopencomment%
3228 }% text
3229 }% mathjax
3230 {%
3231 \LWR@origltx@label{#1}%
3232 }%
3233 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
3234 \newcommand*{\LWR@beginhideamsmath}{
3235 \LWR@stoppars
3236
3237 \LWR@htmlopencomment
```

```

3238
3239 \begingroup
3240 \LWR@restoremathlatexformatting
3241 }

```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```

3242 \newcommand*{\LWR@endhideamsmath}{
3243 \endgroup
3244
3245 \LWR@htmlclosecomment
3246
3247 \LWR@startpars
3248 }

```

54.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing.

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 $\mathscr{A}\mathscr{M}\mathscr{S}$ 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 `multline`

```

3249 \BeforeBeginEnvironment{multline}{
3250 \ifbool{mathjax}
3251 {
3252 \LWR@syncmathjax
3253 \booltrue{LWR@amsmultline}
3254 \LWR@beginhideamsmath
3255 }
3256 {
3257 \lateximage[LWRAMSMATHBODY]
3258 }
3259 }
3260
3261 \AfterEndEnvironment{multline}{
3262
3263 \ifbool{mathjax}
3264 {

```

```
3265 \LWR@endhideamsmath
3266 \boolfalse{LWR@amsmultline}
3267 \LWR@addmathjax{multline}{\the\@envbody}
3268 }
3269 {\endlateximage}
3270
3271 }
```

Env **multline***

```
3272 \BeforeBeginEnvironment{multline*}{
3273 \ifbool{mathjax}
3274 {
3275 \LWR@syncmathjax
3276 \booltrue{LWR@amsmultline}
3277 \LWR@beginhideamsmath
3278 }
3279 {
3280 \lateximage[LWRAMSMATHBODY]
3281 }
3282 }
3283
3284 \AfterEndEnvironment{multline*}{
3285
3286 \ifbool{mathjax}
3287 {
3288 \LWR@endhideamsmath
3289 \boolfalse{LWR@amsmultline}
3290 \LWR@addmathjax{multline*}{\the\@envbody}
3291 }
3292 {\endlateximage}
3293
3294 }
3295
```

Env **gather**

```
3296 \BeforeBeginEnvironment{gather}{
3297 \ifbool{mathjax}
3298 {
3299 \LWR@syncmathjax
3300 \boolfalse{LWR@amsmultline}
3301 \LWR@beginhideamsmath
3302 }
3303 {
3304 \lateximage[LWRAMSMATHBODY]
3305 }
3306 }
3307
```

```

3308 \AfterEndEnvironment{gather}{
3309
3310 \ifbool{mathjax}
3311 {
3312 \LWR@endhideamsmath
3313 \LWR@addmathjax{gather}{\the\@envbody}
3314 }
3315 {\endlateximage}
3316
3317 }

```

Env **gather***

```

3318 \BeforeBeginEnvironment{gather*}{
3319 \ifbool{mathjax}
3320 {
3321 \LWR@syncmathjax
3322 \boolfalse{LWR@amsmultline}
3323 \LWR@beginhideamsmath
3324 }
3325 {
3326 \lateximage[LWRAMSMATHBODY]
3327 }
3328 }
3329
3330 \AfterEndEnvironment{gather*}{
3331
3332 \ifbool{mathjax}
3333 {
3334 \LWR@endhideamsmath
3335 \LWR@addmathjax{gather*}{\the\@envbody}
3336 }
3337 {\endlateximage}
3338
3339 }

```

Env **align**

```

3340 \BeforeBeginEnvironment{align}{
3341 \ifbool{mathjax}
3342 {
3343 \LWR@syncmathjax
3344 \boolfalse{LWR@amsmultline}
3345 \LWR@beginhideamsmath
3346 }
3347 {
3348 \lateximage[LWRAMSMATHBODY]
3349 }
3350 }

```

```
3351
3352 \AfterEndEnvironment{align}{
3353
3354 \ifbool{mathjax}
3355 {
3356 \LWR@endhideamsmath
3357 \LWR@addmathjax{align}{\the\@envbody}
3358 }
3359 {\endlateximage}
3360
3361 }
```

Env **align***

```
3362 \BeforeBeginEnvironment{align*}{
3363 \ifbool{mathjax}
3364 {
3365 \LWR@syncmathjax
3366 \boolfalse{LWR@amsmultiline}
3367 \LWR@beginhideamsmath
3368 }
3369 {
3370 \lateximage[LWRAMSMATHBODY]
3371 }
3372 }
3373
3374 \AfterEndEnvironment{align*}{
3375
3376 \ifbool{mathjax}
3377 {
3378 \LWR@endhideamsmath
3379 \LWR@addmathjax{align*}{\the\@envbody}
3380 }
3381 {\endlateximage}
3382
3383 }
```

Env **flalign**

```
3384 \BeforeBeginEnvironment{flalign}{
3385 \ifbool{mathjax}
3386 {
3387 \LWR@syncmathjax
3388 \boolfalse{LWR@amsmultiline}
3389 \LWR@beginhideamsmath
3390 }
3391 {
3392 \lateximage[LWRAMSMATHBODY]
3393 }
```

```

3394 }
3395
3396 \AfterEndEnvironment{flalign}{
3397
3398 \ifbool{mathjax}
3399 {
3400 \LWR@endhideamsmath
3401 \LWR@addmathjax{flalign}{\the\@envbody}
3402 }
3403 {\endlateximage}
3404
3405 }

```

Env **flalign***

```

3406 \BeforeBeginEnvironment{flalign*}{
3407 \ifbool{mathjax}
3408 {
3409 \LWR@syncmathjax
3410 \boolfalse{LWR@amsmultline}
3411 \LWR@beginhideamsmath
3412 }
3413 {
3414 \lateximage[LWR@MSMATHBODY]
3415 }
3416 }
3417
3418 \AfterEndEnvironment{flalign*}{
3419
3420 \ifbool{mathjax}
3421 {
3422 \LWR@endhideamsmath
3423 \LWR@addmathjax{flalign*}{\the\@envbody}
3424 }
3425 {\endlateximage}
3426
3427 }

3428 \end{warpHTML}

```

55 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 \LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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

for HTML output: 3429 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
3430 \newcounter{LWR@lateximagenumber}
3431 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
3432 \newcounter{LWR@lateximagedepth}
3433 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
3434 \ifcsdef{\LWR@file}{-}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3435 \edef\LWR@hashmark{\string#} % for use in \write
3436 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@Lipage` Used to reference the PDF page number of a `lateximage` to be written into `lateximages.txt`.

```
3437 \newcounter{LWR@Lipage}
```

```
3438 \end{warpHTML}
```

for HTML & PRINT: 3439 `\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.

```
3440 \newcommand*{\LateximageFontSizeName}{large}

3441 \end{warpall}
```

for HTML output: 3442 \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...

```
3443 \newcommand{\LWR@HTMLsanitize}[1]{%
3444 \protect\StrSubstitute{\detokenize{#1}}{%
3445 {\detokenize{&}}{%
3446 {\detokenize{&}}[\LWR@strresult]}%
3447 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3448 {\detokenize{<}}{%
3449 {\detokenize{<}}[\LWR@strresult]}%
3450 [\LWR@strresult]}%
3451 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3452 {\detokenize{>}}{%
3453 {\detokenize{>}}[\LWR@strresult]}%
3454 [\LWR@strresult]}%
3455 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3456 {\detokenize{##}}{%
3457 {\detokenize{#}}[\LWR@strresult]}%
3458 [\LWR@strresult]}%
3459 \LWR@strresult%
3460 }
```

\LWR@HTMLsanitizeexpand {*\langle text \rangle*}

This version expands the argument before sanitizing it.

```
3461 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3462 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
3463 {\detokenize{&}}{%
3464 {\detokenize{&}}[\LWR@strresult]}%
3465 [\LWR@strresult]}%
3466 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3467 {\detokenize{<}}{%
3468 {\detokenize{<}}[\LWR@strresult]}%
3469 [\LWR@strresult]}%
```

```

3470 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3471 {\detokenize{>}}}%
3472 {\detokenize{&gt;}}}%
3473 [\LWR@strresult]%
3474 \LWR@strresult%
3475 }

```

Env `lateximage` [*alttag*]

```

3476 \NewDocumentEnvironment{lateximage}{0{image}}{%
3477 \LWR@traceinfo{lateximage: starting on page \arabic{page}}}%
3478 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

3479 {%
3480 \addtocounter{LWR@lateximagedepth}{1}%
3481 }%

```

Otherwise, this is the outer-most `lateximage`:

```

3482 {% start of outer-most lateximage

```

Starting a new `lateximage`:

```

3483 \addtocounter{LWR@lateximagenumber}{1}%
3484 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%

```

While inside a `lateximage`, do not use `mathjax`:

```

3485 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

3486 \LWR@ensuredoingapar%

```

Next file:

```

3487 \addtocounter{LWR@externalfilecnt}{1}%
3488 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%

```

Figure out what the next page number will be:

```

3489 \setcounter{pageref}{LWR@Lipage}{LWR@lateximage\theLWR@lateximagenumber}%
3490 \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:

```
3491 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
3492 class="lateximagesource"{}} \LWR@orignewline
```

Write instructions to the lateximages.txt file:

```
3493 \immediate\write\LWR@file{|\theLWR@Lipage|\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.

```
3494 \LWR@htmlopencomment%
3495 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
3496 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
3497 \LWR@origminipage{6in}%
3498 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
3499 \let\hspace\LWR@orighspace%
3500 \let\rule\LWR@origrule%
3501 \let\,\LWR@origcomma% disable HTML short unbreakable space
3502 \let\textit\LWR@origtextit%
3503 \let\textbf\LWR@origtextbf%
3504 \let\texttt\LWR@origtexttt%
3505 \let\textsc\LWR@origtextsc%
3506 \let\textsf\LWR@origtextsf%
3507 \let\textrm\LWR@origtextrm%
3508 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3509 \let\textsuperscript\LWR@origtextsuperscript%
3510 \let\textsubscript\LWR@origtextsubscript%
3511 \let~\LWR@origtilde%
3512 \let\enskip\LWR@origenskip%
3513 \let\quad\LWR@origquad%
3514 \let\qquad\LWR@origqquad%
3515 \let\tabular\LWR@origtabular%
3516 \let\endtabular\LWR@origendtabular%
3517 \let\newline\LWR@orignewline%
3518 \LWR@origlabel{LWR@lateximage\arabic{LWR@lateximagenumber}}% *8*
3519 }% end of outer-most lateximage
3520 }% end of \begin{lateximage}
3521 {% start of \end{lateximage}
```

```
3522 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3523 {%
3524 \addtocounter{LWR@lateximagedepth}{-1}%
3525 }%
```

if this is the outer-most lateximage:

```
3526 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
3527 \LWR@origendminipage%
3528 \LWR@orignewpage%
3529 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
3530 \LWR@htmlclosecomment{}\LWR@orignewline%
3531 \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.

```
3532 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
3533 {%
3534 \LWR@subinlineimage[%
3535 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
3536 ]%
3537 {lateximage}%
3538 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
3539 {svg}%
3540 {}%
3541 }%
3542 {%
3543 \LWR@subinlineimage[#1]{lateximage}%
3544 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
3545 }%
3546 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
3547 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
3548 \LWR@htmltag{/span}%
3549 \LWR@htmlcomment{End of lateximage}%
3550 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one `lateximage` level:

```
3551 \addtocounter{LWR@lateximagedepth}{-1}%
3552 }% end of outer-most lateximage
3553 \LWR@traceinfo{lateximage: done}
3554 }%
3555 \end{warpHTML}
```

```
for PRINT output: 3556 \begin{warpprint}
3557 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
3558 \end{warpprint}
```

56 center, flushleft, flushright

```
for HTML output: 3559 \begin{warpHTML}
```

Env **center** Replace `center` functionality with CSS tags:

```
3560 \renewenvironment*{center}
3561 {\BlockClass{center}}
3562 {\endBlockClass}
```

Env **flushright**

```
3563 \renewenvironment*{flushright}
3564 {\BlockClass{flushright}}
3565 {\endBlockClass}
```


Env **flushleft**

```
3566 \renewenvironment*{flushleft}
3567 {\BlockClass{flushleft}}
3568 {\endBlockClass}
```

```
3569 \end{warpHTML}
```

57 Siunitx

Pkg **siunitx**

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

for HTML output: 3570 `\begin{warpHTML}`

Options for siunitx:

```
3571 \PassOptionsToPackage{
3572 detect-mode=true,
3573 per-mode=symbol,% fraction is not seen by pdftotext
3574 text-celsius = {\HTMLentity{deg}C},
3575 text-degree = {\HTMLentity{deg}},
3576 }{siunitx}

3577 \end{warpHTML}
```

58 Graphics

Pkg **graphics**


Pkg **graphicx**

\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 CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L^AT_EX, so expect some ugly results for scaling and rotating.

for HTML output: 3578 `\begin{warpHTML}`

58.1 `\graphicspath`

`\graphicspath` `{\langle path \rangle}`

3579 `\newcommand*\thisgraphicspath{\{}}`

3580 `\renewcommand*\thisgraphicspath[1]{\renewcommand*\thisgraphicspath{\#1}}`


`\DeclareGraphicsExtensions` `{\langle list \rangle}`

`\DeclareGraphicsRule` `{\langle \rangle} {\langle \rangle} {\langle \rangle} {\langle \rangle}`

3581 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

3582 `\renewcommand*\DeclareGraphicsRule[4]{}`

58.2 Length conversions and graphics options

 **whitespace** A scaled image in L^AT_EX 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.

3583 `\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.

3584 `\newlength{\LWR@igwidth}`

3585 `\newlength{\LWR@igheight}`

3586 `\newcommand*\LWR@igwidthstyle{}`

3587 `\newcommand*\LWR@igheightstyle{}`

3588 `\newcommand*\LWR@igorigin{}`

```

3589 \newcommand*{\LWR@igangle}{%
3590 \newcommand*{\LWR@igxscale}{1}
3591 \newcommand*{\LWR@igyscale}{1}
3592 \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:

```

3593 \define@key{igraph}{width}{%
3594 \setlength{\LWR@igwidth}{#1}%
3595 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3596 {%

```

Default to use the converted fixed length given:

```

3597 \uselengthunit{PT}%
3598 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

3599 \IfEndWith{#1}{ex}%
3600 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
3601 {}% not ex
3602 \IfEndWith{#1}{em}%
3603 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
3604 {}% not em
3605 \IfEndWith{#1}{\}%
3606 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
3607 {}% not percent
3608 \IfEndWith{#1}{px}%
3609 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
3610 {}% not px
3611 }{}% end of length > 0pt
3612 }

```

If an optional height was given, set an HTML style:

```

3613 \define@key{igraph}{height}{%
3614 \setlength{\LWR@igheight}{#1}%
3615 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
3616 {%

```

Default to use the converted fixed length given:

```

3617 \uselengthunit{PT}%
3618 \renewcommand*{\LWR@igheightstyle}{%
3619 height:\rndprintlength{\LWR@igheight} %

```


3620 }%

If ex or em dimensions were given, use those instead:

```

3621 \IfEndWith{#1}{ex}%
3622 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
3623 {}% not ex
3624 \IfEndWith{#1}{em}%
3625 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
3626 {}% not em
3627 \IfEndWith{#1}{\}%
3628 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
3629 {}% not percent
3630 \IfEndWith{#1}{px}%
3631 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
3632 {}% not px
3633 }{}% end of length > Opt
3634 }

```

Handle origin key:

```

3635 \define@key{igraph}{origin}{%
3636 \renewcommand*{\LWR@igorigin}{#1}%
3637 }

```

Handle angle key:

```

3638 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

3639 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
3640

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

3641 \define@key{igraph}{scale}{%
3642 \renewcommand*{\LWR@igxscale}{#1}%
3643 \renewcommand*{\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

3644 \define@key{igraph}{bb}{}
3645 \define@key{igraph}{bblx}{}
3646 \define@key{igraph}{bbly}{}
3647 \define@key{igraph}{bburx}{}
3648 \define@key{igraph}{bbury}{}
3649 \define@key{igraph}{natwidth}{}

```

```

3650 \define@key{igraph}{natheight}{}
3651 \define@key{igraph}{hiresbb}{}
3652 \define@key{igraph}{viewport}{}
3653 \define@key{igraph}{trim}{}
3654 \define@key{igraph}{totalheight}{}
3655 \define@key{igraph}{keepaspectratio}{}
3656 \define@key{igraph}{clip}{}
3657 \define@key{igraph}{draft}{}
3658 \define@key{igraph}{type}{}
3659 \define@key{igraph}{ext}{}
3660 \define@key{igraph}{read}{}
3661 \define@key{igraph}{command}{}

```

`\LWR@rotstyle` $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

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.

```

3662 \newcommand*{\LWR@rotstyle}[2]{%
3663   #1transform:rotate(-#2deg);
3664 }

```

`\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.

```

3665 \newcommand*{\LWR@scalestyle}[3]{%
3666   #1transform:scale(#2,#3);
3667 }

```

58.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

3668 \newbool{LWR@infloatrow}
3669 \boolfalse{LWR@infloatrow}

3670 \newcommand*{\LWR@imageextension}{}
3671 \newcommand*{\LWR@expgraphicsfilename}{}

```

`\LWR@includegraphicsb` [*options*] {*filename*}

```
3672
3673 \newcommand*\LWR@includegraphicsb[2] []
3674 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
3675 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
3676 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3677 \ifbool{LWR@infloatrow}%
3678 {}
3679 {% not in a minipage or a floatrow:
3680 \setlength{\linewidth}{6in}%
3681 \setlength{\textwidth}{6in}%
3682 \setlength{\textheight}{9in}%
3683 }%
3684 }{}%
```

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.

```
3685 \edef\LWR@expgraphicsfilename{#2}
3686 \renewcommand*\LWR@imageextension{}%
3687 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
3688 {\renewcommand*\LWR@imageextension}{.jpg}}{}%
3689 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
3690 {\renewcommand*\LWR@imageextension}{.JPG}}{}%
3691 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
3692 {\renewcommand*\LWR@imageextension}{.png}}{}%
3693 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
3694 {\renewcommand*\LWR@imageextension}{.PNG}}{}%
3695 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
3696 {\renewcommand*\LWR@imageextension}{.gif}}{}%
3697 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
3698 {\renewcommand*\LWR@imageextension}{.GIF}}{}%
3699 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
3700 {\renewcommand*\LWR@imageextension}{.svg}}{}%
3701 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
3702 {\renewcommand*\LWR@imageextension}{.SVG}}{}%
3703 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
3704 {\renewcommand*\LWR@imageextension}{.svgz}}{}%
```

```

3705 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
3706 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in \LWR@strresult:

Must also \detokenize .pdf and .svg comparison strings.

```

3707 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
3708 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
3709 %
3710 \StrSubstitute{\LWR@strresult}%
3711 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

3712 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

3713 \setlength{\LWR@igwidth}{0pt}%
3714 \setlength{\LWR@igheight}{0pt}%
3715 \renewcommand*{\LWR@igwidthstyle}{}%
3716 \renewcommand*{\LWR@igheightstyle}{}%
3717 \renewcommand*{\LWR@igorigin}{}%
3718 \renewcommand*{\LWR@igangle}{}%
3719 \renewcommand*{\LWR@igxscale}{1}%
3720 \renewcommand*{\LWR@igyscale}{1}%
3721 \renewcommand*{\LWR@igclass}{inlineimage}%

```

Options are in #1

```

3722 \setkeys{igraph}{#1}%

```

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.

```

3723 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
3724 {% start of href
3725 \LWR@htmltag{% start of image tags
3726 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3727 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

3728 \ifthenelse{

```

```

3729 \NOT\equal{\LWR@igwidthstyle}{ } \OR
3730 \NOT\equal{\LWR@igheightstyle}{ } \OR
3731 \NOT\equal{\LWR@igorigin}{ } \OR
3732 \NOT\equal{\LWR@igangle}{ } \OR
3733 \NOT\equal{\LWR@igxscale}{1} \OR
3734 \NOT\equal{\LWR@igyscale}{1}
3735 }%
3736 {\LWR@origtilde{ } style="%
3737 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
3738 {\LWR@igwidthstyle; }}%
3739 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
3740 {\LWR@igheightstyle; }}%
3741 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
3742 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@originewline}{ }%
3743 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
3744 {%
3745 \LWR@rotstyle{-ms-}{\LWR@igangle}%
3746 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
3747 \LWR@rotstyle{ }{\LWR@igangle%
3748 } }%
3749 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
3750 \NOT\equal{\LWR@igyscale}{1}}%
3751 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
3752 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
3753 \LWR@scalestyle{ }{\LWR@igxscale}{\LWR@igyscale}}}%
3754 " \LWR@originewline}{ }%

```

Set the class:

```

3755 \LWR@origtilde{ } class="\LWR@igclass" \LWR@originewline%
3756 }% end of image tags
3757 }% end of href
3758 \endgroup

```

Return to small-sized output:

```

3759 \LWR@origscriptsize
3760 }

```

`\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.

```

3761 \renewcommand*\includegraphics{
3762 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

3763 \LWR@ensuredoingapar%
3764 \begingroup%
3765 \LWR@includegraphicsb%
3766 }

3767 \end{warpHTML}

```

for PRINT output: For print output, accept and then discard the new `class` key:

```

3768 \begin{warpprint}
3769 \define@key{Gin}{class}{}
3770 \end{warpprint}

```

58.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 3771 \begin{warpHTML}

`\LWR@rotboxorigin` Holds the origin key letters.

```
3772 \newcommand*{\LWR@rotboxorigin}{}

```

`\LWR@originname` $\{ \langle letter \rangle \}$

Given one L^AT_EX origin key value, translate into an HTML origin word:

```

3773 \newcommand*{\LWR@originname}[1]{%
3774 \ifthenelse{\equal{#1}{t}}{top}{}%
3775 \ifthenelse{\equal{#1}{b}}{bottom}{}%
3776 \ifthenelse{\equal{#1}{c}}{center}{}%
3777 \ifthenelse{\equal{#1}{l}}{left}{}%
3778 \ifthenelse{\equal{#1}{r}}{right}{}%
3779 }

```

`\LWR@originnames` $\{ \langle letters \rangle \}$

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```

3780 \newcommand*{\LWR@originnames}[1]{%
3781 \StrChar{#1}{1}[\LWR@strresult]%
3782 \LWR@originname{\LWR@strresult}%
3783 \StrChar{#1}{2}[\LWR@strresult]%
3784 \LWR@originname{\LWR@strresult}%
3785 }

```

Handle the origin key for `\rotatebox`:

```
3786 \define@key{krotbox}{origin}{%
3787 \renewcommand*{\LWR@rotboxorigin}{#1}%
3788 }
```

These keys are ignored:

```
3789 \define@key{krotbox}{x}{}
3790 \define@key{krotbox}{y}{}
3791 \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.

```
3792 \NewDocumentCommand{\LWR@rotatebox}{0} m +m {%

```

Reset the origin to “none-given”:

```
3793 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
3794 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
3795 \LWR@htmltagc{span style="display: inline-block;

```

If an origin was given, translate and print the origin information:

```
3796 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
3797 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
3798 \LWR@rotstyle{-ms-}{#2} %
3799 \LWR@rotstyle{-webkit-}{#2} %
3800 \LWR@rotstyle{}{#2} %
3801 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
3802 \begin{LWR@nestspan}%
3803 #3%

```

Close the span:

```
3804 \LWR@htmltagc{/span}%
3805 \end{LWR@nestspan}%
3806 }
```

`\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.

```
3807 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
3808 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
3809 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3810 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3811 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3812 "}}%
```

Print the text to be scaled:

```
3813 \begin{LWR@nestspan}%
3814 #3%
```

Close the span:

```
3815 \LWR@htmltagc{/span}%
3816 \end{LWR@nestspan}%
3817 }
```

`\reflectbox` $\{\langle text\rangle\}$

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
3818 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
3819 \end{warpHTML}
```

58.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.

for HTML output: 3820 \begin{warpHTML}

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
3821 \renewcommand{\resizebox}[3]{#3}
```

```
3822 \end{warpHTML}
```

59 Cleverref

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 194 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: 3823 \begin{warpHTML}

`\AtEndPreamble` forces `cleveref` to be loaded last:

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
3824 \AfterEndPreamble{
```

```
\@@setcref {<kindofref>} {<label>}
```

```
3825 \renewcommand*\@@setcref[2]{#1{\ref{#2}}{}{}}
```

```
\@@setcrefrange {<text>} {<label>} {<label>}
```

```
3826 \renewcommand{\@@setcrefrange}[3]{%
```

```
3827 #1{\ref{#2}}{\ref{#3}}{}{}{}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
3828 \newcommand*{\cpagerefFor}{for}
```

`\@@setcpageref` $\langle typeofref \rangle$ $\langle label \rangle$, where *typeofref* is “page” or “pages”

```
3829 \renewcommand*{\@@setcpageref}[2]{%
3830 #1{\cpagerefFor\ \cref{#2}}{}}%
3831 }
```

```
3832 \renewcommand{\@@setcpagerefrange}[3]{%
3833 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}}
3834 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
3835 \let\LWR@origlabel\label
3836 \let\label\LWR@newlabel
3837 \let\LWR@origref\ref
3838 \let\ref\LWR@newref% \end{ syntax highlighting
3839 \let\LWR@origpageref\pageref
3840 \let\pageref\LWR@newpageref
3841
3842
3843
3844 \end{warpHTML}
```

60 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 3845 `\begin{warpHTML}`

Env `picture`


```
3846 \BeforeBeginEnvironment{picture}{%
3847 \lateximage%
3848 \let\makebox\LWR@origmakebox%
3849 }
3850
3851 \AfterEndEnvironment{picture}{\endlateximage}

3852 \end{warpHTML}
```

61 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

 **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`.


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 are ignored. 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` tag, 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 L^AT_EX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 3853 `\begin{warpHTML}`

61.1 Counters and lengths

Ctrl LWR@minipagedepth Used to only reset the line width at the outermost minipage.

```
3854 \newcounter{LWR@minipagedepth}
3855 \setcounter{LWR@minipagedepth}{0}
```

Len \WR@minipagewidth Used to convert the width into printable units.

```
3856 \newlength{\WR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
3857 \newlength{\LWR@minipageheight}
```

Remember the original definitions:

```
3858 \let\LWR@origminipage\minipage
```

```
3859 \let\LWR@origendminipage\endminipage
```

61.2 Footnote handling

Also see section 35 for other forms of footnotes.

61.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L^AT_EX definition and modified to create a `mpfootnotes` div class:

```
3860 \def\LWR@endminipage{%
3861     \par
3862     \unskip
3863     \ifvoid\@mpfootins\else
3864         \vskip\skip\@mpfootins
3865         \normalcolor
3866 \LWR@htmldivclass{mpfootnotes}
3867 \LWR@origmedskip
3868     \unvbox\@mpfootins
3869 \LWR@htmldivclassend{mpfootnotes}
3870     \fi
3871     \@minipagefalse
3872     \color@endgroup
3873 \egroup
3874 \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.

```
3875 \newcommand*{\LWR@subminipage}{%
3876 \LWR@stoppars
3877 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3878 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
3879 \LWR@startpars%
3880 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
3881 \newcommand*{\LWR@endsubminipage}{%
3882 \LWR@stoppars%
3883 \LWR@endminipage% The following empty line is required:
3884
3885 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
3886 \newbool{LWR@minipagefullwidth}
3887 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output:

```
3888 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
3889 \end{warppHTML}
```

for PRINT output:

```
3890 \begin{warpprint}
3891 \newcommand*{\minipagefullwidth}{}
3892 \end{warpprint}
```

for HTML output:

```
3893 \begin{warppHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
3894 \newbool{LWR@minipagethispar}
3895 \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.

```
3896 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
3897 {%
```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```

3898 \LWR@traceinfo{starting minipage of width #4}%
3899 \uselengthunit{in}%
3900 \setlength{\LWR@minipagewidth}{#4}%
3901 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
3902 \addtolength{\LWR@minipagewidth}{3em}% room for frames
3903 \setlength{\linewidth}{6in}%
3904 \setlength{\textwidth}{6in}%
3905 \setlength{\textheight}{9in}%
3906 }{}%
3907 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
3908 \addtocounter{\LWR@minipagedepth}{1}%
3909 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
3910 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%
```

L^AT_EX 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.

```

3911 \LWR@stoppars%
```

Create the <div> tag with optional alignment style:

```

3912 \LWR@traceinfo{minipage: creating div class}%
3913 \LWR@htmltag{div class="minipage" style="%
3914 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
3915 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
3916 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
3917 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
3918 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
3919 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
3920 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%
```

Print the width and optional height styles:

```

3921 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
3922 \uselengthunit{PT}%
3923 \ifbool{\LWR@minipagefullwidth}%
3924 {\boolfalse{\LWR@minipagefullwidth}}%
3925 {%
3926 \ifthenelse{\lengthtest{#4}=\linewidth}%
3927 {}%
3928 {width:\rndprintlength{\LWR@minipagewidth} ; }%
3929 }%
3930 \LWR@traceinfo{minipage: about to print the height}%

```

```

3931 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
3932 "{}}%

```

Finish with an empty line to start L^AT_EX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```

3933
3934 \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.

```

3935 \setlength{\linewidth}{#4}% the original width
3936 \setlength{\textwidth}{6in}%
3937 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

3938 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

3939 \LWR@startpars%
3940 \LWR@traceinfo{minipage: finished starting the minipage}%
3941 }

```

End the environment with L^AT_EX processing and closing tag:

```

3942 {%
3943 \LWR@stoppars%
3944 \LWR@endminipage% The following empty line is required:
3945
3946 \LWR@htmldivclassend{minipage}%
3947 \vspace{1\baselineskip}% required for subcaption
3948 \addtocounter{LWR@minipagedepth}{-1}%
3949 \LWR@startpars%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

3950 \global\booltrue{LWR@minipagethispar}%
3951 }

```

61.4 Parbox, makebox, framebox, fbox, raisebox

`\parbox` [*<pos>*] [*<height>*] [*<inner-pos>*] {*<width>*} {*<text>*}

A parbox uses the minipage code:

```
3952 \RenewDocumentCommand{\parbox}{0{t} o 0{t} m +m}
3953 {
3954 \LWR@traceinfo{parbox of width #4}%
3955 \begin{minipage}[#1][#2][#3]{#4}
3956 #5
3957 \end{minipage}
3958 }
```

`\makebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
3959 \let\LWR@origmakebox\makebox
3960
3961 \RenewDocumentCommand{\makebox}{o o m}{%
3962 \mbox{#3}
3963 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
3964 \RenewDocumentCommand{\framebox}{o o m}{%
3965 \fbox{#3}
3966 }
```

`\fbox` {*<text>*}

```
3967 \let\LWR@origfbox\fbox
3968 %
3969 \renewcommand*\fbox[1]{%
3970 \InlineClass{framebox}{#1}%
3971 }
```

`\raisebox` {*<raiselen>*} [*<height>*] [*<depth>*] {*<text>*}

```
3972 \RenewDocumentCommand{\raisebox}{m o o m}{%
3973 #4%
3974 }

3975 \end{warpHTML}
```


62 Direct formatting

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

For high-level block and inline custom CSS classes, see section 29.7.

for HTML output: 3976 `\begin{warpHTML}`

`\emph` $\{\langle text \rangle\}$

3977 `\renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}`

`\textmd` $\{\langle text \rangle\}$

3978 `\renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}`

`\textbf` $\{\langle text \rangle\}$

3979 `\renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}`

`\textrm` $\{\langle text \rangle\}$

3980 `\renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}`

`\textsf` $\{\langle text \rangle\}$

3981 `\renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}`

`\texttt` $\{\langle text \rangle\}$

3982 `\renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}`

`\textup` $\{\langle text \rangle\}$

3983 `\renewcommand{\textup}[1]{\LWR@htmlspan{textup}{#1}}`

`\textit` $\{\langle text \rangle\}$

3984 `\renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}`

`\textsc` $\{\langle text \rangle\}$

3985 `\renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}`

`\textnormal` $\{\langle text \rangle\}$

3986 `\renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}`

`\mdseries`

3987 `\renewcommand*{\mdseries}{}`

`\bfseries`

3988 `\renewcommand*{\bfseries}{}`

`\rmfamily`

3989 `\renewcommand*{\rmfamily}{}`

`\sffamily`

3990 `\renewcommand*{\sffamily}{}`

`\ttfamily`

3991 `\renewcommand*{\ttfamily}{}`

`\upshape`

3992 `\renewcommand*{\upshape}{}`

`\itshape`

3993 `\renewcommand*{\itshape}{}`

`\scshape`

3994 `\renewcommand*{\scshape}{}`

`\scshape`

3995 `\renewcommand*{\normalfont}{}`

`\sp` $\{\langle text \rangle\}$

For siunitx. Must work in math mode.

3996 `\renewcommand{\sp}[1]{\text{^{#1}}}`

`\sb` $\{\langle text \rangle\}$

For siunitx. Must work in math mode.

```
3997 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}\{}}
```

`\textsuperscript` $\{\langle text \rangle\}$

```
3998 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\{\langle text \rangle\}$

```
3999 \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.

```
4000 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup` $\{\langle text \rangle\}$ Prints superscript.

Supports fmtcount package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
4001 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
4002 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
4003 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
4004 \renewcommand*{\dotfill}{\dots}
```

```
4005 \end{warpHTML}
```

63 Skips, spaces, font sizes

for HTML output: 4006 \begin{warpHTML}

\, must be redefined after \RequirePackage{printlen}

```
4007 \let\LWR@origcomma\,
4008 \let\LWR@origtilde~
4009 \let\LWR@origenskip\enskip
4010 \let\LWR@origquad\quad
4011 \let\LWR@origqqquad\qqquad
4012 \let\LWR@orighspace\hspace
4013 \let\LWR@origrule\rule
4014 \let\LWR@origmedskip\medskip
```

Direct-formatting space commands become HTML entities:

```
4015 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space

4016 \renewcommand*{~}{\HTMLentity{nbsp}}
4017 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
4018 \let\LWR@orignormalsize\normalsize
4019 \let\LWR@origsmall\small
4020 \let\LWR@origfootnotesize\footnotesize
4021 \let\LWR@origscriptsize\scriptsize
4022 \let\LWR@origtiny\tiny
4023 \let\LWR@origlarge\large
4024 \let\LWR@origLarge\Large
4025 \let\LWR@origLARGE\LARGE
4026 \let\LWR@orighuge\huge
4027 \let\LWR@origHuge\Huge
4028 \renewcommand*{\normalsize}{}
4029 \renewcommand*{\small}{}
4030 \renewcommand*{\footnotesize}{}
4031 \renewcommand*{\scriptsize}{}
4032 \renewcommand*{\tiny}{}
4033 \renewcommand*{\large}{}
4034 \renewcommand*{\Large}{}
4035 \renewcommand*{\LARGE}{}
4036 \renewcommand*{\huge}{}
4037 \renewcommand*{\Huge}{}
4038
4039 \renewcommand*{\onecolumn}{}
4040
4041 \renewcommand{\twocolumn}[1][{}]{
```

```

4042
4043 #1
4044
4045 }

```

`\newline` Uses HTML `
` tag

```

4046 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
4047 \let\n newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

4048 \NewDocumentCommand{\LWR@endoffline}{s o}
4049 {%
4050 \newline%
4051 }

```

`\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.

```

4052 \newcommand*{\LWR@minipagestartpars}{%
4053 \ifbool{LWR@minipagethispar}%
4054 {%
4055 \LWR@startpars%

```

```
4056 }{}%
4057 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qqquad`'s HTML output.

```
4058 \newcommand*{\LWR@minipagestoppars}{%
4059 \ifbool{LWR@minipagethispar}%
4060 {%
4061 \LWR@stoppars%
4062 }{}%
4063 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
4064 \renewcommand*{\quad}{%
4065 \LWR@minipagestoppars%
4066 \HTMLUnicode{2001}%
4067 \LWR@minipagestartpars%
4068 }
```

`\qqquad` Handles special minipage & horizontal space interactions.

```
4069 \renewcommand*{\qqquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
4070 \renewcommand*{\enskip}{%
4071 \LWR@minipagestoppars%
4072 \HTMLUnicode{2000}%
4073 \LWR@minipagestartpars%
4074 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```
Len \WR@tempheight 4075 \newlength{\LWR@tempwidth}
Len \WR@tempraise 4076 \newlength{\LWR@tempheight}
4077 \newlength{\LWR@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`.

```
4078 \NewDocumentCommand{\LWR@hspace}{s m}{%
4079 \setlength{\LWR@tempwidth}{#2}%
```

If \fill, change to \qqquad:

```
4080 \ifnum\gluestretchorder\LWR@tempwidth>0%
4081 \setlength{\LWR@tempwidth}{2em}%
4082 \fi%
```

Only if the width is not zero:

```
4083 \ifthenelse{\dimtest{\LWR@tempwidth}={}\{0pt\}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4084 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
4085 \ifthenelse{\dimtest{\LWR@tempwidth}={}\{.16667em\}}
4086 {%
4087 \HTMLUnicode{2009}% thin breakable space
4088 }%
```

Print the span with the converted width. Not rounded.

```
4089 {%
4090 \uselengthunit{PT}%
4091 \LWR@htmltagc{%
4092 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
4093 }%
4094 \LWR@htmltagc{/span}%
4095 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4096 \LWR@minipagestartpars%
4097 }%
4098 }
```

`\hspace` * $\{length\}$

Handles special minipage & horizontal space interactions.

```
4099 \let\hspace\LWR@hspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML `br` tag.

```
4100 \renewcommand*{\linebreak}[1][]{\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
4101 \renewcommand*{\nolinebreak}[1][]{}
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
4102 \renewcommand*{\pagebreak}[1][]{
```

```
4103
```

```
4104 }
```

`\nopagebreak` [$\langle num \rangle$]

```
4105 \renewcommand*{\nopagebreak}[1][]{}
```

`\enlargethispage` * $\langle len \rangle$

```
4106 \RenewDocumentCommand{\enlargethispage}{s m}{}%
```

`\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.

```
4107 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary L^AT_EX length, from which comparisons and conversions may be made:

```
4108 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
4109 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
```

```
4110 {}}% zero- width
```

```
4111 }{% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:


```

4112 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
4113 \lengthtest{\LWR@tempwidth<1pt}}%
4114 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

4115 \setlength{\LWR@tempheight}{#3}%
4116 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
4117 \lengthtest{\LWR@tempheight<1pt}}%
4118 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```

4119 \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 L^AT_EX code.

```

4120 \uselengthunit{PT}%
4121 \LWR@htmltagc{%
4122 span
4123 style=" %

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

4124 background:\LWR@currenttextcolor; %

```

The width and height are printed, converted to PT:

```

4125 width:\printlength{\LWR@tempwidth}; %
4126 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 L^AT_EX length allows a typical L^AT_EX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L^AT_EX length limits the allowable syntax. To do: A superior method would compute a ratio of L^AT_EX ex height, then print that to HTML with an ex unit.

```

4127 \IfValueTF{#1}%
4128 {%
4129 \setlength{\LWR@tempraise}{Opt-#1}%
4130 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4131 \LWR@orignewline%
4132 -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4133 \LWR@orignewline%

```

```

4134 -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4135 \LWR@orignewline%
4136 transform: translate(0pt,\printlength{\LWR@tempraise}); %
4137 \LWR@orignewline%
4138 }{}%

```

Display inline-block to place the span inline with the text:

```

4139 display:inline-block;"%
4140 }%
4141 \LWR@htmltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

4142 \LWR@minipagestartpars%
4143 }% non-zero width
4144 }

```

`\rule` [*⟨raise⟩*] {*⟨width⟩*} {*⟨height⟩*}

Handles special minipage & horizontal space interactions.

```

4145 \let\rule\LWR@rule
4146 \end{warpHTML}

```

64 \phantomsection

for HTML output: 4147 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```

4148 \newcommand*\phantomsection{\section*{}}
4149 \end{warpHTML}

```

65 \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>

65.1 HTML logos

for HTML output: 4150 `\begin{warpHTML}`

`\TeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a CSS class used to properly typeset the E and A in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{u}}\mathrm{a}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{C}^{\mathrm{o}}\mathrm{n}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ t, etc.

```
4151 \newcommand*{\LWR@TeX}
4152 {\InlineClass{latexlogofont}%
4153 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\varepsilon}$

`\LaTeXe`

```
4154 \newcommand*{\LWR@LaTeX}
4155 {\InlineClass{latexlogofont}%
4156 {\InlineClass{latexlogo}%
4157 {L\textsuperscript{a}T\textsubscript{e}X}}
4158
4159 \renewcommand*{\LaTeXe}
4160 {\LaTeX\InlineClass{latexlogofont}%
4161 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX` $\mathrm{L}^{\mathrm{u}}\mathrm{a}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{u}}\mathrm{a}\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`\LuaLaTeX`

```
4162 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
4163 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{X}_{\mathrm{E}}\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and spaces $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in $\text{XeL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and spaces $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ appropriately.

```
4164 \newcommand*{\Xe}
4165   {X\textsubscript{\HTMLUnicode{18e}}}%
4166 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
4167 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

`\ConTeXt` $\text{ConT}_{\text{E}}\text{Xt}$

```
4168 \newcommand*{\LWR@ConTeXt}
4169 {\InlineClass{latexlogofont}{Con}\TeX{}}%
4170 \InlineClass{latexlogofont}{t}}
```

`\BibTeX` $\text{BIBT}_{\text{E}}\text{X}$, *MakeIndex*
`\MakeIndex`

```
4171 \providecommand*{\BibTeX}
4172 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
4173
4174 \newcommand*{\MakeIndex}
4175 {\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.

```
4176 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
4177 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX` $\text{MiKT}_{\text{E}}\text{X}$

```
4178 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` LyX

`lyxlogo` is a CSS class used for the LyX logo.

```
4179 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
4180 \end{warpHTML}
```

65.2 Print logos

for PRINT output: 4181 \begin{warpprint}

```

4182 \newcommand*\XeTeXrevE{
4183   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
4184 \providecommand*\XeTeX{\mbox{X\XeTeXrevE\TeX}}
4185 \providecommand*\XeLaTeX{\mbox{X\XeTeXrevE\LaTeX}}
4186 \providecommand*\AMS{%
4187 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
4188 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
4189 \newcommand*\LyX{\textsf{LyX}}
4190 \providecommand*\LuaTeX{\mbox{Lua\TeX}}
4191 \providecommand*\LuaLaTeX{\mbox{Lua\LaTeX}}
4192 \providecommand*\BibTeX{\mbox{B\textsc{ib}\TeX}}
4193 \providecommand*\MakeIndex{\mbox{\textit{MakeIndex}}}
4194 \providecommand*\ConTeXt{\mbox{Con\TeXt}}
4195 \providecommand*\MiKTeX{\mbox{MiK\TeX}}
4196 \end{warpprint}

```

66 \AtBeginDocument, \AtEndDocument

for HTML output: 4197 \begin{warppHTML}

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.
`\LWR@LwarpEnd`

```

4198 \AfterEndPreamble{\LWR@LwarpStart}
4199 \AtEndDocument{\LWR@LwarpEnd}

4200 \end{warppHTML}

```

67 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
```

Defaults unless already over-ridden by the user:

```
4201 \providecommand*\HomeHTMLFileName{\BaseJobname}
4202 \providecommand*\HTMLFileName{}
4203 \providecommand*\UseLatexmk{false}
```

for PRINT output:

```
4204 \begin{warpprint}
4205 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
4206 \immediate\openout{LWR@file}=lwarpmk.conf
4207 \ifwindows
4208 \immediate\write{LWR@file}{opsystem = "Windows"}
4209 \else
4210 \immediate\write{LWR@file}{opsystem = "Unix"}
4211 \fi
4212 \ifPDFTeX
4213 \immediate\write{LWR@file}{latexname = "pdflatex"}
4214 \fi
4215 \ifXeTeX
4216 \immediate\write{LWR@file}{latexname = "xelatex"}
4217 \fi
4218 \ifLuaTeX
4219 \immediate\write{LWR@file}{latexname = "lualatex"}
4220 \fi
4221 \immediate\write{LWR@file}{sourcename = "\jobname"}
4222 \immediate\write{LWR@file}{%
4223 homehtmlfilename = "\HomeHTMLFileName"%
4224 }
4225 \immediate\write{LWR@file}{htmlfilename = "\HTMLFileName"}
4226 \immediate\write{LWR@file}{uselatexmk = "\UseLatexmk"}
4227 \immediate\closeout{LWR@file}
4228 \end{warpprint}
```

68 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-newproject.sty

69 lwarp-newproject

Pkg	lwarp-newproject	lwarp-newproject is used to create lwarp-specific system files in a new project.
Opt lwarp-newproject	warpprint	Configuration files are only created if the warpprint option was used. When this is the case, lwarp-newproject knows that <code>\jobname</code> is the source code's filename (as opposed to the <code><project>_html.tex</code> filename).
Opt lwarp-newproject	warpHTML	Does not create configuration files. warpprint or warpHTML are passed to both lwarp and lwarp-newproject by lwarpmk, depending on whether lwarpmk print or html was used.
Opt lwarp-newproject	BaseJobName	<code>BaseJobname=somename</code> sets <code>\BaseJobname</code> to <code>somename</code> , which is used to set <code>\HomeHTMLFileName</code> , which is then written to the <code>lwarpmk.conf</code> and <code><project>.lwarpmkconf</code> configuration files. <code>\BaseJobname</code> is also written to <code><project>_html.tex</code> as an option given for the lwarp and lwarp-newproject packages during the creation of HTML output. <code>\BaseJobname</code> is the <code>\jobname</code> of the printed version, even while compiling the HTML version, where <code>\jobname</code> has <code>_html</code> appended.
Opt lwarp-newproject	lwarpmk	Tells lwarp-newproject to generate a local copy of lwarpmk called <code>lwarpmk.lua</code> . Useful for archiving for future use. This file may be made executable and acts just like <code>lwarpmk</code> .

In the document source:

```

\documentclass{article} % or book, report
...
(font selection, input encoding)
...
\newcommand*{\HomeHTMLFileName}{index}, or {projectname}
\newcommand*{\HTMLFileName}{}
\usepackage{lwarp-newproject}
\usepackage{lwarp}
...
(load other packages, the rest of the preamble)
...
\NewCSS{project.css}
\NewHTMLdescription{Summarize this webpage.}

```

```

...
\begin{document}
...
\end{document}

```

Place `\usepackage{lwarp-newproject}` just before `\usepackage{lwarp}` in the document source. The operating system and \TeX engine will be auto-detected, and a customized `lwarpmk.conf` file will be generated, along with the other support files (`*.css`, `lwarp_html.xdy`, `lwarp_mathjax.txt`). Rename `sample_project.css` to your own custom `projectname.css` and edit it if desired.

After the first compile, `\usepackage{lwarp-newproject}` may be commented out unless the configuration changes. For example, it may be re-enabled to switch between `pdflatex`, `xelatex`, and `lualatex`, in which case a new manual compile (without using `lwarpmk`) will auto-configure and regenerate the files. It may also be left enabled at all times, in which case the configuration files will be regenerated each time. Note that your own `projectname.css` file will not be over-written by the newly regenerated `sample_project.css`.

Discard all options for `lwarp-newproject`:

```

1 \ProvidesPackage{lwarp-newproject}
2
3 \RequirePackage{etoolbox}
4 \RequirePackage{comment}
5 \RequirePackage{fancyvrb}
6 \RequirePackage{ifplatform}% sense op-system platform
7 \RequirePackage{iftex}% sense pdflatex/lualatex/xelatex
8 \RequirePackage{kvoptions}
9 \SetupKeyvalOptions{family=LWRNP,prefix=LWRNP@}

```

Generate config files?

```

10 \newbool{LWRNP@genconf}
11 \boolfalse{LWRNP@genconf}

```

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

```

12 \excludecomment{LWR@createlwarpmk}

```

Process options:

```

13 \DeclareVoidOption{warpprint}{\booltrue{LWRNP@genconf}}
14 \DeclareVoidOption{warpHTML}{\boolfalse{LWRNP@genconf}}
15 \DeclareVoidOption{lwarpmk}{\includecomment{LWR@createlwarpmk}}
16 \DeclareStringOption[\jobname]{BaseJobname}
17

```

```

18 \booltrue{LWRNP@genconf}% warpprint
19
20 \ProcessKeyvalOptions*\relax

```

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

```

21 \providecommand*{\BaseJobname}{\LWRNP@BaseJobname}

```

Defaults unless already over-ridden by the user:

```

22 % \ifbool{LWRNP@genconf}{
23 \providecommand*{\HomeHTMLFileName}{\BaseJobname}
24 % }{}
25 \providecommand*{\HTMLFileName}{}
26 \providecommand*{\UseLatexmk}{false}

```

69.1 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.

```

27 \ifbool{LWRNP@genconf}{
28 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}
29 \immediate\openout\LWR@file=\jobname_html.tex
30 \immediate\write\LWR@file{%
31 \detokenize{\PassOptionsToPackage}%
32 {warpHTML,BaseJobname=\jobname}{lwarp}%
33 }
34 \immediate\write\LWR@file{%
35 \detokenize{\PassOptionsToPackage}%
36 {warpHTML,BaseJobname=\jobname}{lwarp-newproject}%
37 }
38 \immediate\write\LWR@file{%
39 \detokenize{\input}\string{\jobname.tex}\string }%
40 }
41 \immediate\closeout\LWR@file
42 }{}

```

69.2 project.lwarpmkconf

File `project.lwarpmkconf` The configuration file for `lwarpmk`.

```

43 \ifbool{LWRNP@genconf}{
44 \ifcsdef{LWR@file}{}{\newwrite{\LWR@file}}

```

```

45 \immediate\openout\LWR@file=\jobname.lwarpmkconf
46 \ifwindows
47 \immediate\write\LWR@file{opsystem = "Windows"}
48 \else
49 \immediate\write\LWR@file{opsystem = "Unix"}
50 \fi
51 \ifPDFTeX
52 \immediate\write\LWR@file{latexname = "pdflatex"}
53 \fi
54 \ifXeTeX
55 \immediate\write\LWR@file{latexname = "xelatex"}
56 \fi
57 \ifLuaTeX
58 \immediate\write\LWR@file{latexname = "lualatex"}
59 \fi
60 \immediate\write\LWR@file{sourcename = "\jobname"}
61 \immediate\write\LWR@file{%
62 homehtmlfilename = "\HomeHTMLFileName"%
63 }
64 \immediate\write\LWR@file{htmlfilename = "\HTMLFileName"}
65 \immediate\write\LWR@file{uselatemk = "\UseLatexmk"}
66 \immediate\closeout\LWR@file
67 }{}

```

69.3 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.

```

68 \begin{VerbatimOut}{lwarp.css}
69 /*
70  CSS stylesheet for the LaTeX lwarp package
71  Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
72 */
73
74
75 /* a fix for older browsers: */
76 header, section, footer, aside, nav, main,
77  article, figure { display: block; }
78
79
80 A:link {color:#000080 ; text-decoration: none ; }
81 A:visited {color:#800000 ; }
82 A:hover {color:#000080 ; text-decoration: underline ;}
83 A:active {color:#800000 ; }

```

```
84
85 a.tocpart {display: inline-block ; margin-left: 0em ;
86     font-weight: bold ;}
87 a.tocchapter {display: inline-block ; margin-left: 0em ;
88     font-weight: bold ;}
89 a.tocsection {display: inline-block ; margin-left: 1em ;
90     text-indent: -.5em ; font-weight: bold ; }
91 a.tocsubsection {display: inline-block ; margin-left: 2em ;
92     text-indent: -.5em ; }
93 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
94     text-indent: -.5em ; }
95 a.tocparagraph {display: inline-block ; margin-left: 4em ;
96     text-indent: -.5em ; }
97 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
98     text-indent: -.5em ; }
99 a.tocfigure {margin-left: 0em}
100 a.tocsubfigure {margin-left: 2em}
101 a.toctable {margin-left: 0em}
102 a.tocsubtable {margin-left: 2em}
103 a.toctheorem {margin-left: 0em}
104 a.toclstlisting {margin-left: 0em}
105
106
107 body {
108     font-family: "DejaVu Serif", "Bitstream Vera Serif",
109         "Lucida Bright", Georgia, serif;
110     background: #FAF7F4 ;
111     color: black ;
112     margin: 0em ;
113     padding: 0em ;
114     font-size: 100% ;
115     line-height: 1.2 ;
116 }
117
118 p {margin: 1.5ex 0em 1.5ex 0em ;}
119
120 /* Holds a section number to add space between it and the name */
121 span.sectionnumber { margin-right: .6em }
122
123 /* Inserted in front of index lines */
124 span.indexitem {margin-left: 0em}
125 span.indexsubitem {margin-left: 2em}
126 span.indexsubsubitem {margin-left: 4em}
127
128 div.hidden { display: none ; }
129
130 kbd {
131     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
132         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
133         "Courier New", monospace;
```

```
134     font-size: 100% ;
135 }
136
137 span.strong { font-weight: bold; }
138
139 span.textmd { font-weight: normal; }
140
141 span.textsc { font-variant: small-caps; }
142
143 span.textup { font-variant: normal; }
144
145 span.textrm {
146     font-family: "DejaVu Serif", "Bitstream Vera Serif",
147     "Lucida Bright", Georgia, serif;
148 }
149
150 span.textsf {
151     font-family: "DejaVu Sans", "Bitstream Vera Sans",
152     Geneva, Verdana, sans-serif ;
153 }
154
155 span.attribution {
156     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
157 }
158
159 span.citetitle {
160     margin-left: 1em ; font-size: 80% ; font-style: oblique;
161 }
162
163 span.poemtitle {
164     font-size: 120% ; font-weight: bold;
165 }
166
167 blockquote {
168     margin-left: 0px ;
169     margin-right: 0px ;
170 }
171
172 blockquote p {
173     line-height: 1.5;
174     text-align: left ;
175     font-size: .85em ;
176     margin-left: 3em ;
177     margin-right: 3em ;
178 }
179
180 blockquotation {
181     margin-left: 0px ;
182     margin-right: 0px ;
183 }
```

```
184
185 blockquotation p {
186   line-height: 1.5;
187   text-align: left ;
188   font-size: .85em ;
189   margin-left: 3em ;
190 margin-right: 3em ;
191 }
192
193 div.epigraph {
194   line-height: 1.2;
195   text-align: left ;
196   padding: 3ex 1em 0ex 1em ;
197 /*   margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
198   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
199 /*   margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
200   font-size: .85em ;
201   max-width: 27em ;
202 }
203
204
205
206 div.epigraphsource{
207   text-align:right ;
208   margin-left:auto ;
209 /*   max-width: 50% ; */
210   border-top: 1px solid #A0A0A0 ;
211   padding-bottom: 3ex ;
212   line-height: 1.2;
213 }
214
215 div.epigraph p { padding: .5ex ; margin: 0ex ;}
216 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
217
218
219
220
221 html body {
222   margin: 0 ;
223   line-height: 1.2;
224 }
225
226
227 body div {
228   margin: 0ex;
229 }
230
231
232 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
233 {
```

```
234     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
235             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
236             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
237             "Times New Roman", serif;
238     font-style: normal ;
239     font-weight: bold ;
240     text-align: left ;
241 }
242
243 h1 { /* title of the entire website, used on each page */
244     text-align: center ;
245     font-size: 2.5em ;
246     padding: .4ex 0em 0ex 0em ;
247 }
248 h2 { font-size: 2.25em }
249 h3 { font-size: 2em }
250 h4 { font-size: 1.75em }
251 h5 { font-size: 1.5em }
252 h6 { font-size: 1.25em }
253 span.paragraph {font-size: 1em ; font-variant: normal ;
254     margin-right: 1em ; }
255 span.subparagraph {font-size: 1em ; font-variant: normal ;
256     margin-right: 1em ; }
257
258
259
260 /* Title of the file */
261 h1 {
262     margin: 0ex 0em 0ex 0em ;
263     line-height: 1.3;
264     text-align: center ;
265 }
266
267 /* Part */
268 h2 {
269     margin: 1ex 0em 1ex 0em ;
270     line-height: 1.3;
271     text-align: center ;
272 }
273
274 /* Chapter */
275 h3 {
276     margin: 3ex 0em 1ex 0em ;
277     line-height: 1.3;
278 }
279
280 /* Section */
281 h4 {
282     margin: 3ex 0em 1ex 0em ;
283     line-height: 1.3;
```

```
284 }
285
286 /* Sub-Section */
287 h5 {
288     margin: 3ex 0em 1ex 0em ;
289     line-height: 1.3;
290 }
291
292 /* Sub-Sub-Section */
293 h6 {
294     margin: 3ex 0em 1ex 0em ;
295     line-height: 1.3;
296 }
297
298
299 div.titlepage {
300     text-align: center ;
301 }
302
303 .footnotes {
304     font-size: .85em ;
305     margin: 3ex 1em 0ex 1em ;
306     padding-bottom: 1ex ;
307 }
308
309 .marginpar {
310     max-width:50%;
311     float:right;
312     text-align:left;
313     margin: 1ex 0.5em 1ex 1em ;
314     padding: 1ex 0.5em 1ex 0.5em ;
315     font-size: 85% ;
316     border-top: 1px solid silver ;
317     border-bottom: 1px solid silver ;
318     overflow-x: auto;
319 }
320
321 .marginpar br { margin-bottom: 2ex ; }
322
323 div.marginblock {
324     max-width:50%;
325     float:right;
326     text-align:left;
327     margin: 1ex 0.5em 1ex 1em ;
328     padding: 1ex 0.5em 1ex 0.5em ;
329     overflow-x: auto;
330 }
331
332 div.marginblock div.minipage {
333     display: block ;
```



```
334     margin: 0pt auto 0pt auto ;
335 }
336
337 div.marginblock div.minipage p { font-size: 85%}
338
339 div.marginblock br { margin-bottom: 2ex ; }
340
341
342 section.textbody div.footnotes{
343     margin: 3ex 0em 0ex 0em ;
344     border-bottom: 2px solid silver ;
345 }
346
347 .footnoteheader {
348     border-top: 2px solid silver ;
349     margin-top: 3ex ;
350     padding-top: 1ex ;
351     font-weight: bold ;
352 }
353
354 .mpfootnotes {
355     text-align: left ;
356     font-size: .85em ;
357     margin-left: 1em ;
358     border-top: 1px solid silver ;
359 }
360
361 /* Remove footnote top border in the title page. */
362 div.titlepage div.mpfootnotes {
363     border-top: none ;
364 }
365
366
367
368 ol {
369     margin: 1ex 1em 1ex 0em;
370     line-height: 1.2;
371 }
372
373 ul, body dir, body menu {
374     margin: 1ex 1em 1ex 0em;
375     line-height: 1.2;
376 }
377
378 li { margin: 0ex 0em 1ex 0em; }
379
380 html {
381     margin: 0;
382     padding: 0;
383 }
```

```
384
385 .programlisting {
386   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
387     "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
388     "Courier New", monospace;
389   margin: 1ex 0ex 1ex 0ex ;
390   padding: .5ex 0pt .5ex 0pt ;
391   overflow-x: auto;
392 }
393
394 section.textbody>pre.programlisting {
395 border-top: 1px solid silver ;
396 border-bottom: 1px solid silver ;
397 }
398
399
400 .inlineprogramlisting {
401   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
402     "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
403     "Courier New", monospace;
404   overflow-x: auto;
405 }
406
407
408 div.abstract {
409   margin: 2em 5% 2em 5% ;
410   padding: 1ex 1em 1ex 1em ;
411   /* font-weight: bold ; */
412   font-size: 90% ;
413 }
414
415 div.abstract dl {line-height:1.5;}
416 div.abstract dt {color:#304070;}
417
418 div.abstracttitle{
419   font-family: "URW Classico", Optima, "Linux Biolinum O",
420     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
421     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
422   font-weight:bold;
423   font-size:1.25em;
424   text-align: center ;
425 }
426
427 span.abstracrunintitle{
428   font-family: "URW Classico", Optima, "Linux Biolinum O",
429     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
430     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
431   font-weight:bold;
432 }
433
```

```
434
435 .verbatim {
436     overflow-x: auto ;
437 }
438
439 .alltt {
440     overflow-x: auto ;
441 }
442
443
444 .bverbatim {
445     margin: 1ex Opt 1ex Opt ;
446     padding: .5ex Opt .5ex Opt ;
447     overflow-x: auto ;
448 }
449
450 .lverbatim {
451     margin: 1ex Opt 1ex Opt ;
452     padding: .5ex Opt .5ex Opt ;
453     overflow-x: auto ;
454 }
455
456 .fancyvrb {
457     font-size:.85em ;
458     margin: 3ex Opt 3ex Opt
459 }
460
461 .fancyvrblabel {
462     font-weight:bold;
463     text-align: center ;
464 }
465
466
467 .verse {
468     font-family: "Linux Libertine Mono O", "Lucida Console",
469                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
470                 "Liberation Mono", "FreeMono", "Andale Mono",
471                 "Nimbus Mono L", "Courier New", monospace;
472     margin-left: 1em ;
473 }
474
475
476 div.singlespace { line-height: 1.2 ; }
477 div.onehalfspace { line-height: 1.5 ; }
478 div.doublespace { line-height: 2 ; }
479
480
481
482
483
```

```
484 /* Minipage environments, vertically aligned to top, center, bottom: */
485 .minipage {
486     /* display: inline-block ; */
487     /* Mini pages which follow each other will be tiled. */
488     margin: .25em .25em .25em .25em;
489     padding: .25em .25em .25em .25em;
490     display: inline-flex;
491     flex-direction: column ;
492     overflow: auto;
493 }
494
495 /* Paragraphs in the flexbox did not collapse their margins. */
496 /* Have not yet researched this. */
497 .minipage p {margin: .75ex 0em .75ex 0em ;}
498
499
500
501 .framebox {
502     margin: 0ex ;
503     padding: 0ex ;
504     border: 1px solid black;
505     border-radius: 0px ;
506     padding: .3ex .2em 0ex .2em ;
507     margin: .1ex ;
508     display: inline-block ;
509 }
510
511
512 .mdframed {
513     /* padding: 0ex ; */
514     /* border: 1px solid black; */
515     /* border-radius: 0px ; */
516     padding: 0ex ;
517     margin: 3ex 5% 3ex 5% ;
518     /* display: inline-block ; */
519 }
520
521 .mdframed p { padding: 0ex .5em 0ex .5em ; }
522
523 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
524
525 .mdframedtitle {
526     padding: .5em ;
527     display: block ;
528     font-size: 130%
529 }
530
531 .mdframedsubtitle {
532     padding: 0ex .5em 0ex .5em ;
533     display: block ;
```

```
534     font-size: 115% ;
535 }
536
537 .mdframedsubsubtitle {
538     padding: 0ex .5em 0ex .5em ;
539     display: block ;
540 }
541
542 .mdtheorem {
543     padding: 0ex .5em 0ex .5em ;
544     margin: 3ex 5% 3ex 5% ;
545 /*     display: inline-block ; */
546 }
547
548
549 /* framed package */
550 .framed {
551     margin: 3ex 0em 3ex 0em ;
552     border: 1px solid black;
553     border-radius: 0px ;
554     padding: .3ex 1em 0ex 1em ;
555     display: block ;
556 }
557
558 .snugframed {
559     margin: 3ex 0em 3ex 0em ;
560     border: 1px solid black;
561     border-radius: 0px ;
562     display: block ;
563 }
564
565 .framedleftbar {
566     margin: 3ex 0em 3ex 0em ;
567     border-left: 3pt solid black;
568     border-radius: 0px ;
569     padding: .3ex .2em .3ex 1em ;
570     display: block ;
571 }
572
573 .framedtitle {
574 margin: 0em ;
575 padding: 0em ;
576     font-size: 130%
577 }
578
579 .framedtitle p { padding: .3em }
580
581
582
583 dl {
```

```
584 margin: 1ex 2em 1ex 0em;
585 line-height: 1.3;
586 }
587
588 dl dt {
589   margin-top: 1ex;
590   font-weight: bold;
591 }
592
593 dl dd p { margin-top: 0em; }
594
595
596 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
597   font-family: "URW Classico", Optima, "Linux Biolinum O",
598     "DejaVu Sans", "Bitstream Vera Sans",
599     Geneva, Verdana, sans-serif ;
600   margin-bottom: 4ex ;
601 }
602
603 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
604   line-height: 1.2 ;
605   margin-top:.5ex ;
606   margin-bottom:.5ex;
607   font-size: .9em ;
608 }
609
610
611
612 img, img.hyperimage, img.borderimage {
613   max-width: 600px;
614   border: 1px solid silver;
615   box-shadow: 3px 3px 3px #808080 ;
616   padding: .5% ;
617   margin: .5% ;
618   background: none ;
619 }
620
621 img.inlineimage{
622   padding: 0px ;
623   box-shadow: none ;
624   border: none ;
625   background: none ;
626   margin: 0px ;
627   display: inline-block ;
628   border-radius: 0px ;
629 }
630
631 img.logoimage{
632   max-width: 300px ;
633   box-shadow: 3px 3px 3px #808080 ;
```

```
634     border: 1px solid black ;
635     background:none ;
636     padding:0 ;
637     margin:.5ex ;
638     border-radius: 10px ;
639 }
640
641
642 .section {
643 /*
644     To have each section float relative to each other:
645 */
646 /*
647     display: block ;
648     float: left ;
649     position: relative ;
650     background: white ;
651     border: 1px solid silver ;
652     padding: .5em ;
653 */
654     margin: 0ex .5em 0ex .5em ;
655     padding: 0 ;
656 }
657
658
659 figure {
660     margin: 3ex auto 3ex auto ;
661     padding: 1ex 1em 1ex 1em ;
662     overflow-x: auto ;
663 }
664
665
666 /* To automatically center images in figures: */
667 /*
668 figure img.inlineimage {
669     margin: 0ex auto 0ex auto ;
670     display: block ;
671 }
672 */
673
674 /* To automatically center minipages in figures: */
675 /*
676 figure div.minipage, figure div.minipage div.minipage {
677     margin: 1ex auto 1ex auto ;
678     display: block ;
679 }
680 */
681
682 figure div.minipage p { font-size: 85% ; }
683
```

```
684 figure.subfigure, figure.subtable {
685     display: inline-block ; margin: 3ex 1em 3ex 1em ;
686 }
687
688 figcaption .minipage { margin:0 ; padding: 0 }
689
690 div.floatrow { text-align: center; }
691
692 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
693
694 div.floatfoot { font-size: .85em ;
695     border-top: 1px solid silver ; line-height: 1.2 ; }
696
697 figcaption , .lstlistingtitle {
698     font-size: .85em ;
699     text-align: center ;
700     font-weight: bold ;
701     margin-top: 1ex ;
702     margin-bottom: 1ex ;
703 }
704
705 figure.subfigure figcaption, figure.subtable figcaption {
706     border-bottom: none ; background: none ;
707 }
708
709 div.nonfloatcaption {
710     margin: 1ex auto 1ex auto ;
711     font-size: .85em ;
712     text-align: center ;
713     font-weight: bold ;
714 }
715
716 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
717 figure div.floatrow div.minipage figcaption {
718     border: none ;
719     background: none ;
720 }
721
722
723 table {
724     margin: 1ex auto 1ex auto ;
725     border-collapse: collapse ;
726     border-spacing: 0px ;
727     line-height: 1.3 ;
728 }
729
730 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
731     margin-bottom: 0ex ; } /* for \hline */
732
733 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
```



```

734     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
735
736 td {padding: 1ex .5em 1ex .5em ;}
737
738 table td.tdl { text-align: left ; vertical-align: middle ; }
739 table td.tdc { text-align: center ; vertical-align: middle ; }
740 table td.tdr { text-align: right ; vertical-align: middle ; }
741 table td.tdp { text-align: left ; vertical-align: bottom ; }
742 table td.tdm { text-align: left ; vertical-align: middle ; }
743 table td.tdb { text-align: left ; vertical-align: top ; }
744 table td.tdP { text-align: center ; vertical-align: bottom ; }
745 table td.tdM { text-align: center ; vertical-align: middle ; }
746 table td.tdB { text-align: center ; vertical-align: top ; }
747 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
748     vertical-align: middle ; } /* for cmidrule */
749 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
750     vertical-align: middle ; }
751 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
752     vertical-align: middle ; }
753 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
754     vertical-align: bottom ; }
755 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
756     vertical-align: middle ; }
757 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
758     vertical-align: top ; }
759 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
760     vertical-align: bottom ; }
761 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
762     vertical-align: middle ; }
763 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
764     vertical-align: top ; }
765
766 /* Margins of paragraphs inside table cells: */
767 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
768     padding-bottom: 1ex ; margin: 0ex ; }
769 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
770     padding-bottom: 1ex ; margin: 0ex ; }
771 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
772     padding-bottom: 1ex ; margin: 0ex ; }
773
774 td.tdp , td.tdprule , td.tdP , td.tdPrule
775     { padding: 0ex .5em 0ex .5em ; }
776 td.tdm , td.tdmrule , td.tdM , td.tdMrule
777     { padding: 0ex .5em 0ex .5em ; }
778 td.tdb , td.tdbrule , td.tdB , td.tdBrule
779     { padding: 0ex .5em 0ex .5em ; }
780
781
782 /* table notes: */
783 .tnotes {

```

```
784     margin: 0ex 5% 1ex 5% ;
785     padding: 0.5ex 1em 0.5ex 1em;
786     font-size:.85em;
787     text-align: left ;
788 }
789
790 .tnotes dl dt p {margin-bottom:0px;}
791
792 .tnoteitemheader {margin-right: 1em;}
793
794
795
796 /* center, flushleft, flushright environments */
797 div.center{text-align:center;}
798 div.center table {margin-left:auto;margin-right:auto;}
799 div.flushleft{text-align:left;}
800 div.flushleft table {margin-left:0em ; margin-right:auto;}
801 div.flushright{text-align:right;}
802 div.flushright table {margin-left:auto ; margin-right: 0em ;}
803
804
805
806
807 /* program listing callouts: */
808 span.callout {
809     font-family: "DejaVu Sans", "Bitstream Vera Sans",
810         Geneva, Verdana, sans-serif ;
811     border-radius: .5em;
812     background-color:black;
813     color:white;
814     padding:0px .25em 0px .25em;
815     margin: 0 ;
816     font-weight: bold;
817     font-size:.72em ;
818 }
819
820 div.programlisting pre.verbatim span.callout{
821 font-size: .85em ;
822 }
823
824
825
826
827
828 div.published
829 {
830     text-align: center ;
831     font-variant: normal ;
832     font-style: italic ;
833     font-size: 1em ;
```

```
834     margin: 3ex 0em 3ex 0em ;
835 }
836
837 div.subtitle
838 {
839     text-align: center ;
840     font-variant: normal ;
841     font-style: italic ;
842     font-size: 1.25em ;
843     margin: 3ex 0em 3ex 0em ;
844 }
845
846 div.subtitle p { margin: 1ex ; }
847
848 div.author
849 {
850     font-variant: normal ;
851     font-style: normal ;
852     font-size: 1em ;
853     margin: 3ex 0em 3ex 0em ;
854 }
855
856 div.author table {
857     margin: 3ex auto 0ex auto ;
858     background: none ;
859 }
860
861 div.author table tbody tr td { padding: .25ex ; }
862
863 span.affiliation {font-size: .85em ; font-variant: small-caps; }
864
865 div.titledate {
866     text-align: center ;
867     font-size: .85em ;
868     font-style: italic;
869     margin: 6ex 0em 6ex 0em ;
870 }
871
872
873 nav.topnavigation{
874     text-align: left ;
875     padding: 0.5ex 1em 0.5ex 1em ;
876 /*     margin: 2ex 0em 3ex 0em ; */
877     margin: 0 ;
878     border-bottom: 1px solid silver ;
879     border-top: 1px solid silver ;
880     clear:right ;
881 }
882
883 nav.botnavigation{
```

```
884     text-align: left ;
885     padding: 0.5ex 1em 0.5ex 1em ;
886 /*     margin: 3ex 0em 2ex 0em ; */
887     margin: 0 ;
888     border-top: 1px solid silver ;
889     border-bottom: 1px solid silver ;
890     clear:right ;
891 }
892
893
894 header{
895     line-height: 1.2 ;
896     font-size: 1em ;
897 /*     border-bottom: 2px solid silver ; */
898     margin: 0px ;
899     padding: 0ex 1em 0ex 1em ;
900     text-align:center ;
901 }
902
903 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
904
905
906 footer{
907     font-size: .85em ;
908     line-height: 1.2 ;
909     margin-top: 1ex ;
910     border-top: 2px solid silver ;
911     padding: 2ex 1em 2ex 1em ;
912     clear:right ;
913     text-align:left ;
914 }
915
916
917 a.linkhome { font-weight:bold ; font-size: 1em ;}
918
919
920 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
921
922 img.lateximage{
923     padding: 0px 0px 0px 0px ;
924     box-shadow: none ;
925     border: none ;
926     background: none ;
927     margin: 0px 0px -.15ex 0px ;
928     /* pdfcrop leaves a slight margin, adjust to baseline */
929     max-width: 100% ;
930     border-radius: 0ex ;
931     border: none ;
932 }
933
```

```
934
935
936 nav.sidetoc {
937     font-family: "DejaVu Serif", "Bitstream Vera Serif",
938         "Lucida Bright", Georgia, serif;
939     float:right ;
940     width: 20%;
941     border-left: 1px solid silver;
942     border-top: 1px solid silver;
943     border-bottom: 1px solid silver;
944 /*     border-top: 2px solid #808080 ; */
945     background: #FAF7F4 ;
946     padding: 2ex 0em 2ex 1em ;
947     margin: 0ex 0em 2ex 1em ;
948     font-size:.9em ;
949     border-radius: 20px 0px 0px 20px ;
950 }
951
952 div.sidetoccontents {
953 /*     border-top: 1px solid silver ; */
954     overflow-y: auto ;
955     width: 100% ;
956     text-align: left ;
957 }
958
959 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
960     text-indent: 0 ; }
961 nav.sidetoc p a {color:black ; font-size: .7em ;}
962 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
963     border-bottom: 1px solid silver ; }
964 nav.sidetoc a:hover {text-decoration: underline ; }
965
966
967
968 section.textbody { margin: 0ex 1em 0ex 1em ;}
969
970
971 div.multicolsheading { -webkit-column-span: all;
972     -moz-column-span: all; column-span: all; }
973 div.multicols { -webkit-columns: 3 380px ;
974     -moz-columns: 3 380px ; columns: 3 380px ; }
975 div.multicols p {margin-top: 0ex}
976
977
978
979 /* Used to support algorithmicx: */
980 span.floatright { float: right ; }
981
982
983
```

```
984
985 /* Native LaTeX theorems: */
986
987 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
988 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
989
990
991 /* theorem, amsthm, and ntheorem packages */
992
993 span.theoremheader,
994 span.theoremheaderplain,
995 span.theoremheaderdefinition,
996 span.theoremheaderbreak,
997 span.theoremheadermarginbreak,
998 span.theoremheaderchangebreak,
999 span.theoremheaderchange,
1000 span.theoremheadermargin
1001 {
1002 font-style:normal ; font-weight: bold ; margin-right: 1em ;
1003 }
1004
1005 span.amsthmnameplain,
1006 span.amsthmnamedefinition,
1007 span.amsthmnumberplain,
1008 span.amsthmnumberdefinition
1009 {
1010 font-style:normal ; font-weight: bold ;
1011 }
1012
1013
1014 span.amsthmnameremark,
1015 span.amsthmnumberremark
1016 {font-style:italic ; font-weight: normal ; }
1017
1018
1019 span.amsthmnoteplain,
1020 span.amsthmnotedefinition
1021 {font-style:normal ;}
1022
1023
1024 span.theoremheaderremark,
1025 span.theoremheaderproof,
1026 span.amsthmproofname
1027 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1028
1029 span.theoremheadersc
1030 {
1031 font-style:normal ;
1032 font-variant: small-caps ;
1033 font-weight: normal ;
```

```
1034 margin-right: 1em ;
1035 }
1036
1037 .theoremdemark {float:right}
1038
1039 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1040 div.theorembodybreak, div.theorembodynonumberbreak,
1041 div.theorembodymarginbreak,
1042 div.theorembodychangebreak,
1043 div.theorembodychange,
1044 div.theorembodymargin
1045 {
1046 font-style:italic;
1047 margin-top: 3ex ; margin-bottom: 3ex ;
1048 }
1049
1050 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1051 div.theorembodyplainupright, nonumberplainuprightsc,
1052 div.amsthmbodydefinition, div.amsthmbodyremark,
1053 div.amsthmproof
1054 {
1055 font-style: normal ;
1056 margin-top: 3ex ; margin-bottom: 3ex ;
1057 }
1058
1059 span.amsthmnoteremark {}
1060
1061
1062
1063 /*
1064 For CSS LaTeX and related logos:
1065 Based on:
1066 http://edward.oconnor.cx/2007/08/tex-poshlet
1067 http://nitens.org/taraborelli/texlogo
1068 */
1069
1070 .latexlogofont {
1071     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1072                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1073     font-variant: normal ;
1074 }
1075
1076 .latexlogo {
1077     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1078                 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1079     letter-spacing: .03em ;
1080     font-size: 1.1em;
1081 }
1082
1083 .latexlogo sup {
```

```
1084 text-transform: uppercase;
1085 letter-spacing: .03em ;
1086 font-size: 0.85em;
1087 vertical-align: 0.15em;
1088 margin-left: -0.36em;
1089 margin-right: -0.15em;
1090 }
1091
1092 .latexlogo sub {
1093 text-transform: uppercase;
1094 vertical-align: -0.5ex;
1095 margin-left: -0.1667em;
1096 margin-right: -0.125em;
1097 font-size: 1em;
1098 }
1099
1100 .xetexlogo {
1101 font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1102 "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1103 letter-spacing: .03em ;
1104 font-size: 1.1em;
1105 }
1106
1107 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1108 .xetexlogo sub {
1109 text-transform: uppercase;
1110 vertical-align: -0.5ex;
1111 margin-left: -0.0667em;
1112 margin-right: -0.2em;
1113 font-size: 1em;
1114 letter-spacing: .03em ;
1115 }
1116
1117 /* A large gap between Xe and LaTeX v.s. TeX: */
1118 .xelatexlogo sub {
1119 text-transform: uppercase;
1120 vertical-align: -0.5ex;
1121 margin-left: -0.0667em;
1122 margin-right: -.05em;
1123 font-size: 1em;
1124 letter-spacing: .03em ;
1125 }
1126
1127 .amslogo {
1128 font-family: "TeXGyreChorus", "URW Chancery L",
1129 "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1130 "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1131 "Hoefler Text", Times, "Times New Roman", serif;
1132 font-style: italic;
1133 }
```



```
1134
1135 .lyxlogo {
1136     font-family: "URW Classico", Optima, "Linux Biolinum O",
1137         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1138     Verdana, sans-serif ;
1139 }
1140
1141
1142
1143
1144 /* Only display top and bottom navigation if a small screen: */
1145 /* Hide the sidetoc if a small screen: */
1146 nav.topnavigation { display:none; }
1147 nav.botnavigation { display:none; }
1148
1149 @media screen and (max-width: 45em) {
1150     /*     nav.sidetoc {display:none;} */
1151     nav.sidetoc {
1152         float: none ;
1153         width: 100% ;
1154         margin: 5ex 0px 5ex 0px ;
1155         padding: 0 ;
1156         border-radius: 0 ;
1157         border-bottom: 1px solid black ;
1158         border-top: 1px solid black ;
1159         box-shadow: none ;
1160     }
1161     /*     nav.topnavigation { display:block } */
1162     nav.botnavigation { display:block }
1163     .marginpar {
1164         max-width: 100%;
1165         float: none;
1166         display:block ;
1167         margin: 1ex 1em 1ex 1em ;
1168     }
1169 }
1170
1171 @media print {
1172     body {
1173         font-family: "Linux Libertine O",
1174             "DejaVu Serif", "Bitstream Vera Serif",
1175             "Liberation Serif", "Nimbus Roman No 9 L",
1176             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1177     }
1178     nav.sidetoc { display:none; }
1179     nav.topnavigation { display: none; }
1180     nav.botnavigation { display: none; }
1181 }
1182
1183 @media handheld {
```

```

1184     nav.sidetoc { display:none; }
1185     nav.topnavigation { display:block }
1186     nav.botnavigation { display:block }
1187 }
1188
1189 @media projection {
1190     nav.sidetoc { display:none; }
1191     nav.topnavigation { display:block }
1192     nav.botnavigation { display:block }
1193 }
1194 \end{VerbatimOut}
1195 % \end{Verbatim}% for syntax highlighting

```

69.4 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.

```

1196 \begin{VerbatimOut}{lwarp_sagebrush.css}
1197 @import url("lwarp.css") ;
1198
1199
1200 A:link {color:#105030 ; text-decoration: none ; }
1201 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1202 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1203 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1204
1205
1206
1207 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1208 {
1209     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1210         "Linux Libertine 0", "Liberation Serif",
1211         "Nimbus Roman No 9 L", "FreeSerif",
1212         "Hoefler Text", Times, "Times New Roman", serif;
1213     font-variant: small-caps ;
1214     font-weight: normal ;
1215     color: #304070 ;
1216     text-shadow: 2px 2px 3px #808080;
1217 }
1218
1219 h1 { /* title of the entire website, used on each page */
1220     font-variant: small-caps ;
1221     color: #304070 ;
1222     text-shadow: 2px 2px 3px #808080;

```

```
1223     background-color: #F7F7F0 ;
1224     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1225 }
1226
1227 h1 {
1228     border-bottom: 1px solid #304070;
1229     border-top: 2px solid #304070;
1230 }
1231
1232 h2 {
1233     border-bottom: 1px solid #304070;
1234     border-top: 2px solid #304070;
1235     background-color: #F7F7F0 ;
1236     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1237 }
1238
1239
1240
1241 div.abstract {
1242     background: #f5f5eb ;
1243     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1244
1245     border: 1px solid silver;
1246     border-radius: 1em ;
1247 }
1248
1249 div.abstract dl {line-height:1.5;}
1250 div.abstract dt {color:#304070;}
1251
1252 div.abstracttitle{
1253     font-family: "URW Classico", Optima, "Linux Biolinum O",
1254         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1255         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1256     font-weight:bold;
1257     font-variant: small-caps ;
1258     font-size:1.5em;
1259     border-bottom: 1px solid silver ;
1260     color: #304070 ;
1261     text-align: center ;
1262     text-shadow: 1px 1px 2px #808080;
1263 }
1264
1265 span.abstracrunintitle{
1266     font-family: "URW Classico", Optima, "Linux Biolinum O",
1267         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1268         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1269     font-weight:bold;
1270 }
1271
1272
```

```
1273 div.epigraph {
1274     background: #f5f5eb ;
1275     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1276
1277     border: 1px solid silver ;
1278     border-radius: 1ex ;
1279     box-shadow: 3px 3px 3px #808080 ;
1280 }
1281
1282
1283 .example {
1284     background-color: #f5f5eb ;
1285     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1286
1287 }
1288
1289 div.exampletitle{
1290     font-family: "URW Classico", Optima, "Linux Biolinum O",
1291         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1292         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1293     font-weight:bold;
1294     font-variant: small-caps ;
1295     border-bottom: 1px solid silver ;
1296     color: #304070 ;
1297     text-align: center ;
1298     text-shadow: 1px 1px 2px #808080;
1299 }
1300
1301
1302 .sidebar {
1303     background-color: #f5f5eb ;
1304     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1305
1306 }
1307
1308 div.sidebartitle{
1309     font-family: "URW Classico", Optima, "Linux Biolinum O",
1310         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1311         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1312     font-weight:bold;
1313     font-variant: small-caps ;
1314     border-bottom: 1px solid silver ;
1315     color: #304070 ;
1316     text-align: center ;
1317     text-shadow: 1px 1px 2px #808080;
1318 }
1319
1320
1321 .fancyvrblabel {
1322     font-family: "URW Classico", Optima, "Linux Biolinum O",
```

```
1323         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1324         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1325     font-weight:bold;
1326     font-variant: small-caps ;
1327 font-size: 1.5em ;
1328     color: #304070 ;
1329     text-align: center ;
1330     text-shadow: 1px 1px 2px #808080;
1331 }
1332
1333
1334
1335 .minipage {
1336     background-color: #eeeeee7 ;
1337     border: 1px solid silver ;
1338     border-radius: 1ex ;
1339 }
1340
1341 .framed .minipage , .framedleftbar .minipage {
1342 border: none ;
1343 background: none ;
1344 padding: 0ex ;
1345 margin: 0ex ;
1346 }
1347
1348 figure.figure .minipage, figcaption .minipage { border: none; }
1349
1350 div.marginblock div.minipage { border: none; }
1351
1352 figure , div.marginblock {
1353     background-color: #eeeeee7 ;
1354     border: 1px solid silver ;
1355     border-radius: 1ex ;
1356     box-shadow: 3px 3px 3px #808080 ;
1357 }
1358
1359 figure figure {
1360     border: 1px solid silver ;
1361     margin: 0em ;
1362     box-shadow: none ;
1363 }
1364
1365 /*
1366 figcaption {
1367     border-top: 1px solid silver ;
1368     border-bottom: 1px solid silver ;
1369     background-color: #e8e8e8 ;
1370 }
1371 */
1372
```

```
1373
1374 div.table {
1375     box-shadow: 3px 3px 3px #808080 ;
1376 }
1377
1378 /*
1379 .tnotes {
1380     background: #e8e8e8;
1381     border: 1px solid silver;
1382 }
1383 */
1384
1385
1386 nav.topnavigation{
1387     background-color: #b0b8b0 ;
1388     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1389 }
1390
1391 nav.botnavigation{
1392     background-color: #b0b8b0 ;
1393     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1394 }
1395
1396
1397
1398 header{
1399     background-color: #F7F7F0 ;
1400     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1401 }
1402
1403 footer{
1404     background-color: #F7F7F0 ;
1405     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1406 }
1407
1408
1409
1410 nav.sidetoc {
1411     background-color: #F7F7F0 ;
1412     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1413     box-shadow: 3px 3px 3px #808080 ;
1414     border-radius: 0px 0px 0px 20px ;
1415 }
1416
1417 div.sidetoctitle {color: #304070 ; }
1418 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1419
1420
1421 @media screen and (max-width: 45em) {
1422     nav.sidetoc { border-radius: 0 ; }
```

```

1423 }
1424
1425
1426 \end{VerbatimOut}
1427 % \end{Verbatim}% for syntax highlighting

```

69.5 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.

```

1428 \begin{VerbatimOut}{lwarp_formal.css}
1429 @import url("lwarp.css") ;
1430
1431
1432
1433 A:link {color:#802020 ; text-decoration:none; }
1434 A:visited {color:#802020 ; text-shadow:none ;}
1435 A:hover {color:#400000 ; text-shadow:none ;}
1436 A:active {color:#C00000 ; text-shadow:none ;}
1437
1438
1439 body {
1440     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1441         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1442         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1443         "Times New Roman", serif;
1444     background: #fffcf5;
1445 }
1446
1447 span.textrm {
1448     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1449         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1450         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1451         "Times New Roman", serif;
1452 }
1453
1454 span.textsf {
1455     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1456         Geneva, Verdana, sans-serif ;
1457 }
1458
1459
1460
1461 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph

```

```
1462 {
1463     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1464         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1465         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1466         "Times New Roman", serif;
1467     color: #800000 ;
1468     text-shadow: none ;
1469 }
1470
1471 h1, h2 {
1472     background-color: #fffcf5 ;
1473     background-image: none ;
1474     border-bottom: 1px solid #808080;
1475     border-top: 2px solid #808080;
1476 }
1477
1478 div.abstracttitle {
1479     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1480         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1481         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1482         "Times New Roman", serif;
1483     color: black ;
1484     text-shadow: none ;
1485 }
1486
1487 span.abstractrunintitle {
1488     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1489         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1490         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1491         "Times New Roman", serif;
1492     color: black ;
1493     text-shadow: none ;
1494 }
1495
1496 div.abstract { font-size: 100% }
1497
1498 .sidebar {
1499     background: #fffcf5;
1500     background-image: none ;
1501     margin: 2em 5% 2em 5%;
1502     padding: 0.5em 1em;
1503     border: none ;
1504     border-top : 1px solid silver;
1505     border-bottom : 1px solid silver;
1506     font-size: 90% ;
1507 }
1508
1509 div.sidebartitle{
1510     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1511         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```



```
1512         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1513         "Times New Roman", serif;
1514     color: #800000 ;
1515     text-shadow: none ;
1516     border: none ;
1517 }
1518
1519 .example {
1520     background: #fffcf5;
1521     background-image: none ;
1522     margin: 2em 5% 2em 5%;
1523     padding: 0.5em 1em;
1524     border: none ;
1525     border-top : 1px solid silver;
1526     border-bottom : 1px solid silver;
1527 }
1528
1529 div.exampletitle{
1530     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1531         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1532         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1533         "Times New Roman", serif;
1534     color: #800000 ;
1535     text-shadow: none ;
1536     border: none ;
1537 }
1538
1539 div.fancyvrblabel{
1540     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1541         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1542         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1543         "Times New Roman", serif;
1544     color: #800000 ;
1545     text-shadow: none ;
1546     border: none ;
1547 }
1548
1549
1550
1551 .verse {
1552     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1553         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1554         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1555         "Times New Roman", serif;
1556 }
1557
1558
1559 figure {
1560     margin: 3ex 5% 3ex 5% ;
1561     padding: 1ex 1em 1ex 1em ;
```

```
1562     background-color: #fffcf5 ;
1563     overflow-x: auto ;
1564     border: none ;
1565 /*     border-top: 1px solid silver; */
1566 /*     border-bottom: 1px solid silver; */
1567 }
1568
1569
1570 figcaption , .lstlisting {
1571     border: none ;
1572 /*     border-top: 1px solid silver ; */
1573 /*     border-bottom: 1px solid silver ; */
1574     background-color: #fffcf5 ;
1575 }
1576
1577 .tnotes {
1578     background: #fffcf5 ;
1579 }
1580
1581 .theorem {
1582     background: none ;
1583 }
1584
1585 .minipage {
1586     background-color: #fffcf5 ;
1587     border: none ;
1588 }
1589
1590 div.floatrow figure { border: none ; }
1591
1592 figure figure { border: none ; }
1593
1594
1595 nav.toc, nav.lof, nav.lot, nav.lol {
1596     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1597         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1598         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1599         "Times New Roman", serif;
1600 }
1601
1602 nav.sidetoc {
1603     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1604         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1605         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1606         "Times New Roman", serif;
1607     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
1608     border-radius: 0px 0px 0px 20px ;
1609 }
1610
1611 div.sidetoc{title{
```

```

1612     color: #800000 ;
1613 }
1614
1615 header{
1616     background-color: #e0e0e0 ;
1617     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
1618     text-align:center ;
1619 }
1620
1621 footer{
1622     background-color: #e0e0e0 ;
1623     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
1624     padding: 2ex 1em 2ex 1em ;
1625     clear:right ;
1626     text-align:left ;
1627 }
1628
1629 nav.botnavigation {
1630     background: #dedcd5 ;
1631     border-top: 1px solid black ;
1632 }
1633 \end{VerbatimOut}
1634 % \end{Verbatim}% for syntax highlighting

```

69.6 sample_project.css

File `sample_project.css` The project-specific CSS file. Use with `\NewCSS`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```

1635 \begin{VerbatimOut}{sample_project.css}
1636 /* ( --- Start of project.css --- ) */
1637 /* A sample project-specific CSS file for lwarp --- ) */
1638
1639 /* Load default lwarp settings: */
1640 @import url("lwarp.css") ;
1641 /* or lwarp_formal.css, lwarp_sagebrush.css */
1642
1643 /* Project-specific CSS setting follow here. */
1644 /* . . . */
1645
1646 /* ( --- End of project.css --- ) */
1647 \end{VerbatimOut}
1648 % \end{Verbatim}% for syntax highlighting

```

69.7 lwarp_html.xdy

File `lwarp_html.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.

```

1649 \begin{VerbatimOut}{lwarp_html.xdy}
1650 (require "tex/inputenc/latin.xdy")
1651 (merge-rule "\\PS *" "Postscript")
1652 (require "texindy.xdy")
1653 (require "page-ranges.xdy")
1654 (require "book-order.xdy")
1655 (markup-locref :open "\hyperindexref{" :close "{")
1656 \end{VerbatimOut}
1657 % \end{Verbatim}% for syntax highlighting

```

69.8 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.

```

1658 \begin{VerbatimOut}{lwarp_mathjax.txt}
1659 <!-- https://groups.google.com/forum/#!topic/
1660                               mathjax-users/jUtewUcE2bY -->
1661 <script type="text/x-mathjax-config">
1662 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1663     var seteqsectionDefault = {name: "", num: 0};
1664     var seteqsections = {}, seteqsection = seteqsectionDefault;
1665     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1666     var AMS = MathJax.Extension["TeX/AMSmath"];
1667     TEX.Definitions.Add({
1668     macros: {
1669         seteqsection: "mySection",
1670         seteqnumber: "mySetEqNumber"
1671     }
1672     });
1673
1674     PARSE.Augment({
1675     mySection: function (name) {
1676         seteqsection.num = AMS.number;
1677         var n = this.GetArgument(name);
1678         if (n === "") {
1679             seteqsection = seteqsectionDefault;
1680         } else {

```

```

1681         if (!seteqsections["_"+n])
1682             seteqsections["_"+n] = {name:n, num:0};
1683         seteqsection = seteqsections["_"+n];
1684     }
1685     AMS.number = seteqsection.num;
1686 },
1687 mySetEqNumber: function (name) {
1688     var n = this.GetArgument(name);
1689     if (!n || !n.match(/^ *[0-9]+ *$/))
1690         n = ""; else n = parseInt(n)-1;
1691     <!-- $ syntax highlighting -->
1692     if (n === "" || n < 1)
1693         TEX.Error
1694             ("Argument to "+name+" should be a positive integer");
1695     AMS.number = n;
1696 }
1697 });
1698 MathJax.Hub.Config({
1699   TeX: {
1700     equationNumbers: {
1701       formatTag: function (n)
1702         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
1703       formatID: function (n) {
1704         n = (seteqsection.name+'.'+n).replace
1705           (/[:'"<>&]/g,"").replace(/\./,"");
1706         return 'mjax-eqn-' + n;
1707       }
1708     }
1709   }
1710 });
1711 }>
1712 </script>
1713
1714 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1715 <script type="text/x-mathjax-config">
1716   MathJax.Ajax.config.path["Contrib"] =
1717     "https://cdn.mathjax.org/mathjax/contrib";
1718 </script>
1719
1720 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1721                                     tree/master/siunitx -->
1722 <script type="text/x-mathjax-config">
1723   MathJax.Hub.Config({
1724     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
1725     jax: ["input/TeX", "output/HTML-CSS"],
1726     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
1727     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}
1728   });
1729 </script>
1730

```

```

1731 <script type="text/x-mathjax-config">
1732 MathJax.Hub.Config({
1733     TeX: {
1734         equationNumbers: {
1735             autoNumber: "AMS"
1736         }
1737     }
1738 });
1739 </script>
1740
1741 <script
1742     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1743 </script>
1744 \end{VerbatimOut}
1745 % \end{Verbatim}% for syntax highlighting

```

69.9 Lwarpmk option

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

```

1746 \begin{LWR@createlwarpmk}

```

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

```

1747 \begin{VerbatimOut}{lwarpmk.lua}
1748 #!/usr/bin/env texlua
1749
1750 -- Copyright 2016-2017 Brian Dunn
1751
1752 -- Print the usage of the lwarpmk command:
1753
1754 printversion = "v0.26"
1755
1756 function printhelp ()
1757 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1758 end
1759
1760 function printusage ()
1761 print ( [[
1762
1763 lwarpmk print [project]: Compile a print version.
1764 lwarpmk printindex [project]: Process the index for the print version.
1765 lwarpmk html [project]: Compile an HTML version.
1766 lwarpmk htmlindex [project]: Process the index for the html version.
1767 lwarpmk again [project]: Touch the source code to trigger recompiles.
1768 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
1769 lwarpmk pdftohtml [project]:

```

```

1770     For use with latexmk or a Makefile:
1771     Convert project_html.pdf to project_html.html and
1772     individual HTML files.
1773 lwarpmk clean [project]: Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
1774 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1775 lwarpmk -h: Print this help message.
1776 lwarpmk --help: Print this help message.
1777
1778 ]] )
1779 printconf ()
1780 end
1781
1782 -- Print the format of the configuration file lwarpmk.conf:
1783
1784 function printconf ()
1785 print ( [[
1786 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1787 --
1788 opsystem = "Unix"      (or "Windows")
1789 latexname = "pdflatex" (or "lualatex", or "xelatex")
1790 sourcename = "projectname" (the source-code filename w/o .tex)
1791 homehtmlfilename = "index" (or perhaps the project name)
1792 htmlfilename = ""      (or "projectname" - filename prefix)
1793 uselatexmk = "false"   (or "true" to use latexmk to build PDFs)
1794 --
1795 Filenames must contain only letters, numbers, underscore, or dash.
1796 Values must be in "quotes".
1797
1798 ]] ) ;
1799 end
1800
1801
1802 -- Split one large sourcefile into a number of files,
1803 -- starting with destfile.
1804 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1805
1806 function splitfile (destfile,sourcefile)
1807 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1808 io.input(sourcefile)
1809 io.output(destfile)
1810 for line in io.lines() do
1811 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
1812 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1813 io.output(newfilename) ;
1814 else -- not a splitpoint
1815 io.write (line .. "\n") ;
1816 end
1817 end -- do
1818 end -- function
1819

```

```
1820 -- Incorrect value, so print an error and exit.
1821
1822 function cvalueerror ( line, linenum , cvalue )
1823     print ( linenum .. " : " .. line ) ;
1824     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
1825     printconf ( ) ;
1826     os.exit(1) ;
1827 end
1828
1829 -- Load settings from the project's "lwarpmk.conf" file:
1830
1831 function loadconf ( )
1832 -- Default configuration filename:
1833 local conffile = "lwarpmk.conf"
1834 -- Optional configuration filename:
1835 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1836 -- Verify the file exists:
1837 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1838 print("lwarpmk: " .. conffile .. " does not exist.")
1839 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1840 printhelp ( ) ;
1841 os.exit(1) -- exit the entire lwarpmk script
1842 else -- file exists
1843 -- Read the file:
1844 print ("lwarpmk: Reading " .. conffile .. ".")
1845 io.input(conffile) ;
1846 -- Scan each line:
1847 local linenum = 0
1848 for line in io.lines() do -- scan lines
1849 linenum = linenum + 1
1850 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w-_]*)\"") ;
1851 -- Error if incorrect enclosing characters:
1852 if ( i == nil ) then
1853 print ( linenum .. " : " .. line ) ;
1854 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
1855 printconf ( ) ;
1856 os.exit(1) ;
1857 end
1858 if ( cvarname == "opssystem" ) then
1859     -- Verify choice of opssystem:
1860     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1861         opssystem = cvalue
1862     else
1863         cvalueerror ( line, linenum , cvalue )
1864     end
1865 elseif ( cvarname == "latexname" ) then
1866     -- Verify choice of LaTeX compiler:
1867     if (
1868         (cvalue == "pdflatex") or
1869         (cvalue == "xelatex") or
```



```

1870         (cvalue == "lualatex")
1871     ) then
1872         latexname = cvalue
1873     else
1874         cvalueerror ( line, linenum , cvalue )
1875     end
1876 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1877 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1878 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1879 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1880 else
1881 print ( linenum .. " : " .. line ) ;
1882 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
1883 printconf ( ) ;
1884 os.exit(1) ;
1885 end
1886 end -- do scan lines
1887 end -- file exists
1888 -- Select some operating-system commands:
1889 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1890 rmname = "rm"
1891 mvname = "mv"
1892 touchnamepre = "touch"
1893 touchnamepost = ""
1894 dirslash = "/"
1895 opquote= "\""
1896 elseif opsystem=="Windows" then -- For Windows
1897 rmname = "DEL"
1898 mvname = "MOVE"
1899 touchnamepre = "COPY /b"
1900 touchnamepost = "+,,"
1901 dirslash = "\\"
1902 opquote= "\""
1903 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1904 end --- for Windows
1905 end -- loadconf
1906
1907
1908 function refreshdate ( )
1909 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
1910 end
1911
1912
1913 -- Scan the LaTeX log file for the phrase "Rerun to get",
1914 -- indicating that the file should be compiled again.
1915 -- Return true if found.
1916
1917 function reruntoget (filesourse)
1918 io.input(filesourse)
1919 for line in io.lines() do

```

```
1920 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1921 end
1922 return false
1923 end
1924
1925 -- Compile one time, return true if should compile again.
1926 -- fsuffix is "" for print, "_html" for HTML output.
1927
1928 function onetime (fsuffix)
1929 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
1930 err = os.execute(
1931 -- "echo " ..
1932 latexname .. " " .. sourcename..fsuffix )
1933 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1934 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
1935 end
1936
1937 -- Compile up to five times.
1938 -- fsuffix is "" for print, "_html" for HTML output
1939
1940 function manytimes (fsuffix)
1941 if onetime(fsuffix) == true then
1942 if onetime(fsuffix) == true then
1943 if onetime(fsuffix) == true then
1944 if onetime(fsuffix) == true then
1945 if onetime(fsuffix) == true then
1946 end end end end end
1947 end
1948
1949 -- Exit if the given file does not exist.
1950
1951 function verifyfileexists (filename)
1952 if (lfs.attributes ( filename , "modification" ) == nil ) then
1953 print ( "lwarpmk: " .. filename .. " not found." ) ;
1954 os.exit (1) ;
1955 end
1956 end
1957
1958 -- Convert <project>_html.pdf into HTML files:
1959
1960 function pdftohtml ()
1961 -- Convert to text:
1962 print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html"
1963 os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf " .. sou
1964 -- Split the result into individual HTML files:
1965 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
1966 end
1967
1968 -- Remove auxiliary files:
1969
```

```

1970 function removeaux ()
1971     os.execute ( rmname .. " " ..
1972         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
1973         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
1974         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
1975         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
1976         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
1977         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
1978         sourcename .. ".log " .. sourcename .. "_html.log "
1979     )
1980 end
1981
1982
1983
1984 -- Create lateximages based on lateximages.txt:
1985 function createlateximages ()
1986 print ("lwarpmk: Creating lateximages.")
1987 io.input("lateximages.txt")
1988 -- Create the lateximages directory, ignore error if already exists
1989 err = os.execute("mkdir lateximages")
1990 -- Scan lateximages.txt
1991 for line in io.lines() do
1992 -- lwimgpage is the page number in the PDF which has the image
1993 -- lwimgnum is the sequential lateximage number to assign for the image
1994 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
1995 -- For each entry:
1996 if ( i~=nil ) then
1997 -- Separate out the image into its own single-page pdf:
1998 err = os.execute(
1999 "pdfseparate -f " .. lwimgpage .. " -1 " ..
2000 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2001 -- Crop the image:
2002 err = os.execute(
2003 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2004 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2005 -- Convert the image to svg:
2006 err = os.execute(
2007 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2008 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2009 -- Move the result into lateximages/:
2010 err = os.execute(
2011 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2012 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2013 -- Remove the temporary files:
2014 err = os.execute(
2015 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2016 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2017 end
2018 end -- do
2019 end -- function

```

```
2020
2021
2022
2023
2024 -- lwarpmk --version :
2025
2026 if (arg[1] == "--version") then
2027 print ( "lwarpmk: " .. printversion )
2028
2029 else -- not -- version
2030
2031 -- print intro:
2032
2033 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2034
2035 -- lwarpmk print:
2036
2037 if arg[1] == "print" then
2038 loadconf ()
2039 if ( uselatexmk == "true" ) then
2040     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\"\" .. latexname .." %0 %S\" " .. sourcenam
2041     print ("lwarpmk: Done.")
2042 else -- not latexmk
2043     verifyfileexists (sourcename .. ".tex") ;
2044     -- See if up to date:
2045     if (
2046         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2047         (
2048             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2049             lfs.attributes ( sourcename .. ".pdf" , "modification" )
2050         )
2051     ) then
2052         -- Recompile if not yet up to date:
2053         manytimes("")
2054         print ("lwarpmk: Done.") ;
2055     else
2056         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
2057     end
2058 end -- not latexmk
2059
2060 -- lwarp printindex:
2061 -- Compile the index then touch the source
2062 -- to trigger a recompile of the document:
2063
2064 elseif arg[1] == "printindex" then
2065 loadconf ()
2066 print ("lwarpmk: Processing the index.")
2067 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. ".idx")
2068 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2069 refreshdate ()
```

```
2070 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2071 print ("lwarpmk: Done.")
2072
2073 -- lwarpmk html:
2074
2075 elseif arg[1] == "html" then
2076 loadconf ()
2077 if ( uselatexmk == "true" ) then
2078     -- The recorder option is required to detect changes in <project>.tex
2079     -- while we are loading <project>_html.tex.
2080     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2081         .. "-e "
2082         .. opquote .. "$makeindex = q/texindy -M lwarp_html.xdy/" .. opquote
2083         .. " -pdflatex=\"\" .. latexname .. " %0 %S\" "
2084         .. sourcename .. "_html.tex" ) ;
2085     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2086     pdftohtml ()
2087     print ("lwarpmk: Done.")
2088 else -- not latexmk
2089     verifyfileexists ( sourcename .. ".tex" ) ;
2090     -- See if exists and is up to date:
2091     if (
2092         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2093         (
2094             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2095             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2096         )
2097     ) then
2098         -- Recompile if not yet up to date:
2099         manytimes("_html")
2100         pdftohtml ()
2101         print ("lwarpmk: Done.")
2102     else
2103         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2104     end
2105 end -- not latexmk
2106
2107 elseif arg[1] == "pdftohtml" then
2108     loadconf ()
2109     pdftohtml ()
2110
2111 -- lwarpmk htmlindex:
2112 -- Compile the index then touch the source
2113 -- to trigger a recompile of the document:
2114
2115 elseif arg[1] == "htmlindex" then
2116 loadconf ()
2117 print ("lwarpmk: Processing the index.")
2118 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. "_html.idx")
2119 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
```

```
2120 refreshdate ()
2121 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2122 print ("lwarpmk: Done.")
2123
2124 -- lwarpmk limages:
2125 -- Scan the lateximages.txt file to create lateximages,
2126 -- then touch the source to trigger a recompile.
2127
2128 elseif arg[1] == "limages" then
2129 loadconf ()
2130 print ("lwarpmk: Processing images.")
2131 createlateximages ()
2132 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2133 refreshdate ()
2134 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2135 print ("lwarpmk: Done.")
2136
2137 -- lwarpmk again:
2138 -- Touch the source to trigger a recompile.
2139
2140 elseif arg[1] == "again" then
2141 loadconf ()
2142 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2143 refreshdate ()
2144 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2145 print ("lwarpmk: Done.")
2146
2147 -- lwarpmk clean:
2148 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2149
2150 elseif arg[1] == "clean" then
2151 loadconf ()
2152 removeaux ()
2153 print ("lwarpmk: Done.")
2154
2155 -- lwarpmk cleanall
2156 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2157 --      and also project.pdf, *.html
2158
2159 elseif arg[1] == "cleanall" then
2160 loadconf ()
2161 removeaux ()
2162 os.execute ( rmname .. " " ..
2163      sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2164      "*.html"
2165      )
2166 print ("lwarpmk: Done.")
2167
2168 -- lwarpmk with no argument :
2169
```

```
2170 elseif (arg[1] == nil) then
2171 printhelp ()
2172
2173 -- lwarpmk -h or lwarpmk --help :
2174
2175 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2176 printusage ()
2177
2178 else
2179 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2180 printhelp ()
2181 end
2182
2183 end -- not --version
2184 \end{VerbatimOut}
2185 % \end{Verbatim}% for syntax highlighting
2186 \end{LWR@createlwarpmk}
```

Package 3

lwarp-abstract.sty

70 Abstract

(Based on original code by PETER WILSON.)

`Pkg abstract` abstract is supported and patched by lwarp.

abstract is supported. 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}{\BlockClass{abstract}}
4 \AfterEndEnvironment{abstract}{\endBlockClass}
5 }
6
7 \renewcommand{\@bsrunintitle}{%
8 \hspace*{\abstitlekip}%
9 {\abstractnamefont%
10 \InlineClass{abstractrunintitle}{\abstractname}%
11 \@bslabeldelim}%
12 }
13
14 \if@titlepage
15 \renewenvironment{abstract}{%
16 % \titlepage
17 \null\vfil
18 \@beginparpenalty\@lowpenalty
19 \if@bsrunin
20 \else
21 \if@bsstyle
22 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
23 \else
24 \ifnumber@bs
25 \num@bs
26 \else
27 \begin{\absnamepos}%
28 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}

```

```

29         \@endparpenalty\@M
30         \end\absnamepos%
31 %%         \vspace{\abstitleskip}%
32         \fi
33     \fi
34     \vspace{\abstitleskip}%
35 \fi
36 \put@bsintoc%
37 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
38 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
39 }
40 \else
41 \renewenvironment{abstract}{%
42 \if@bsrunin
43 \else
44 \if@bsstyle
45 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
46 \else
47 \ifnumber@bs
48 \num@bs
49 \else
50 \begin{\absnamepos}%
51 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
52 \end\absnamepos%
53 %%         \vspace{\abstitleskip}%
54         \fi
55     \fi
56     \vspace{\abstitleskip}%
57 \fi
58 \put@bsintoc%
59 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
60 {\par\end{@bstr@ctlist}}
61 \fi
62

```

Package 4

lwarp-afterpage.sty

71 Afterpage

Pkg afterpage Not used.

for **HTML output**: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

Package 5

lwarp-algorithmicx.sty

72 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 `span` of the required length, and comments are placed inside a `span` which is floated right.

 package conflicts If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [151.1](#).

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 6

lwarp-alltt.sty

73 Alltt

Pkg alltt alltt is patched for use by lwarp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}

2 \AfterEndPreamble{
3 \AtBeginEnvironment{alltt}{\LWR@atbeginverbatim{alltt}\unskip\vspace*{-\baselineskip}}
4 \AfterEndEnvironment{alltt}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
5 }
```

Package 7

lwarp-amsthm.sty

74 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: span of class amsthmname<theoremstyle>
Theorem Number: span of class amsthmnumber<theoremstyle>
Theorem Note: span of class amsthmnote<theoremstyle>
Proof: div of class amsthmproof
Proof Name: span 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{\exp\@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   \BlockClass{amsthmbody\LWR@thisthmstyle}% new
47   \deferred@thm@head{
48 \the\thm@headfont \thm@indent
49   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% new
50   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% new
51   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% new
52   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
53   \the\thm@headpunct~
54   \thmheadnl % possibly a newline.
55   \hskip\thm@headsep
56 }%

```

```
57 \ignorespaces}
```

Patched for CSS:

```
58 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }
```

Proof QED symbol:

```
59 \AtBeginDocument{
60 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
61 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
62 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
63 }
```

Patched for CSS:

```
64 \renewenvironment{proof}[1][\proofname]{\par
65 \BlockClass{amsthmproof}% new
66 \pushQED{\qed}%
67 \normalfont \topsep6\p@\@plus6\p@\relax
68 \trivlist
69 \item[\hskip\labelsep
70 \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
71 }{%
72 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
73 \endBlockClass% new
74 \@endpefalse
75 }
```

Package 8

lwarp-bookmark.sty

75 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 9

lwarp-booktabs.sty

76 Booktabs

Pkg booktabs booktabs is emulated during HTML output, and the booktabs package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{booktabs}

Booktabs emulation is spread among the tabular code.

Emulated for source compatibility.

2 \newcommand*{\addlinespace}[1]{}

3 \newcommand*{\morecmidrules}{}

4 \newcommand*{\specialrule}[3]{}

Package 10

lwarp-ccaption.sty

77 Ccaption

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

Package 11

lwarp-changepage.sty

78 Changepage

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpag
3 \DeclareRobustCommand{\checkoddpag}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{ }
5 \DeclareRobustCommand{\changepag}[9]{ }
6 \newenvironment{adjustwidth}[2]{ }{ }
7 \newenvironment{adjustwidth*}[2]{ }{ }
```

Package 12

lwarp-cutwin.sty

79 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 13

lwarp-dcolumn.sty

80 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 14

lwarp-draftwatermark.sty

81 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 15

lwarp-ellipsis.sty

82 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 16

lwarp-emptypage.sty

83 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 17

lwarp-endnotes.sty

84 Endnotes

(Based on original code by JOHN LAVAGNINO.)

Pkg endnotes

Discard all options for lwarp-endnotes:

```
for HTML output: 1 \LWR@ProvidesPackagePass{endnotes}

2 \def\noteformat{%
3 % \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5 % \llap{
6 \makeenmark
7 % }
8 }
9
10 \def\@makeenmark{\hbox{\textsuperscript{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

Package 18

lwarp-enumerate.sty

85 Enumerate

Pkg `enumerate` `enumerate` is ignored. `enumitem` is then modified per the `shortlabels` option.

`enumerate` conflicts with `enumitem` if both are loaded at the same time, but `lwarp` does not actually load `enumerate`. While generating HTML, `lwarp` only loads `enumitem`, and `enumerate` is simulated by `enumitem` using the functionality of the `shortlabels` option.

A problem may occur during print output if `enumitem` is loaded, either manually or by some other package such as `siunitx`. If these are used, `enumerate` will conflict with `enumitem` during print output.

for HTML output: Discard all options for `lwarp-enumerate`:

```

1 \LWR@ProvidesPackageDrop{enumerate}

2 % \DeclareOption{shortlabels}
3 % {
4 \def\enit@shl#1{%
5     \ifnum\enit@type=\tw@
6         \enit@toks{#1}%
7     \else
8         \def\enit@c{#1}%
9         \enit@first#1,\@nil\@nil % Returns enit@toks
10    \fi}
11 % }
```


Package 19

lwarp-epigraph.sty

86 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 20

lwarp-eso-pic.sty

87 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 \let\AddToShipoutPicture\AddToShipoutPictureBG
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6] [] {}

```

Package 21

lwarp-everypage.sty

88 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 22

lwarp-extramarks.sty

89 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 23

lwarp-fancyhdr.sty

90 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*{\fancypagestyle}[2] {}
6 \newcommand*{\lhead}[1] {}
7 \newcommand*{\chead}[1] {}
8 \newcommand*{\rhead}[1] {}
9 \newcommand*{\lfoot}[1] {}
10 \newcommand*{\cfoot}[1] {}
11 \newcommand*{\rfoot}[1] {}
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 24

lwarp-float.sty

91 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 52.2 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 {
5 \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
6 }
7 {\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.

```
8 \cslet{listof#1s}\relax
9 \cslet{listof#1es}\relax
10 }
```

`\floatname` $\{\langle type \rangle\} \{\langle name \rangle\}$

Sets the text name of the float, such as “Figure”.

```
11 \NewDocumentCommand{\floatname}{m +m}{%
12 \SetupFloatingEnvironment{#1}{name=#2}%
13 }
```

`\floatplacement` $\{\langle type \rangle\} \{\langle placement \rangle\}$

Float placement is ignored.

```
14 \newcommand*{\floatplacement}[2]{%
15 \SetupFloatingEnvironment{#1}{placement=#2}%
16 }
```

`\floatstyle` $\{\langle style \rangle\}$

Float styles are ignored.

```
17 \newcommand{\floatstyle}[1]{%
18 }
```

`\restylefloat` $* \{\langle style \rangle\}$

Float styles are ignored.

```
19 \NewDocumentCommand{\restylefloat}{s m}{%
20 }
```

Package 25

lwarp-floatflt.sty

92 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 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 % \BlockClass{marginblock}
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 % \endBlockClass
13 \LWR@htmldivclassend{div}
14 }
15
16 \DeclareDocumentEnvironment{floatingfigure}{o m}
17 {\begin{KFLT@marginfloat}{figure}}
18 {\end{KFLT@marginfloat}}
19
20 \DeclareDocumentEnvironment{floatingtable}{o +m}
21 {\begin{KFLT@marginfloat}{table}#2}
22 {\end{KFLT@marginfloat}}
```


Package 26


lwarp-floatrow.sty

93 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 captype} [3 width] [4 height] [5 vert pos]
{6 caption} {7 object}
```

Only parameters for captype, width, caption, and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

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 \begingroup
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 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
73 \booltrue{LWR@infloatrow}
74 \setlength{\linewidth}{6in/#1}
75 }
76 {
77 \boolfalse{LWR@infloatrow}
78 \endBlockClass
79 }
```

Keys for \DeclareNewFloatType:

```
80 \newcommand*{\LWR@frowkeyplacement}{}
81 \newcommand*{\LWR@frowkeyname}{}
82 \newcommand*{\LWR@frowkeyfileext}{}
83 \newcommand*{\LWR@frowkeywithin}{}
84 \newcommand*{\LWR@frowkeycapstyle}{}
85
86 \define@key{frowkeys}{placement}{}%
87 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
88 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
89 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
90 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use \listof{type}{Title} to print a list of the floats.

```
91 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
92 \renewcommand*{\LWR@frowkeyplacement}{}
93 \renewcommand*{\LWR@frowkeyname}{}
94 \renewcommand*{\LWR@frowkeyfileext}{}
95 \renewcommand*{\LWR@frowkeywithin}{}
96 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```

97 \LWR@traceinfo{about to setkeys frowkeys}%
98 \setkeys{frowkeys}{#2}%
99 \LWR@traceinfo{finished setkeys frowkeys}%

```

Create a new float with optional [within]:

```

100 \ifthenelse{\equal{\LWR@frowkeywithin}{}}{
101 {
102 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\
103 \LWR@frowkeyfileext}%
104 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
105 }%
106 {%
107 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\
108 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
109 \newfloat{#1}{\LWR@frowkeyplacement}%
110 {\LWR@frowkeyfileext}[\LWR@frowkeywithin}%
111 \LWR@traceinfo{finished newfloat #1}
112 }%

```

Rename the float if a name was given:

```

113 \ifthenelse{\equal{\LWR@frowkeyname}{}}{
114 {}
115 {\floatname{#1}{\LWR@frowkeyname}}%
116 }

```

Not used:

```

117 \newcommand{\buildFBBBOX}[2]{
118 \newcommand*{\CenterFloatBoxes}{
119 \newcommand*{\TopFloatBoxes}{
120 \newcommand*{\BottomFloatBoxes}{
121 \newcommand*{\PlainFloatBoxes}{
122
123 \newcommand{\capsubrowsettings}{
124
125 \NewDocumentCommand{\RawFloats}{o o}{

```

To be used inside a minipage or parbox.

```

126 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS div of class floatfoot.

```

127 \NewDocumentCommand{\floatfoot}{s +m}{%
128 \begin{BlockClass}{floatfoot}
129 #2
130 \end{BlockClass}

```

```
131 }
```

Used to compute `\linewidth`.

```
132 \newbool{LWR@insubfloatrow}  
133 \boolfalse{LWR@insubfloatrow}
```

```
134 \newenvironment*{subfloatrow}[1][2]  
135 {
```

The row of floats is placed into a `div` of class `floatrow`:

```
136 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
137 \begingroup  
138 \booltrue{LWR@insubfloatrow}  
139 }  
140 {  
141 \endgroup  
142 \endBlockClass  
143 \boolfalse{LWR@insubfloatrow}  
144 }
```

Package 27

lwarp-fontenc.sty

94 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 28

lwarp-fontspec.sty

95 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 29

lwarp-footmisc.sty

96 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 30

lwarp-footnote.sty

97 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}{\LWR@htmltagc{/\LWR@tagregularparagraph}}
```

Removed print-version formatting:

```
10 \def\fn@startfntext{%
11 \setbox\z@\vbox\bgroup%
12 \fn@startnote%
13 \fn@prefntext%
14 \ignorespaces%
15 }
```

Removed print-version formatting, added closing paragraph tag:

```
16 \def\fn@endfntext{%
17 \LWR@htmltagc{/\LWR@tagregularparagraph}%
18 \fn@postfntext%
19 \egroup%
20 \begingroup%
21 \let\@makefntext\@empty%
22 \let\@finalstrut\@gobble%
23 \let\rule\@gobbletwo%
24 \@footnotetext{\unvbox\z@}%
25 \endgroup%
26 }
```

These have been redefined, so re-\let them again:


```
27 \let\endfootnote\fn@endfntext
28 \let\endfootnotetext\endfootnote
```

Package 31

lwarp-footnotehyper.sty

98 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 32

lwarp-framed.sty

99 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
3 \renewenvironment{framed}{%
4 \BlockClass{framed}%
5 }
6 {\endBlockClass}
7
8 \renewenvironment{oframed}{%
9 \BlockClass{framed}%
10 }
11 {\endBlockClass}
12
13
14 \renewenvironment{shaded}{%
15 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
16 \BlockClass{framed}[background: \#\LWR@tempcolor]%
17 }
18 {\endBlockClass}
19
20 \renewenvironment{shaded*}{%
21 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
22 \BlockClass{framed}[background: \#\LWR@tempcolor]%
23 }
24 {\endBlockClass}
25
26
27 \renewenvironment{leftbar}{%
28 \BlockClass{framedleftbar}
29 \def\FrameCommand{}%
30 \MakeFramed {}
31 }%
32 {\endMakeFramed\endBlockClass}
33
34
```

```

35 \renewenvironment{snugshade}{%
36 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
37 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
38 }
39 {\endBlockClass}
40
41 \renewenvironment{snugshade*}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \BlockClass{snugframed}[background: \#\LWR@tempcolor]%
44 }
45 {\endBlockClass}
46
47 \let\oframed\framed
48 \let\endoframed\endframed
49
50
51 \RenewEnviron{titled-frame}[1]{%
52 \CustomFBox{#1}{\Opt}{0pt}{0pt}{0pt}{\BODY}
53 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

54 \renewcommand{\CustomFBox}[7]{%
55 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
56 \begin{BlockClass}{framed}[border: 3px solid \#\LWR@tempcolor]%
57 \ifthenelse{\isempty{#1}}{\}% not empty
58 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
59 \textcolor{TFTitleColor}{\textbf{#1}}%
60 \end{BlockClass}
61 }% not empty
62
63 #7
64
65 \ifthenelse{\isempty{#2}}{\}% not empty
66 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
67 \begin{BlockClass}{framedtitle}[background: \#\LWR@tempcolor]%
68 \textcolor{TFTitleColor}{\textbf{#2}}%
69 \end{BlockClass}
70 }% not empty
71 \end{BlockClass}
72 }

\TitleBarFrame [{<marker>}] {<title>} {<contents>}

73 \renewcommand\TitleBarFrame[3][{}{
74 \CustomFBox
75   {#2}{}%
76   \fboxrule\fboxrule\fboxrule\fboxrule

```

```
77     {#3}%  
78 }  
  
79 \renewcommand{\TF@Title}[1]{#1}  
  
MakeFramed {\langle settings \rangle}  
  
80 \let\MakeFramed\relax  
81 \let\endMakeFramed\relax  
82  
83 \NewEnviron{MakeFramed}[1]{%  
84 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%  
85 }  
  
\fb@put@frame {\langle frame cmd no split \rangle} {\langle frame cmd split \rangle}  
  
86 \renewcommand*\fb@put@frame[2]{%  
87 \relax%  
88 \@tempboxa%  
89 }
```

Package 33

lwarp-ftnright.sty

100 Ftnright

Pkg `ftnright` `ftnright` is ignored.

for **HTML output:** Discard all options for `lwarp-ftnright`:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

Package 34

lwarp-geometry.sty

101 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 35

lwarp-graphics.sty

102 Graphics

Pkg graphics graphics is not used. The user is required to use graphicx instead.

for HTML output: 1 \LWR@loadnever{graphics}{graphicx}

Package 36

lwarp-graphicx.sty

103 Graphicx

Pkg graphicx graphicx is emulated during HTML output, and the graphicx package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 37

lwarp-hyperref.sty

104 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 L^AT_EX 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, `\cleverref` is used instead.

```

50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }

```

For HTML, `\cleverref` 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 38

lwarp-indentfirst.sty

105 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

Package 39

lwarp-inputenc.sty

106 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 40

lwarp-keyfloat.sty

107 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 % \BlockClass{marginblock}
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 % \endBlockClass
25 \LWR@htmldivclassend{div}
26 }

27 }% AtBeginDocument

```

Package 41

lwarp-layout.sty

108 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

Package 42

lwarp-letterspace.sty

109 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] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

Package 43

lwarp-lips.sty

110 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 44

lwarp-listings.sty

111 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}{lol}{#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 reenale line numbers.

```
15 \LWR@traceinfo{About to create verbatim.}%
16 \let\lsthk@EveryPar\relax%
17 \LWR@atbeginverbatim{programlisting}%
18
19 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
20 \else%
```

Inline, so open a ``

```
21 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
22 \fi%
23 }
```

```
24 \renewcommand*\lst@DeInit{%
25 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenable line numbers:

```
26 \let\lsthk@EveryPar\relax%
27
28 \LWR@afterendverbatim%
29 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
30 \else%
```

Inline, so create the closing ``:

```
31 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
32 \fi%
```

Final listings deinit:

```
33 \LWR@origlst@DeInit%
34 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
35 \def\lst@MakeCaption#1{%
36 \LWR@traceinfo{MAKING CAPTION at #1}%
37 \lst@ifdisplaystyle
38 \LWR@traceinfo{making a listings display caption}%
39 \ifx #1t%
40 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
41 \expandafter\refstepcounter
42 \fi {lstlisting}%
43 \LWR@traceinfo{About to assign label: !\lst@label!}%
44 % \ifx\lst@label\@empty\else
45 % \label{\lst@label}\fi
46 \LWR@traceinfo{Finished assigning the label.}%
47 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
48 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
49 \lst@ifnolol\else
50 \ifx\lst@@caption\@empty
```

```

51             \ifx\lst@caption\@empty
52             \ifx\lst@intname\@empty \else \def\lst@temp{ }%
53             \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```

54 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
55 %             \addcontentsline{lol}{lstlisting}{\lst@name}
56             \fi\fi
57             \fi
58             \else

```

This would have to be modified for lwarp:

```

59 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
60             \addcontentsline{lol}{lstlisting}%
61 {\protect\numberline{\thelstlisting}%
62 {\protect\ignorespaces \lst@@caption \protect\relax}}%
63             \fi
64             \fi
65             \fi
66             \ifx\lst@caption\@empty\else
67 \LWR@traceinfo{lst@caption not empty-}%
68             \lst@ifsubstring #1\lst@captionpos
69             {\begingroup
70 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

71 %             \let\@@vskip\vskip
72 %             \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
73 %             \def\lst@vskip{\nobreak\@@vskip\@tempkipa\nobreak}%
74 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
75 %             \ifx #1t\allowbreak \fi
76             \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

77             \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
78             \else

```

New lwarp code to create a title:

```

79 %             \lst@maketitle\lst@title % (AS)
80 \LWR@traceinfo{Making title: \lst@title}%
81 \begin{BlockClass}{lstlistingtitle}% lwarp
82 \lst@maketitle\lst@title% lwarp
83 \end{BlockClass}% lwarp

```



```

84          \fi
85 \LWR@traceinfo{About to assign label: !\lst@label!}%
86          \ifx\lst@label\@empty\else
87 \leavevmode% gets rid of bad space factor error
88 \GetTitleStringExpand{\lst@caption}%
89 \edef\LWR@lntemp{\GetTitleStringResult}%
90 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
91 \label{\lst@label}\fi
92 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

93 %          \ifx #1b\allowbreak \fi
94          \endgroup{}%
95          \fi
96 \LWR@traceinfo{end of making a listings display caption}%
97          \else
98 \LWR@traceinfo{INLINE}%
99          \fi
100 \LWR@traceinfo{DONE WITH CAPTION at #1}%
101 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

102 \lst@Key{numbers}{none}{%
103   \let\lst@PlaceNumber\@empty
104   \lstKV@SwitchCases{#1}%
105   {none&\\%
106     left&\def\lst@PlaceNumber{%
107 % \llap{
108 \LWR@orignormalfont%
109 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
110 % }
111 }
112 \\%
113   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
114     \kern\VerbatimHTMLWidth \kern\lst@numbersep
115     \lst@numberstyle{\thelstnumber}}}%
116   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
117 \end{warpHTML}

```

Package 45

lwarp-longtable.sty

112 Longtable

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{longtable}`

To emulate `longtable`:



For `longtable` `\endhead`, `\endfoot`, and `\endlastfoot` rows, 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.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` * `[<horizontal alignment>] {<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] [] {%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaption}%

```

```

17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaptype}%
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}

\kill is ignored, place a \kill line inside
    \begin{warpprint} ... \end{warpprint} or \warpingprintonly.

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

Package 46

lwarp-lscape.sty

113 Lscape

Pkg lscape lscape is nullified.
 for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

Package 47

lwarp-ltcaption.sty

114 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 48

lwarp-marginfix.sty

115 Marginfix

Pkg marginfix Not used.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\margin skip}[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 49

lwarp-marginnote.sty

116 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 50

lwarp-mcaption.sty

117 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 51


lwarp-mdframed.sty

118 Mdfamed

Pkg **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

for HTML output: `1 \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**.

For title font, use

`frametitlefont=\textbf,`

font instead of

`frametitlefont=\bfseries,`

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

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

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

ignored options **userdefinedwidth** and **align** are currently ignored.

CSS classes Environments created or encapsulated by **mdframed** are enclosed in a **div** of class **md<environmentname>**, or **mdframed** otherwise.

Frame titles are placed into a **span** of class **mdframedtitle**. Subtitles are in a **span** of class **mdframedsubtitle**, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

amsthm must be loaded before **mdframed**

`2 \LWR@origRequirePackage{amsthm}`

Do not require *Tikz* or *pstricks*:

```
3 \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.

```
4 \mdfsetup{
5 startcode={\LWR@mdframedstart\LWR@origraggedright},
6 endcode={\LWR@mdframedend},
7 startinnercode={\LWR@startpars\LWR@origraggedright},
8 endinnercode={\LWR@stoppars},
9 }
```

Given the `mdframed` key, print the color.

```
10 \newcommand*{\LWR@mdfprintcolor}[1]{%
11 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
12 \#\LWR@tempcolor
13 }
```

Given the `mdframed` key, print the length.

```
14 \newcommand*{\LWR@mdfprintlength}[1]{%
15 \rndprintlength{\csuse{mdf@#1@length}}
16 }
```

Actions before an `mdframe` starts.

Encapsulate a frame inside a `div` of the desired `class`.

```
17 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
18 \LWR@stoppars%
```

Below, print HTML pt units:

```
19 \uselengthunit{PT}%
```

Open a `div` and with custom `class` and custom `style`:

```
20 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
21 style=" \LWR@orignewline
```

Convert and print the background color:

```
22 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```


Convert and print the border color and width:

```
23 border: \LWR@mdfprintlength{linewidth} solid
24 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
25 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
26 \ifbool{mdf@shadow}{%
27 box-shadow:
28 \LWR@mdfprintlength{shadowsize}
29 \LWR@mdfprintlength{shadowsize}
30 \LWR@mdfprintlength{shadowsize}
31 \LWR@mdfprintcolor{shadowcolor} ;
32 }
33 {box-shadow: none ;}
34 \LWR@orignewline

35 "}
36 % \LWR@htmldivclass{\LWR@mdthisenv}
```

`mdframed` environment may not work with the modified `\hspace` and `\rule`, so restore them to their originals while inside `mdframed`:

```
37 \let\hspace\LWR@orighspace%
38 \let\rule\LWR@origrule%
39 }
```

Actions after an `mdframe` ends.

After closing the `div`, globally restore to the default environment type:

```
40 \newcommand*{\LWR@mdframedend}{}
```

Close the custom `div`:

```
41 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
42 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
43 \LWR@startpars%
44 }
```

Encapsulation of the original which places the title inside a `span` of class `mdframedtitle`:

```
45 \let\LWR@origmdfframedtitleenv\mdfframedtitleenv
46
47 \newlength{\LWR@titleroundcorner}
48
49 \renewrobustcmd\mdfframedtitleenv[1]{%
50 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
51 \uselengthunit{PT}%
```

Open a `span` with a custom class and custom style:

```
52 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
53 style=" \LWR@orignewline
```

Convert and print the title background color:

```
54 background:
55 \LWR@mdfprintcolor{frametitlebackgroundcolor}
56 ; \LWR@orignewline
```

Convert and print the title rule:

```
57 \ifbool{mdf@frametitlerule}{%
58 border-bottom:
59 \LWR@mdfprintlength{frametitlerulewidth}
60 solid
61 \LWR@mdfprintcolor{frametitlerulecolor}
62 ; \LWR@orignewline
63 }{}%
```

The title's top border radius is adjusted for the line width:

```
64 border-radius:
65 \setlength{\LWR@titleroundcorner}
66 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
67 \rndprintlength{\LWR@titleroundcorner}
68 \rndprintlength{\LWR@titleroundcorner}
69 0pt 0pt
70 \LWR@orignewline
```

Finish the custom style and the opening `span` tag:

```
71 " \LWR@orignewline
72 }% span
```

Restrict paragraph tags inside a span:

```
73 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
74 #1%
```

Close the span and unnest the paragraph tag restriction:

```
75 \LWR@htmltagc{/span}%
```

```
76 \end{LWR@nestspan}%
```

```
77 }
```

```
78 }
```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a span of class mdframedsubtitle:

```
79 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
```

```
80 {% the following empty line is required
```

```
81
```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```
82 \let\par\LWR@origpar%
```

Open a span with a custom class and custom style:

```
83 \LWR@htmltagc{span class="mdframed#1title"
```

```
84 style=" \LWR@orignewline
```

Convert and print the background color:

```
85 background:
```

```
86 \LWR@mdfprintcolor{#1titlebackgroundcolor}
```

```
87 ; \LWR@orignewline
```

Convert and print the above line:

```
88 \ifbool{mdf@#1titleaboveline}{%
```

```
89 border-top:
```

```
90 \LWR@mdfprintlength{#1titleabovelinewidth}
```

```
91 solid
```

```
92 \LWR@mdfprintcolor{#1titleabovelinecolor}
```

```
93 ; \LWR@orignewline
```

```
94 }{}%
```

Convert and print the below line:

```

95 \ifbool{mdf@#1titlebelowline}{%
96 border-bottom:
97 \LWR@mdfprintlength{#1titlebelowlinewidth}
98 solid
99 \LWR@mdfprintcolor{#1titlebelowlinecolor}
100 ; \LWR@orignewline
101 }{}%
```

Finish the custom style and the opening span tag:

```

102 "% span
```

Restrict paragraph tags inside a span:

```

103 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```

104 \IfNoValueTF{#2}
105 {\csuse{LWR@origmdf#1title}{#3}}%
106 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```

107 \LWR@htmltagc{/span}% the following empty line is required
108 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
109
110 }
```

```

111 \let\LWR@origmdfsubtitle\mdfsubtitle
112
113 \newcommand*{\LWR@mdfsubtitle}{%
114 \LWR@mdfsubtitlecommon{sub}%
115 }
116 \let\mdfsubtitle\LWR@mdfsubtitle
```

```

117 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
118
119 \newcommand*{\LWR@mdfsubsubtitle}{%
120 \LWR@mdfsubsubtitlecommon{subsub}%
121 }
122 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

Stores the environment of the frame about to be created:

```

123 \newcommand*{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

124 \renewrobustcmd*\newmdenv[2] [] {%
125 \newenvironment{#2}%
126 {%
127 \mdfsetup{#1}%
128 \renewcommand*\LWR@mdthisenv}{md#2}%
129 \begin{mdframed}%
130 }
131 {\end{mdframed}}%
132 }

```

Modified from the original to remember the environment.

```

133 \renewrobustcmd*\surroundwithmdframed[2] [] {%
134 \BeforeBeginEnvironment{#2}{%
135 \renewcommand*\LWR@mdthisenv}{md#2}%
136 \begin{mdframed}[#1]}%
137 \AfterEndEnvironment{#2}{\end{mdframed}}%
138 }

```

$[\langle \textit{numberedlike} \rangle]$ $\{\langle \textit{caption} \rangle\}$ $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

139 \let\LWR@origmdtheorem\mdtheorem
140
141 \DeclareDocumentCommand{\LWR@mdtheorem}{0}{ m o m o }{%
142 \LWR@origmdtheorem[#1]{#2}[#3]{#4}[#5}%
143 \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}%
144 }
145
146 \let\mdtheorem\LWR@mdtheorem

```

$[\langle \textit{numberedlike} \rangle]$ $\{\langle \textit{caption} \rangle\}$ $[\langle \textit{within} \rangle]$

Modified from the original to remember the environment.

```

147 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
148 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
149   {\newtheorem{#2}{#4}}{%
150     \IfValueTF{#3}{\newtheorem{#2}[#3]{#4}}{%
151       \IfValueTF{#5}{\newtheorem{#2}{#4}[#5]}{%
152         }%
153     }%
154   \BeforeBeginEnvironment{#2}{%
155     \renewcommand*\LWR@mdthisenv}{md#2}%
156     \begin{mdframed}[#1]}%
157   \AfterEndEnvironment{#2}{%
158     \end{mdframed}}%
159 }

```

Package 52

lwarp-microtype.sty

119 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 \renewcommand*\DeclareMicrotypeSet[3][]{\{ }
3 \renewcommand*\UseMicrotypeSet[2][]{\{ }
4 \renewcommand*\DeclareMicrotypeSetDefault[2][]{\{ }
5 \renewcommand*\SetProtrusion[3][]{\{ }
6 \renewcommand*\SetExpansion[3][]{\{ }
7 \renewcommand*\SetTracking[3][]{\{ }
8 \renewcommand*\SetExtraKerning[3][]{\{ }
9 \renewcommand*\SetExtraSpacing[3][]{\{ }
10 \renewcommand*\DisableLigatures[2][]{\{ }
11 \renewcommand*\DeclareCharacterInheritance[3][]{\{ }
12 \renewcommand*\DeclareMicrotypeVariants[1]{\{ }
13 \renewcommand*\DeclareMicrotypeAlias[2]{\{ }
14 \renewcommand*\LoadMicrotypeFile[1]{\{ }
15 \renewcommand*\DeclareMicrotypeBabelHook[2]{\{ }
16 \renewcommand*\microtypesetup[1]{\{ }
17 \renewcommand*\microtypecontext[1]{\{ }
18 \renewcommand*\textmicrotypecontext[2]{\{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{\%
20 \renewcommand*\lsstyle{\{ }
21 \renewcommand*\textls[2][]{\{ }
22 \def\textls#1#\{ }
23 \renewcommand*\lslig[1]{\{#1}
24 }
25 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
26 \def\DeclareMicrotypeVariants#1#{\@gobble}
27 \@onlypreamble\DeclareMicrotypeSet
28 \@onlypreamble\UseMicrotypeSet
29 \@onlypreamble\DeclareMicrotypeSetDefault
30 \@onlypreamble\DisableLigatures
31 \@onlypreamble\DeclareMicrotypeVariants
32 \@onlypreamble\DeclareMicrotypeBabelHook

```

Package 53

lwarp-mparhack.sty

120 Mparhack

Pkg mparhack Not used.

for **HTML output:** Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

Package 54

lwarp-multicol.sty

121 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 {\BlockClass{multicols}
```

Optional HTML div class for the heading:

```
5 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the div:

```
6 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
7 \newcommand*{\columnbreak}{}
8 \newcommand*{\RLmulticolcolumns}{}
9 \newcommand*{\LRmulticolcolumns}{}
10
11 \newlength{\premulticols}
12 \newlength{\postmulticols}
13 \newlength{\multicolsep}
14 \newlength{\multicolbaselineskip}
15 \newlength{\multicoltolerance}
16 \newlength{\multicolpretolerance}
17 \newcommand*{\columnseprulecolor}{\normalcolor}
```

```
18 \newcounter{columnbadness}
19 \newcounter{finalcolumnbadness}
20 \newcounter{collectmore}
21 \newcounter{unbalance}
22 \newlength{\multicolovershoot}
23 \newlength{\multicolundershoot}

24 \end{warpHTML}
```

Package 55

lwarp-multirow.sty

122 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

Package 56

lwarp-nameref.sty

123 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 57

lwarp-needspace.sty

124 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 58

lwarp-newclude.sty

125 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 59

lwarp-newunicodechar.sty

126 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 60

lwarp-nextpage.sty

127 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 61

lwarp-nowidow.sty

128 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 62

lwarp-ntheorem.sty

129 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: `span` 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 L^AT_EX font settings are ignored.

⚠ **Equation numbering** `ntheorem` has a bug with equation numbering in 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\csname\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@thisthmstyle{\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@thisthmstyle{\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@addtheoremline#1\endcsname{%
77       \noexpand\@num@addtheoremline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
79       \noexpand\@nonum@addtheoremline{#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@thisthmstyle{\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       \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
102       \LWR@inctheorem% new
103     }#2}%
104     \def\@opargbegintheorem####1####2####3{%
105       \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
106       \LWR@inctheorem% new
107     }#3}%
108   }%
109 }%
110 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
111 }
112
113 \renewtheoremstyle{plain}%
114   {\item[\hskip\labelsep \theorem@headerfont
115 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
116   {\item[\hskip\labelsep \theorem@headerfont
117 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
118
119 \renewtheoremstyle{break}%
120   {\item[
121 % \rlap{\vbox{\hbox{
122 \hskip\labelsep \theorem@headerfont
123 \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
124 % }\hbox{\strut}}}}
125 ]}%
126   {\item[
127 % \rlap{\vbox{\hbox{
128 \hskip\labelsep \theorem@headerfont
129 \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline
130 % }\hbox{\strut}}}}
131 ]}

```

```

132
133 \renewtheoremstyle{change}%
134   {\item[\hskip\labelsep
135 \theorem@headerfont
136 \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
137   {\item[\hskip\labelsep
138 \theorem@headerfont
139 \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
140
141 \renewtheoremstyle{changebreak}%
142   {\item[
143 % \rlap{\vbox{\hbox{
144 \hskip\labelsep \theorem@headerfont
145 \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
146 % }\hbox{\strut}}}]
147 ]}%
148   {\item[
149 % \rlap{\vbox{\hbox{
150 \hskip\labelsep \theorem@headerfont
151 \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
152 % }\hbox{\strut}}}]
153 ]}
154
155 \renewtheoremstyle{margin}%
156   {\item[\hskip\labelsep\theorem@headerfont
157 \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
158 ]}%
159   {\item[\hskip\labelsep\theorem@headerfont
160 \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
161 ]}
162
163 \renewtheoremstyle{marginbreak}%
164   {\item[\hskip\labelsep\theorem@headerfont
165 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
166 ]}%
167   {\item[\hskip\labelsep\theorem@headerfont
168 \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
169 ]}
170
171 \renewtheoremstyle{nonumberplain}%
172   {\item[\theorem@headerfont\hskip\labelsep
173 \InlineClass{theoremheaderplain}{##1\theorem@separator}]]}%
174   {\item[\theorem@headerfont\hskip\labelsep
175 \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]]}
176
177 \renewtheoremstyle{nonumberbreak}%
178   {\item[
179 % \rlap{\vbox{\hbox{
180 \hskip\labelsep \theorem@headerfont
181 \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline

```

```

182 % }\hbox{\strut}}
183 ]}%
184 {\item[
185 % \rlap{\vbox{\hbox{
186 \hskip\labelsep \theorem@headerfont
187 \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
188 % }\hbox{\strut}}
189 ]}
190
191 \renewtheoremstyle{empty}%
192 {\item[]}%
193 {\item[\theorem@headerfont \hskip\labelsep\relax
194 \InlineClass{theoremheaderplain}{##3}]}
195
196 \renewtheoremstyle{emptybreak}%
197 {\item[]}%
198 {\item[\theorem@headerfont \hskip\labelsep\relax
199 \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```

200 \ifbool{LWR@theoremamsthm}{-}{-}
201 % upright text via CSS
202 \newtheoremstyle{plainupright}%
203 {\item[\hskip\labelsep \theorem@headerfont
204 \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
205 {\item[\hskip\labelsep \theorem@headerfont
206 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
207
208 % upright text and small caps header via CSS
209 \newtheoremstyle{nonumberplainuprightsc}%
210 {\item[\theorem@headerfont\hskip\labelsep
211 \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
212 {\item[\theorem@headerfont\hskip \labelsep
213 \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new CSS:

```

214 \theoremstyle{plainupright}
215 \theorembodyfont{\upshape}
216 \theoremsymbol{\ensuremath{\_ \Box}}
217 \renewtheorem{Example}{Example}
218 \renewtheorem{example}{Example}
219 \renewtheorem{Beispiel}{Beispiel}
220 \renewtheorem{beispiel}{Beispiel}
221 \renewtheorem{Bemerkung}{Bemerkung}
222 \renewtheorem{bemerkung}{Bemerkung}
223 \renewtheorem{Anmerkung}{Anmerkung}

```

```

224 \renewtheorem{anmerkung}{Anmerkung}
225 \renewtheorem{Remark}{Remark}
226 \renewtheorem{remark}{Remark}
227 \renewtheorem{Definition}{Definition}
228 \renewtheorem{definition}{Definition}
229
230 \theoremstyle{nonumberplainuprightsc}
231 \theoremsymbol{\ensuremath{\_ \blacksquare}}
232 \renewtheorem{Proof}{Proof}
233 \renewtheorem{proof}{Proof}
234 \renewtheorem{Beweis}{Beweis}
235 \renewtheorem{beweis}{Beweis}
236 \qedsymbol{\ensuremath{\_ \blacksquare}}
237
238 \theoremsymbol{}
239 }% not amsthm

```

Only if the amsthm option was given:

```

240 \ifbool{LWR@theoremamsthm}{
241
242 \gdef\th@plain{%
243   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
244   \def\@begintheorem##1##2{%
245     \BlockClass{theorembodyplain}% new
246     \LWR@intheorem% new
247     \item[\hskip\labelsep
248 % \theorem@headerfont
249     \InlineClass{theoremheaderplain}{##1\ ##2.}
250 ]}%
251   \def\@opargbegintheorem##1##2##3{%
252     \BlockClass{theorembodyplain}% new
253     \LWR@intheorem% new
254     \item[\hskip\labelsep
255 % \theorem@headerfont
256     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
257 ]}}
258
259 \gdef\th@nonumberplain{%
260   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
261   \def\@begintheorem##1##2{%
262     \BlockClass{theorembodyplain}% new
263     \LWR@intheorem% new
264     \item[\hskip\labelsep
265 % \theorem@headerfont
266     \InlineClass{theoremheaderplain}{##1.}
267 ]}%
268   \def\@opargbegintheorem##1##2##3{%
269     \BlockClass{theorembodyplain}% new
270     \LWR@intheorem% new

```

```

271 \item[\hskip\labelsep
272 % \theorem@headerfont
273 \InlineClass{theoremheaderplain}{##1\ (##3).}
274 ]}]
275
276 \gdef\th@definition{%
277 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
278 \def\@begintheorem##1##2{%
279 \BlockClass{theorembodydefinition}% new
280 \LWR@intheorem% new
281 \item[\hskip\labelsep
282 % \theorem@headerfont
283 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
284 ]}%
285 \def\@opargbegintheorem##1##2##3{%
286 \BlockClass{theorembodydefinition}% new
287 \LWR@intheorem% new
288 \item[\hskip\labelsep
289 % \theorem@headerfont
290 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
291 ]}]
292
293 \gdef\th@nonumberdefinition{%
294 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
295 \def\@begintheorem##1##2{%
296 \BlockClass{theorembodydefinition}% new
297 \LWR@intheorem% new
298 \item[\hskip\labelsep
299 % \theorem@headerfont
300 \InlineClass{theoremheaderdefinition}{##1.}
301 ]}%
302 \def\@opargbegintheorem##1##2##3{%
303 \BlockClass{theorembodydefinition}% new
304 \LWR@intheorem% new
305 \item[\hskip\labelsep
306 % \theorem@headerfont
307 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
308 ]}]
309
310 \gdef\th@remark{%
311 \def\theorem@headerfont{\itshape}\normalfont%
312 \def\@begintheorem##1##2{%
313 \BlockClass{theorembodyremark}% new
314 \LWR@intheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderremark}{##1\ ##2.}
318 ]}%
319 \def\@opargbegintheorem##1##2##3{%
320 \BlockClass{theorembodyremark}% new

```

```

321 \LWR@inctheorem% new
322     \item[\hskip\labelsep
323 % \theorem@headerfont
324 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
325 ]}}
326
327 \gdef\th@nonumberremark{%
328 \def\theorem@headerfont{\itshape}\normalfont%
329 \def\@begintheorem##1##2{%
330 \BlockClass{theorembodyremark}% new
331 \LWR@inctheorem% new
332     \item[\hskip\labelsep
333 % \theorem@headerfont
334 \InlineClass{theoremheaderremark}{##1.}
335 ]}%
336 \def\@opargbegintheorem##1##2##3{%
337 \BlockClass{theorembodyremark}% new
338 \LWR@inctheorem% new
339     \item[\hskip\labelsep
340 % \theorem@headerfont
341 \InlineClass{theoremheaderremark}{##1\ (##3).}
342 ]}}
343
344 \gdef\th@proof{%
345 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
346 \def\@begintheorem##1##2{%
347 \BlockClass{theorembodyproof}% new
348 \LWR@inctheorem% new
349     \item[\hskip\labelsep
350 % \theorem@headerfont
351 \InlineClass{theoremheaderproof}{##1.}
352 ]}%
353 \def\@opargbegintheorem##1##2##3{%
354 \BlockClass{theorembodyproof}% new
355 \LWR@inctheorem% new
356     \item[\hskip\labelsep
357 % \theorem@headerfont
358 \InlineClass{theoremheaderproof}{##1\ (##3).}
359 ]}}
360
361
362
363 \newcounter{proof}%
364 \if@thmmarks
365 \newcounter{currproofctr}%
366 \newcounter{endproofctr}%
367 \fi
368
369 \gdef\proofSymbol{\openbox}
370

```

```

371 \newcommand{\proofname}{Proof}
372
373 \newenvironment{proof}[1][\proofname]{
374 \th@proof
375 \def\theorem@headerfont{\itshape}%
376 \normalfont
377 \theoremsymbol{\ensuremath{\_\blacksquare}}
378 \@thm{proof}{proof}{#1}
379 }%
380 {\@endtheorem}
381
382 }{}% amsthm option

```

Patched for CSS:

```

383 \let\LWR@origendtheorem\@endtheorem
384 \renewcommand{\@endtheorem}{%
385 \ifbool{LWR@theoremmarks}{%
386 \ifsetendmark%
387 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
388 \setendmarkfalse%
389 \fi%
390 }{}%
391 \LWR@origendtheorem%
392 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
393 \endBlockClass%
394 }

395 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {\langle 1: printed type\rangle} {\langle 2: \#\rangle} {\langle 3: optional\rangle} {\langle 4: page\rangle}

396 \renewcommand{\thm@@thmline@noname}[4]{%
397 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
398 }
399
400 \renewcommand{\thm@@thmline@name}[4]{%
401 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
402 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

403 \def\thm@@thmline{\thm@@thmline@name}

```

Patch for CSS:

```

404 \def\listtheorems#1{

```

```

405 \LWR@htmlelementclass{nav}{lothm}%
406 \begingroup
407 \c@tocdepth=-2%
408 \def\thm@list{#1}\thm@processlist
409 \endgroup
410 \LWR@htmlelementclassend{nav}{lothm}%
411 }

```

Proof QED symbol:

```

412
413 \newcommand{\qed}{\quad\the\qedsymbol}
414
415 \AtBeginDocument{
416 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
417 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
418 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
419 }

```

`\thref {<label>}`

```

420 \newcommand*{\thref}[1]{\cref{#1}}

```


Package 63

lwarp-pagenote.sty

130 Pagenote

Pkg pagenote pagenote works as-is.

It is only included as an `lwarp-pagenote.sty` file because past versions of `lwarp` used `pagenote` to emulate footnotes, and so the file may exist on current installations, and should be over-written by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{pagenote}

Package 64

lwarp-parskip.sty

131 Parskip

Pkg `parskip` `parskip` is ignored.

for **HTML output:** Discard all options for `lwarp-parskip`.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

Package 65

lwarp-placeins.sty

132 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 66

lwarp-ragged2e.sty

133 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 67

lwarp-rotating.sty

134 Rotating

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

This code is untested.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotating}

2 \let\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \let\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 68

lwarp-setspace.sty

135 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 {\BlockClass{singlespace}}
11 {\endBlockClass}
12
13 \newenvironment*{singlespace*}
14 {\BlockClass{singlespace}}
15 {\endBlockClass}
16
17 \newenvironment*{spacing}[1]{
18
19 }{
20
21 }
22
23 \newenvironment*{onehalfspace}
24 {\BlockClass{onehalfspace}}
25 {\endBlockClass}
26
27 \newenvironment*{doublespace}
28 {\BlockClass{doublespace}}
29 {\endBlockClass}

clearpage

```

Package 69

lwarp-showidx.sty

136 Showidx

Pkg `showidx` `showidx` is ignored.

for **HTML output**: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 70

lwarp-showkeys.sty

137 Showkeys

Pkg `showkeys` `showkeys` is ignored.

for **HTML output**: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```

Package 71

lwarp-sidecap.sty

138 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 72

lwarp-sidenotes.sty

139 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 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 % \BlockClass{marginblock}
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}}{}
```

```

28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 % \endBlockClass
33 \LWR@htmldivclassend{div}
34 }
35
36 \RenewDocumentEnvironment{marginfigure}{o}
37   {\begin{KFLT@marginfloat}{figure}}
38   {\end{KFLT@marginfloat}}
39
40 \RenewDocumentEnvironment{margintable}{o}
41   {\begin{KFLT@marginfloat}{table}}
42   {\end{KFLT@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the L^AT_EX 2_ε `article.cls` source:

```

43 \renewenvironment{figure*}
44     {\@dblfloat{figure}}
45     {\end@dblfloat}
46
47 \renewenvironment{table*}
48     {\@dblfloat{table}}
49     {\end@dblfloat}

```


Package 73

lwarp-subfig.sty

140 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}

\sfc@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}

The outer minipage allows side-by-side subfloats with \hfill between.

2 \long\def\sfc@@@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         \@tempskipb\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 width0pt height0pt depth0pt
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 width0pt height0pt depth0pt
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 {\langle 1 type\rangle} {\langle 2 lof entry\rangle} {\langle 3 caption\rangle}

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\@captive\space
209     \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@captive}%
211     \@nameuse{thesub\@captive}.)}}

```

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 \let\LWR@orig@newsfloat\@newsfloat
217
218 \def\@newsfloat[#1]#2{%
219 \LWR@orig@newsfloat[#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 74

lwarp-tabularx.sty

141 Tabularx

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for **HTML output**: Discard all options for `lwarp-tabularx`:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \newenvironment{tabularx}[2]
3 {\tabular{#2}}
4 {\endtabular}
5
6 \newenvironment{tabularx*}[2]
7 {\tabular{#2}}
8 {\endtabular}
```


Package 75

lwarp-textpos.sty

142 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 76

lwarp-theorem.sty

143 Theorem

(Based on original code by FRANK MITTELBAACH.)

Pkg theorem theorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: div of class theorembody<theoremstyle>

Theorem Header: span of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 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     \BlockClass{theorembody\LWR@thisthmstyle}% new
65       \item[\hskip\labelsep
66 \InlineClass{theoremheader}{##1\ ##2}
67 ]}%
68 \def\@opargbegintheorem##1##2##3{%
69 \BlockClass{theorembody\LWR@thisthmstyle}% new
70   \item[\hskip\labelsep
71 \InlineClass{theoremheader}{##1\ ##2\ (##3)}
72 ]}
73 }
74
75 \gdef\th@break{%\normalfont\slshape
76   \def\@begintheorem##1##2{%
77 \BlockClass{theorembody\LWR@thisthmstyle}% new
78 \item[\hskip \labelsep
79 \InlineClass{theoremheader}{##1\ ##2}\newline%
80 ]}%
81 \def\@opargbegintheorem##1##2##3{%
82 \BlockClass{theorembody\LWR@thisthmstyle}% new
83   \item[\hskip \labelsep
84 \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
85 ]}
86 }
87
88 \gdef\th@marginbreak{%\normalfont\slshape
89   \def\@begintheorem##1##2{
90 \BlockClass{theorembody\LWR@thisthmstyle}% new
91 \item[\hskip\labelsep %
92 \InlineClass{theoremheader}{##2 \quad ##1}\newline
93 ]}%
94 \def\@opargbegintheorem##1##2##3{%
95 \BlockClass{theorembody\LWR@thisthmstyle}% new
96 \item[\hskip\labelsep %
97 \InlineClass{theoremheader}{##2 \quad ##1\ %
98 (##3)}\newline
99 ]}
100 }
101
102 \gdef\th@changebreak{%\normalfont\slshape
103   \def\@begintheorem##1##2{
104 \BlockClass{theorembody\LWR@thisthmstyle}% new
105 \item[\hskip\labelsep
106 \InlineClass{theoremheader}{##2\ ##1}\newline
107 ]}%

```

```

108 \def\@opargbegintheorem##1##2##3{%
109 \BlockClass{theorembody\LWR@thisthmstyle}% new
110 \item[\hskip\labelsep
111 \InlineClass{theoremheader}{ ##2\ ##1\ %
112 (##3)}\newline
113 ]}
114 }
115
116 \gdef\th@change{%\normalfont\slshape
117 \def\@begintheorem##1##2{
118 \BlockClass{theorembody\LWR@thisthmstyle}% new
119 \item[\hskip\labelsep
120 \InlineClass{theoremheader}{##2\ ##1}
121 ]}%
122 \def\@opargbegintheorem##1##2##3{%
123 \BlockClass{theorembody\LWR@thisthmstyle}% new
124 \item[\hskip\labelsep
125 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
126 ]}
127 }
128
129 \gdef\th@margin{%\normalfont\slshape
130 \def\@begintheorem##1##2{
131 \BlockClass{theorembody\LWR@thisthmstyle}% new
132 \item[\hskip\labelsep
133 \InlineClass{theoremheader}{##2 \quad ##1}
134 ]}%
135 \def\@opargbegintheorem##1##2##3{%
136 \BlockClass{theorembody\LWR@thisthmstyle}% new
137 \item[\hskip\labelsep
138 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
139 ]}
140 }

```

Patched for CSS:

```

141 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

Package 77

lwarp-threeparttable.sty

144 Threeparttable

Pkg `threeparttable` `threeparttable` is emulated during HTML output, and the `threeparttable` package is ignored.

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 \BlockClass{tnotes}%
7 \setlist[description]{format=\LWR@printtablenote}%
8 \description%
9 }
10 {%
11 \enddescription%
12 \endBlockClass%
13 }
```

```
14 \newcommand{\tnote}[1]{\textsuperscript{#1}}
```

Package 78

lwarp-tikz.sty

145 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.

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 79

lwarp-titles.sty

146 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 \RenewDocumentCommand{\newpagestyle}{m o m}{}
3 \RenewDocumentCommand{\renewpagestyle}{m o m}{}

4 \RenewDocumentCommand{\sethead}{o o o m m m}{}
5 \RenewDocumentCommand{\setfoot}{o o o m m m}{}

6 \RenewDocumentCommand{\settitledmarks}{s m}{}

7 \renewcommand*{\headrule}{}
8 \renewcommand*{\footrule}{}

9 \renewcommand*{\setheadrule}[1]{}
10 \renewcommand*{\setfootrule}[1]{}

11 \newcommand*{\makeheadrule}{}
12 \newcommand*{\makefootrule}{}

13 \renewcommand{\setmarkboth}[1]{}

14 \RenewDocumentCommand{\widenhead}{s o o m m}{}

15 \renewcommand*{\bottitledmarks}{}
16 \renewcommand*{\toptitledmarks}{}
17 \renewcommand*{\firsttitledmarks}{}
18 \renewcommand*{\nexttoptitledmarks}{}
19 \renewcommand*{\outertitledmarks}{}
20 \renewcommand*{\innertitledmarks}{}

21 \RenewDocumentCommand{\newtitledmark}{s m}{}

```

```
22 \RenewDocumentCommand{\pretitlemark}{s m m}{}

23 \renewcommand{\ifsamemark}[4]{}

24 \NewDocumentCommand{\setfloathead}{s o o m m m m}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o m m m m}{}

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o 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{\nexttoextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

Package 80

lwarp-titlesec.sty

147 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 81

lwarp-titletoc.sty

148 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 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 82

lwarp-titling.sty

149 Titling

`Pkg titling` titling is used by lwarp. The following patches are not needed by lwarp, but are required if the user requests titling.

lwarp uses page notes for footnotes, so 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 83

lwarp-tocloft.sty

150 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*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

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{\cftpartincent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpfont}{}
31 \newcommand*{\cftpapresnum}{}
32 \newcommand*{\cftpataftersnum}{}
33 \newcommand*{\cftpataftersnumb}{}
34 \newcommand*{\cftpleader}{}
35 \newcommand*{\cftpdotsep}{1}
36 \newcommand*{\cftpfont}{}
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*{\cftchapfont}{}
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*{\cftsecfont}{}
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*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
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*{\cftsubparaftersnum}{}
99 \newcommand*{\cftsubparaftersnumb}{}
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}
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 84

lwarp-trivfloat.sty

151 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{\exemplename}{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}`

151.1 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{tflfloatcnt}{1} % start trivfloats with .lob
\makeatletter
```

```
9 \end{warpall}
```

Package 85

lwarp-verse.sty

152 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` 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{
```

At the beginning of the `verse` environment:

```

3 \AtBeginEnvironment{verse}
4 {%

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
5 \ifdef{\vleftskip}{%
6 \setlength{\vleftskip}{\HTMLvleftskip}
7 \setlength{\leftmargini}{\HTMLleftmargini}
8 }{}
9 \LWR@atbeginverbatim{verse}
10 \unskip\vspace{-\baselineskip}
11 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

12 \AfterEndEnvironment{verse}{
13 \unskip\vspace{-\baselineskip}
14 \LWR@afterendverbatim
15 }
```

Patch to place poemtitle inside an HTML span of class `poemtitle`:

```

16 \ifdef{\poemtitle}{
17 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
18   \vspace{\beforepoemtitleskip}%
19   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
20   \vspace{\afterpoemtitleskip}%
21 }
22 }{}
23
24 }
```

Package 86

lwarp-wallpaper.sty

153 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}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}
```

Package 87

lwarp-wrapfig.sty

154 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
4 \newcommand*{\LWR@wrapposition}{}
5
6 \newcommand*{\LWR@subwrapfigure}[2]{%
7 \LWR@maybeinthisfloat%
8 \renewcommand*{\LWR@wrapposition}{}%
9 \ifthenelse{%
10 \equal{#1}{r}\OR\equal{#1}{R}\OR%
11 \equal{#1}{o}\OR\equal{#1}{O}%
12 }{%
13 {\renewcommand*{\LWR@wrapposition}{float:right}}%
14 {\renewcommand*{\LWR@wrapposition}{float:left}}%
15 \setlength{\LWR@wrapwidth}{#2}%
16 \addtolength{\LWR@wrapwidth}{4em}%
17 \uselengthunit{PT}%
18 % \BlockClass{marginblock}{%
19 % width:\rndprintlength{\LWR@wrapwidth} ; %
20 % \LWR@wrapposition%
21 % ]%
22 \LWR@stoppars%
23 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
24 style="width:\rndprintlength{\LWR@wrapwidth} ; %
25 \LWR@wrapposition"%
26 }
27 \LWR@startpars
28 }
29
30
31 \NewDocumentEnvironment{wrapfigure}{o m o m}
32 {%
33 \LWR@subwrapfigure{#2}{#4}%
34 \captionsetup{type=figure}%
```

```
35 }
36 {
37 \LWR@htmldivclassend{div}
38 }
39
40
41 \NewDocumentEnvironment{wraptable}{o m o m}
42 {%
43 \LWR@subwrapfigure{#2}{#4}%
44 \captionsetup{type=table}%
45 }
46 {
47 \LWR@htmldivclassend{div}
48 }
49
50
51 \NewDocumentEnvironment{wrapfloat}{m o m o m}
52 {%
53 \LWR@subwrapfigure{#3}{#5}%
54 \captionsetup{type=#1}%
55 }
56 {
57 \LWR@htmldivclassend{div}
58 }
59
60 \newlength{\wrapoverhang}
```


Package 88

lwarp-xcolor.sty

155 Xcolor

Pkg `xcolor` `xcolor` is supported by `lwarp`.

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.

for HTML output: `1 \LWR@ProvidesPackagePass{xcolor}`

`2 \newcommand*{\LWR@tempcolor}{}%`

defaulting to black.

`3 \newcommand*{\LWR@currenttextcolor}{black}`

Creates a styled span with a color converted to HTML hex colorspace. Uses `LWR@spandepth` to prevent paragraph tags inside the span. If used for `\textcolor`, with a styletext of `color:`, then the new color is copied into `\LWR@currenttextcolor` for possible re-use in `\rule`.

`4 \NewDocumentCommand{\LWR@colorstyle}{m m m m}{%`

Use the `xcolor` package to convert to an HTML color space:

`5 \convertcolorspec{#2}{#3}{HTML}\LWR@tempcolor%`

If is a `\textcolor`, save a copy of this color for use by `\rule`:

`6 \ifthenelse{\equal{#1}{color:}}{%`

`7 {\renewcommand*{\LWR@currenttextcolor}{\#\LWR@tempcolor}}{}}%`

Create the HTML span with the styled color:

`8 \LWR@htmltagc{span style="#1\#\LWR@tempcolor"}{}}%`

`9 \begin{\LWR@nestspan}%`

Prevent additional paragraph tags inside this span:

Print the contents then close the span:

```
10 #4%
11 \LWR@htmltagc{/span}%
12 \end{LWR@nestspan}%
```

For paragraph-tag handling:

```
13 \LWR@ensuredoingapar%
14 }
```

`\color` appears in the L^AT_EX 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*} is converted into an HTML hex color span.

```
15 \NewDocumentCommand{\LWR@textcolor}{0{named} m m}{%
16 \begingroup%
17 \LWR@colorstyle{color:}{#1}{#2}{#3}%
18 \endgroup%
19 }
```

`\pagecolor` [*model*] {*color*} is ignored. Use `\NewCSS` instead.

```
20 \newcommand*{\LWR@pagecolor}[2][named]{}
```

`\colorbox` [*model*] {*color*} {*text*} is converted into an HTML hex background color span.

```
21 \NewDocumentCommand{\LWR@colorbox}{0{named} m m}{%
22 \begingroup%
23 \LWR@colorstyle{background:}{#1}{#2}{#3}%
24 \endgroup%
25 }
```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*} is converted into a framed HTML hex background color span.

A background color of "none" creates a colored frame without a background color.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{0{named} m 0{named} m m}{%
```

```
27 \begingroup%
28 \ifthenelse{\equal{#4}{none}}{% no background color
29 \LWR@colorstyle{border:1px solid }{#1}{#2}{#5}%
30 }{% yes background color
31 \LWR@colorstyle{border:1px solid }{#1}{#2}%
32 {\LWR@colorstyle{background:}{#3}{#4}{#5}}%
33 }%
34 \endgroup%
35 }
```

Redirect to new definitions:

```
36 \let\textcolor\LWR@textcolor
37 \let\pagecolor\LWR@pagecolor
38 \let\colorbox\LWR@colorbox
39 \let\fcolorbox\LWR@fcolorbox
```

Package 89

lwarp-xfrac.sty

156 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 (lrm) 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}

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 \let\scalebox\LWR@noscalebox%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
21 denominator-format = {%
22 \let\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 \let\scalebox\LWR@noscalebox%
29 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
30 denominator-format = {%
31 \let\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 \let\scalebox\LWR@noscalebox%
38 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
39 denominator-format = {%
40 \let\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 \let\scalebox\LWR@noscalebox%
47 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
48 denominator-format = {%
49 \let\scalebox\LWR@noscalebox%
50 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},
```

For `pdftotext`, do not scale the text:

```
51 scaling = false
52 }
```

```
53 \end{warpHTML}
```

Change History and Index

Change History

v0.10		Docs: Commands for a successful HTML conversion.	57
General: 2016/03/08 Initial version	1		
v0.11		Docs: Commands into a warpprint environment.	56
General: 2016/03/11	1	Docs: Newclude limitations. . .	68
Added section:		Docs: Table: Cross-referencing data structures.	193
Operating-System portability.	92	Docs: Table: Float data structures.	203
Added section: Selecting the operating system.	55	Docs: Trademarks section. . .	271
Test Suite: limages and index in README.txt	1	Docs: Troubleshooting cross-references.	71
Test Suite: MS-Windows in README.txt	1	Test Suite: Assigned cleveref name for Test Float.	1
v0.12		Test Suite: Floatrow	1
\LWR@newhtmlfile: Bugfix: TOC with numbered files.	123	v0.15	
General: 2016/03/14	1	\printpublished: No div if empty.	144
Global: Uses \p@(type) in float captions.	1	\printsubtitle: No div if empty.	145
Test Suite: Sub-figures	1	General: 2016/04/06	1
v0.13		Added	349
\CaptionSeparator: Fix for newer babel package.	205	Ampersand (&): Fixed handling when passed as an argument.	168
\LWR@LwarpStart: \up and \fup	135	Docs: Added warning icons for items needing special attention.	76
\published: Default to empty published.	141	Docs: Clarify print/HTML output.	55
\subtitle: Default to empty subtitle.	141	Docs: Moved the supported functions table to the introduction.	23
General: 2016/03/24	1	Files: lwarp_formal.css added.	1
Removed package: subfig	1	Fix: steps counter	348
Test Suite: Ordinals, Subcaption	1	Fixed & handling.	346
tikzpicture: Fix dollar-redefined bug for newer package.	423	Test Suite: test_suite_formal.css file added.	1
v0.14		v0.16	
\LWR@htmlsectionfilename: Fix: Links to home page.	99	General: 2016/04/11	1
General: 2016/03/31	1		
floatrow: Added.	344		

\titlingpage: Improved print-output spacing.	146	\LWR@subhyperrefclass: Improved HTML output linebreaks. . .	200
Added XeLaTeX, LuaLaTeX support.	83	\LWR@subinlineimage: Surpress extra space.	201
Docs: Font and UTF-8 support.	54	General: 2016/05/19	1
Docs: Moved location of \usepackage{lwarp}.	55	File: lwarp.css: Improved TOC outline display.	1
Docs: Text not converting.	71	Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1
Fix: amsmath options clash	86	Microtype disabled during HTML generation	83
Fix: newtxmath compatibility.	86	PDF Unicode input characters.	77
Lwarp no longer selects fonts.	54, 83	Test Suite: Verse package	1
Removed package: suffix	1	lateximage: pdfcrop: --hires added.	234
Test Suite: Improved titlingpage.	146	Reorganize \HomeHTMLFilename logic.	234
Test Suite: Lwarp no longer selects fonts.	1	Surpress extra space.	234
Test Suite: Supports XeLaTeX, LuaLaTeX.	1	verse: Supports verse, memoir packages.	436
xfrac: Adjusted for the use of any font:	445	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	253
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	99	\HTMLFileName: Docs: Escape filename underscores.	98
General: 2016/04/14	1	\HomeHTMLFileName: Docs: Escape filename underscores.	98
mdframed: Added.	375	\LWR@LwarpStart: Enabled \\ equal to \newline.	134
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doubledollar: MathJax support.	220
Test Suite: Mdframed	1	\LWR@filestart: lwarp_mathjax.txt loaded.	132
v0.18		\LWR@hspace: Fix: \hspace length computations.	262
\LWR@hspace: \hspace supported.	262	\LWR@minipagestartpars: Surpresses paragraph tags between minipages.	261
\LWR@includegraphicsb: Add: svgz file extension.	243	\LWR@singledollar: MathJax support.	220
em, ex, %, px dimensions preserved.	243	\LWR@tabledatamultirowtag: Added optional args.	188
Fix: \linewidth, \textwidth, \textheight inside a minipage.	243	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	232
Improved HTML output linebreaks.	243	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	253
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic.	209		
\LWR@newhtmlfile: Sidetoc after title, improving responsive design.	122		
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic.	137		
\LWR@subhyperref: Improved HTML output linebreaks.	200		

<code>\rule</code> : Added	266	<code>\LWR@htmldivclass</code> : Added	
<code>\warpHTMLonly</code> : Added	78	optional style	106
<code>\warpprintonly</code> : Replaces		<code>\LWR@htmlclass</code> : Added	
<code>\rowprintedonly</code>	78	optional style	105
General: 2016/06/08	1	<code>\LWR@htmlsectionfilename</code> :	
Added	444	HTMLFileName: removed	
Avoids MathJax	219	additional trailing ‘-’, and may	
cleveref: Loaded		be empty	99
<code>\AtEndPreamble</code>	249	Sections called “Index” or “index”	
CSS for table note item	422	have an underscore prepended	
Docs: <code>multirow</code> browser bug		to their filenames if no prefix	99
workaround	64, 165	<code>\LWR@includegraphicsb</code> : Fix:	
Docs: Math options	55	<code>\linewidth</code> in a floatrow	243
Docs: Table: Cross-referencing		Fix: Expands filename	243
data structures, updated	193	<code>\LWR@longtabledatacaptiontag</code> :	
File: <code>lwarp_mathjax.txt</code> added	1	Fix: Pars in captions	185
File: <code>lwarp.css</code> :		<code>\LWR@section</code> : Combined	
<code>tnoteitemheader</code> added	1	higher-level sections together	
Introduction: MathJax support		into files	126
mentioned	22	<code>\LWR@subhtmlclass</code> :	
MathJax support		Factored code	105
added	221, 225, 227	<code>\SetHTMLFileNumber</code> : Add:	
Options: <code>mathsvg</code> and <code>mathjax</code>	79	Control file numbers	98
Supports colored <code>\rule</code>	441	<code>\cpagerefFor</code> : User-redefinable	
titles: null <code>\pagestyle</code> and		word for page references	250
<code>\thispagestyle</code> for HTML	87	<code>\dotfill</code> : Inserts an ellipsis	259
<code>tikzpicture</code> : Adapts to <code>tikz</code>		<code>\hfill</code> : Inserts a <code>\qqquad</code>	259
version	423	<code>\hrulefill</code> : Inserts a short rule	259
<code>equation</code> : MathJax support	224	<code>\hyperindexref</code> : Print mode	
v0.20		provided in case <code>hyperref</code> not	
<code>\BlockClassSingle</code> : Renamed		used	216
from “ <code>LWR@htmldivclassline</code> ”	107	<code>\pageref</code> : Added	199
<code>\HTMLFileName</code> : No longer escape		<code>\tracinglwarp</code> : Added	91
underscores	98	<code>\warpOSwindows</code> : Auto-detects	
<code>\HomeHTMLFileName</code> : No longer		operating system	93
escape underscores	98	General: 2017/02/09	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added	322
“ <code>inlineclass</code> ”	108	<code>alltt</code> : Added	324
<code>\LWR@LwarpStart</code> : Fix: math cross		bookmark: Added	327
references	135	caption and subcaption	
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		supported	1
extra spaces	111	cleveref and referencing patches:	
No break tags in the start/end of		Applied <code>\AfterEndPreamble</code>	249
a tabular	111	<code>draftwatermark</code> : Added	333
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	261	<code>eso-pic</code> : Added	338
<code>\LWR@filestart</code> : Adds meta		<code>everypage</code> : Added	339
description	132	<code>extramarks</code> : Added	339
<code>\LWR@hspace</code> : Add: Supports		<code>fancyhdr</code> : Added	340
HTML thin breakable space	262	<code>hyperref</code> : Additional user	
		macros	359

keyfloat: Added.	363	Verbatim: Added.	157
letterspace: User-interface		verbatim: Added.	154
emulated.	364	BlockClass: Added optional style.	107
listings: Added.	366	Renamed from "blockclass".	107
lrcaption: Added.	372	LWR@nestspan: Fix: Minipages	
lwarp-newproject: Added.	272	inside a span.	103
microtype: User-interface		v0.21	
emulated.	382	\LR@LwarpStart: Changed	
needspace: Added.	387	lateximages to a .txt file.	134
nowidow: Added.	389	\LR@filestart: Skip title if not	
placeins: Added.	402	given.	132
ragged2e: Added.	403	\LR@newhtmlfile: Skip title if not	
setspace: Improved support.	405	given.	122
textpos: Added.	417	\marginpar: Fixed source listing.	119
titles: Added.	424	General: 2017/02/23	1
titlesec: Added.	426	fontenc: Added.	350
titletoc: Added.	427	fontspec: Added.	350
titling: Improved compatibility.	428	inputenc: Added.	362
tocloft: Added.	429	newclue: Added.	387
wallpaper: Added.	438	newunicodechar: Added.	388
wrapfig: Added.	439	Add: Errors for misplaced	
Added @, <, > columns.	165	packages.	80
Added \NewHTMLdescription.	116	Docs: Added internet class.	28
Added single-expansion data		Docs: Added TeX2page,	
arrays.	97	GladTeX.	28
Code factored into independent		Docs: Installing on Windows.	34
lwarp_html files.	319	File lwarp_tutorial.txt added.	38
Docs: Examples for generating		lwarpmk: Fix: lwarpmk again	
HTML file names.	48	for Windows.	310
Docs: Improved index.	1	lwarpmk: Fix: lwarpmk limages	
Enhanced titling support.	143	for Windows.	310
File: lwarp.css: Minor fixes for		lwarpmk: Fix: lwarpmk uses	
validation.	1	lateximages text file instead of	
File: lwarpmk used to compile		shell script.	310
print, HTML, indexes, and		v0.22	
lateximages.	1	\LR@parseDcolumn: Added	
Fix: \linewidth in a floatrow.	347	tabular D column.	174
Improved float caption type		\LR@parsebangcolumn: Added	
handling.	341	tabular ! column.	171
Moved sidebar and example code		\LR@parsetablecols: Unknown	
to test suite.	1	table column types become l.	
Page geometry set to 6in wide		Added tabular D, !, X columns.	175
with large margins.	84	\LR@printmccoldata: Added	
Parallel versions of aux files for		tabular D, !, and X columns.	182
print/HTML.	1	General: 2017/03/02	1
Removed reliance on make, grep,		abstract: Added.	320
gawk.	1	changepage: Added.	330
Tabular: \unskip extra spaces.	165	dcolumn: Added.	332
Test Suite: HTML meta		enumerate: Added.	336
descriptions.	1	ftnright: Added.	357

geometry: Nullified commands.	357	ccaption: Prevented.	329
indentfirst: Added.	362	ellipsis: Added.	333
layout: Added.	364	emptypage: Added.	334
lscap: Added.	372	framed: Added.	354
mcaption: Added.	374	lips: Added.	365
nameref: Added.	386	mdframed: Help avoid	
nextpage: Added.	388	hyphenation.	376
parskip: Added.	402	ntheorem: Added.	390
showkeys: Added.	406	showidx: Added.	406
sidecap: Added.	407	theorem: Added.	418
tabularx: Added.	416	Basic L ^A T _E X theorems: improved	
varioref: Supported.	67	CSS.	160
verse: Added.	436	Docs: Adds credits for patched	
v0.23		code.	1
\LWR@parsetablecols: Fix for vert		Docs: Testing <code>lwarp</code> .	70
bar column type.	175	Fix: Allows XE ^L A _T E _X and	
\LWR@printmccoldata: Fix for vert		Lua ^L A _T E _X to preload graphics	
bar column type.	182	and <code>graphicx</code> .	81
General: 2017/03/02	1	v0.26	
v0.24		General: 2017/03/31	1
\LWR@hspace: Add: <code>\hspace \fill</code>		cutwin: Added.	331
converts to 2em	262	endnotes: Added.	335
\LWR@htmlfileref: Fix: Index		floatflt: Added.	343
links while <code>\tracinglwarp</code> .	195	footmisc: Added.	350
\hypertocfloat: List of floats		footnotehyper: Added.	353
responds to <code>lofdepth</code> ,		footnote: Added.	352
<code>lotdepth</code> .	214	marginfix: Added.	373
General: 2017/03/15	1	marginnote: Added.	374
floatrow: Support for <code>subfig</code> .	344	mparhack: Added.	383
subfig: Added.	410	pagenote: Supported as-is.	401
tikz: For tikz v3.0.0 or later,		sidenotes: Added.	408
auto-loads tikz babel library if		<code>lwarp.css</code> : Improved responsive	
necessary.	423	<code>marginpar</code> and <code>marginblock</code> .	275
Docs: Filename underscore.	47, 57	Docs: Improved MiK _T E _X install	
No longer preloads <code>subcaption</code> ;		instructions.	31, 34
conflicted with <code>subfig</code> .	86	Dollar span avoided in a	
<code>picture</code> : Fix for inline images.	250	<code>lateximage</code> .	219
<code>tikzpicture</code> : Fix for inline		Footnotes now are L ^A T _E X boxes	
images.	423	instead of <code>pagenotes</code> .	117
v0.25		<code>lateximage</code> : Labels track page	
\LWR@loadnever: Added the ability		numbers of <code>lateximages</code> .	234
to prevent conflicting packages.	81	Print mode now uses a <code>minipage</code>	
\addcontentsline: Handles		of <code>\linewidth</code> .	234
theorems.	208	<code>picture</code> : Fix for <code>\makebox in</code>	
General: 2016/03/22	1	<code>picture</code> .	250
<code>amsthm</code> : Added.	325		

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		
$\backslash \$$	<i>219</i>	$\backslash \text{AmS}$ <i>4176</i>
$\backslash \&$	<i>97</i> , <i>2127</i>	amsmath (package) <i>86</i>
$\backslash ($	<i>3124</i>	amsthm (package) <i>325</i>
$\backslash)$	<i>3124</i>	$\backslash \text{and}$ <i>140</i>
$\backslash @\text{setcpageref}$	<i>3829</i>	array (package) <i>192</i>
$\backslash @\text{setcref}$	<i>3825</i>	AsciiDoc (program) <i>29</i>
$\backslash @\text{setcrefrange}$	<i>3826</i>	AsciiDoctor (program) <i>29</i>
$@\text{author}$	<i>139</i>	Asciidoctor-LaTeX (program) <i>29</i>
$\backslash @\text{begintheorem}$	<i>1986</i>	$\backslash \text{attrib}$ <i>67</i> , <i>153</i> , <i>436</i>
$\backslash @\text{capytype}$	<i>2845</i>	$\backslash \text{attribution}$ <i>1769</i>
$\backslash @\text{currentlabelname}$	<i>2638</i>	$\backslash \text{author}$ <i>59</i> , <i>140</i> , <i>1436</i>
$@\text{date}$	<i>139</i>	autosec <i>125</i>
$\backslash @\text{dlbfloat}$	<i>2828</i>	
$\backslash @\text{endtheorem}$	<i>1994</i>	
$\backslash @\text{float}$	<i>2828</i>	
$\backslash @\text{fnsymbol}$	<i>1720</i>	
$\backslash @\text{makecaption}$	<i>2847</i>	
$\backslash @\text{maketitle}$	<i>1731</i>	
$\backslash @\text{nameedef}$	<i>69</i>	
$\backslash @\text{pargbegintheorem}$	<i>1990</i>	
$@\text{published}$	<i>139</i>	
$@\text{subtitle}$	<i>139</i>	
$@\text{title}$	<i>139</i>	
$\backslash \backslash$	<i>261</i>	
$\$$	<i>220</i>	
$\$ \$$	<i>220</i>	
A		B
abstract (environment)	<i>1760</i>	babel (package) <i>68</i>
abstract (package)	<i>320</i>	$\backslash \text{backmatter}$ <i>1120</i>
$\backslash \text{abstractname}$	<i>49</i> , <i>1759</i>	BaseJobName (option) [lwrap- newproject] <i>272</i>
accents	<i>124</i>	$\backslash \text{BaseJobname}$ <i>504</i>
$\backslash \text{addcontentsline}$	<i>2899</i>	BaseJobname (option) [lwrap] <i>79</i>
Adobe (program)	<i>30</i>	baseline
$\backslash \text{affiliation}$	<i>1429</i>	tabular <i>175</i>
afterpage (package)	<i>322</i>	$\backslash \text{bfseries}$ <i>3988</i>
algorithmic		$\backslash \text{BibTeX}$ <i>4171</i>
with newfloat, trivfloat	<i>434</i>	BlockClass (environment) <i>718</i>
algorithmicx (package)	<i>323</i>	$\backslash \text{BlockClassSingle}$ <i>730</i>
align (environment)	<i>3340</i>	bookmark (package) <i>327</i>
align* (environment)	<i>3362</i>	booktabs (package) <i>329</i>
alltt (package)	<i>324</i>	booleans:
		$\text{CombineHigherDepths}$ <i>48</i> , <i>120</i>
		FileSectionNames <i>48</i> , <i>99</i>
		LWR@amsmultline <i>225</i>
		LWR@doingapar <i>109</i>
		LWR@doinghline <i>166</i>
		$\text{LWR@doingstartpars}$ <i>109</i>
		LWR@doingtbrule <i>167</i>
		$\text{LWR@freezethisfloat}$ <i>205</i>
		LWR@infloatrow <i>242</i>
		$\text{LWR@intabularmetadata}$ <i>167</i>
		$\text{LWR@minipagefullwidth}$ <i>253</i>
		$\text{LWR@minipagethispar}$ <i>253</i>
		$\text{LWR@skippingmrowcell}$ <i>167</i>
		$\text{LWR@starredlongtable}$ <i>185</i>

LWR@startedrow	166	LWR@spandepth	109
LWR@tableparcell	167	LWR@tablecolspos	170
LWR@tracinglwarp	91	LWR@tablecolswidth	170
LWR@validtablecol	175	LWR@tabletotalcols	170
LWR@verbtags	155	LWR@tabletotalcolsnext	170
mathjax	78	LWR@thisfloat	205
usingOSwindows	92	SideTOCDepth	47, 212
warpingHTML	78	tocdepth	47
warpingprint	78	\cpagerefFor	3828
bugs	71	cross-references	
BVerbatim (environment)	1972	missing or incorrect	71
C			
calc (package)	85	CSS	
caption (package)	86	file selection	60
\caption@begin	2878	lwarp.css	60
\caption@end	2878	per HTML page	60
\captionlistentry	2882	project-specific changes	60
\captionof	2912	cutwin (package)	331
\CaptionSeparator	2846	D	
ccaption (package)	329	danger icon	76
center (environment)	3560	\date	59
\centering	135	dcolumn (package)	332
changepage (package)	330	debugging	71
\chapter	1238	\DeclareGraphicsExtensions	3581
\citetitle	1776	\DeclareGraphicsRule	3581
classs:		Deja Vu	54
internet	28	description	
cleveref (package)	249	HTML meta tag	59, 117
cmap (package)	55	description (environment)	2056
CombineHigherDepths (boolean)	48, 120	displaymath (environment)	3133
comment (package)	78	\dotfill	4004
Computer Modern	54	draftwatermark (package)	333
\ConTeXt	4168	E	
counters:		ellipsis (package)	333
FileDepth	47, 119	\emph	3977
lofdepth	214	emptypage (package)	334
lotdepth	214	\end@dlbfloat	2837
LWR@externalfilecnt	219	\end@float	2837
LWR@htmlfilenumber	99	endnotes (package)	335
LWR@latestautopage	206	\enlargethispage	4106
LWR@lateximagedepth	232	\enskip	261, 4070
LWR@lateximagenumber	232	enumerate (environment)	2025
LWR@LIpage	232	enumerate (package)	69, 336
LWR@midrulecounter	180	enumitem (package)	69, 86
LWR@minipagedepth	251	environ (package)	87
LWR@nextautofloat	206	environments:	
LWR@nextautopage	206	abstract	1760
LWR@nextequation	221	align	3340
LWR@prevFileDepth	126	align*	3362

HTMLvleftskip ..	67, 153, 154, 436	lwarp_formal.css (file)	303
LWR@minipageheight	252	lwarp_html.xdy (file)	308
LWR@minipagewidth	251	lwarp_mathjax.txt (file)	308
LWR@tempheight	262	lwarp_sagebrush.css (file)	298
LWR@tempraise	262	lwarp_tutorial.txt (file)	38
LWR@tempwidth	262	lwarpmk (file)	310
\VerbatimHTMLwidth	155	lwarpmk (option) [lwarp-newproject]	272
vleftmargini	67, 153, 436	lwarpmk.conf (file)	270
vleftskip	67, 153, 436, 437	\LWR@addmathjax	3189
letterspace (package)	364	\LWR@afterendverbatim	1824
LibreOffice (program)	29	LWR@amsmultiline (boolean)	225
ligatures	54, 55, 84	\LWR@atbeginverbatim	1816
line numbers	76	\LWR@beginhideamsmath	3234
\linebreak	4100	\LWR@botnavigation	564
\LinkHome	556	\LWR@caption@begin	2859
lips (package)	365	\LWR@caption@end	2869
list item, empty	57	\LWR@clearmidrules	2353
listings (package)	366	\LWR@closeparagraph	805
\listof	2973	\LWR@closeprevious	584
lmodern (package)	54	\LWR@closetabledatcell	2105
lofdepth (counter)	214	LWR@colafterspec	171
longtable (environment)	2	LWR@colatspec	171
longtable (package)	370	LWR@colbangspec	171
lotdepth (counter)	214	LWR@colbeforespec	171
lscap (package)	372	\LWR@copyfile	318
ltcaption (package)	372	\LWR@createautosec	1126
LuaLaTeX		\LWR@currentcss	906
detection	83	\LWR@currentHTMLdescription	921
file & section names	124	\LWR@descitem	2044
\LuaLaTeX	4162	\LWR@docmidrule	2372
LuaLaTeX (program) [requirement] ..	32	LWR@doingapar (boolean)	109
\LuaTeX	4162	LWR@doinghline (boolean)	166
LVerbatim (environment)	1983	LWR@doingstartpars (boolean) ..	109
lwarp		LWR@doingtbrule (boolean)	167
loading	55	\LWR@domulticolumn	2445
options	55	\LWR@doubledollar	3102
[lwarp]:		\LWR@endhideamsmath	3242
BaseJobname (option)	79	\LWR@endminipage	3860
mathjax (option)	55, 79	\LWR@endoffline	4048
mathsvg (option)	55, 79	\LWR@endsubminipage	3881
warpHTML (option)	55, 79	\LWR@ensuredoingapar	779
warpprint (option)	55, 78	LWR@externalfilecnt (counter) ..	219
lwarp (package)	55	\LWR@filenamenoblanks	988
[lwarp-newproject]:		\LWR@filestart	1290
BaseJobName (option)	272	\LWR@findword	254
lwarpmk (option)	272	\LWR@floatbegin	2817
warpHTML (option)	272	\LWR@floatend	2830
warpprint (option)	272	LWR@freezethisfloat (boolean) ..	205
lwarp-newproject (package)	272	\LWR@futurenonospacelet	2084
lwarp.css (file)	60, 275	\LWR@getexparray	490

<code>\LWR@getmynexttoken</code>	2091	<code>LWR@midrules</code>	180
<code>\LWR@hidelatexequation</code>	3175	<code>LWR@minipagedepth (counter)</code>	251
<code>\LWR@hspace</code>	4078	<code>LWR@minipagefullwidth (boolean)</code> ..	253
<code>\LWR@htmlblockcomment</code>	657	<code>LWR@minipageheight (length)</code>	252
<code>\LWR@htmlblocktag</code>	665	<code>\LWR@minipagestartpars</code>	4052
<code>\LWR@htmlblocktagcomment</code>	670	<code>\LWR@minipagestoppars</code>	4058
<code>\LWR@htmlclosecomment</code>	644	<code>LWR@minipagethispar (boolean)</code> ..	253
<code>\LWR@htmlcomment</code>	653	<code>LWR@minipagewidth (length)</code>	251
<code>\LWR@htmldivclass</code>	695	<code>\LWR@multicolother</code>	2397
<code>\LWR@htmldivclassend</code>	698	<code>\LWR@multicolparttext</code>	2392
<code>\LWR@htmlmlelement</code>	708	<code>\LWR@multicolskip</code>	2403
<code>\LWR@htmlmlelementclass</code>	684	<code>\LWR@mynexttoken</code>	2083
<code>\LWR@htmlmlelementclassend</code>	689	<code>\LWR@myshorttoc</code>	2924
<code>\LWR@htmlmlelementclassline</code>	701	<code>\LWR@nameref</code>	2661
<code>\LWR@htmlmlelementend</code>	711	<code>LWR@nestspan (environment)</code>	613
<code>LWR@htmlfilenumber (counter)</code>	99	<code>\LWR@newhtmlfile</code>	1066
<code>\LWR@htmlfileref</code>	2664	<code>\LWR@newlabel</code>	2698
<code>\LWR@htmlmathlabel</code>	3220	<code>\LWR@newref</code>	2747
<code>\LWR@htmlopencomment</code>	644	<code>LWR@nextautofloat (counter)</code>	206
<code>\LWR@htmlrefsectionfilename</code>	549	<code>LWR@nextautopage (counter)</code>	206
<code>\LWR@HTMLsanitize</code>	3443	<code>LWR@nextequation (counter)</code>	221
<code>\LWR@HTMLsanitizeexpand</code>	3461	<code>\LWR@openparagraph</code>	784
<code>\LWR@htmlsectionfilename</code>	516	<code>\LWR@origcolspec</code>	2158
<code>\LWR@htmlspan</code>	619	<code>\LWR@originname</code>	3773
<code>\LWR@htmlspanclass</code>	627	<code>\LWR@originnames</code>	3780
<code>\LWR@htmltag</code>	635	<code>\LWR@parseaftercolumn</code>	2215
<code>\LWR@htmltagc</code>	606	<code>\LWR@parseatcolumn</code>	2163
<code>\LWR@includegraphicsb</code>	3672	<code>\LWR@parsebangcolumn</code>	2185
<code>\LWR@indexitem</code>	3064, 3068, 3072	<code>\LWR@parsebeforecolumn</code>	2206
<code>\LWR@indexsection</code>	3048	<code>\LWR@parseDcolumn</code>	2242
<code>LWR@infloatrow (boolean)</code>	242	<code>\LWR@parsenormalcolumn</code>	2228
<code>LWR@intabularmetadata (boolean)</code> ..	167	<code>\LWR@parsepcolumn</code>	2238
<code>\LWR@itemizeitem</code>	2000	<code>\LWR@parseskipcolumn</code>	2225
<code>LWR@latestautopage (counter)</code> ...	206	<code>\LWR@parsetablecols</code>	2247
<code>LWR@lateximagedepth (counter)</code> ..	232	<code>LWR@prevFileDepth (counter)</code>	126
<code>\LWR@lateximagedepthref</code>	2668	<code>\LWR@printmccoldata</code>	2406
<code>LWR@lateximagenumber (counter)</code> ..	232	<code>\LWR@printmccoltype</code>	2376
<code>\LWR@lateximagenumberref</code>	2671	<code>\LWR@printheadtitle</code>	1562
<code>LWR@LIpage (counter)</code>	232	<code>\LWR@ProvidesPackageDrop</code>	308
<code>\LWR@loadafter</code>	70	<code>\LWR@ProvidesPackagePass</code>	296
<code>\LWR@loadbefore</code>	81	<code>\LWR@pushoneclose</code>	1129
<code>\LWR@loadnever</code>	86	<code>\LWR@requesttoc</code>	1403
<code>\LWR@longtabledatacaptiontag</code> ..	2466	<code>\LWR@requirepackagenames</code>	253
<code>\LWR@lookforpackagename</code>	257	<code>\LWR@restoremathlatexformatting</code>	3157
<code>\LWR@LwarpEnd</code>	1410, 4198	<code>\LWR@rotboxorigin</code>	3772
<code>\LWR@LwarpStart</code>	1344, 4198	<code>\LWR@rotstyle</code>	3662
<code>\LWR@maketitlesetup</code>	1711	<code>\LWR@rule</code>	4107
<code>\LWR@maybeinthisfloat</code>	2842	<code>\LWR@scalestyle</code>	3665
<code>\LWR@maybeinnewtablerow</code>	2302	<code>\LWR@section</code>	1136
<code>LWR@midrulecounter (counter)</code> ...	180	<code>\LWR@sectionnumber</code>	1123

minipage			
horizontal space between	261		
in an <code>\fbox</code>	58		
minipage (environment)	3896		
<code>\minipagefullwidth</code>	3888		
missing sections	47, 48		
mparhack (package)	383		
<code>\mrowcell</code>	2592		
MS-Windows (program)	55, 92		
multicol (package)	384		
multirow (package)	188, 386		
multline (environment)	3249		
multline* (environment)	3272		
N			
<code>\Nameref</code>	2768		
<code>\nameref</code>	2759		
nameref (package)	386		
needspace (package)	387		
newclude (package)	68, 387		
<code>\NewCSS</code>	46, 49, 60, 907		
newfloat			
with <code>trivfloat</code> , <code>algorithmic</code>	434		
<code>\newfloat</code>	2		
newfloat (package)	86		
<code>\NewHTMLdescription</code>	50, 59, 922		
<code>\newline</code>	4046		
<code>\newlist</code>	2074		
<code>\newtheorem</code>	160		
newtxmath (package)	68		
newunicodechar (package)	54, 388		
nextpage (package)	388		
<code>\nolinebreak</code>	4101		
<code>\nolinkurl</code>	2792		
<code>\nopagebreak</code>	4105		
nowidow (package)	389		
ntheorem (package)	390		
<code>\numberline</code>	3003		
numbers			
left margin	76		
O			
OpenOffice (program)	29		
options:			
[<code>lwrap-newproject</code>]:			
BaseJobName	272		
lwrapmk	272		
warpHTML	272		
warpprint	272		
[<code>lwrap</code>]:			
BaseJobname	79		
mathjax	55, 79		
mathsvg	55, 79		
warpHTML	55, 79		
warpprint	55, 78		
<code>\OSPathSymbol</code>	387		
P			
packages			
required	83		
packages:			
abstract	320		
afterpage	322		
algorithmicx	323		
alltt	324		
amsmath	86		
amsthm	325		
array	192		
babel	68		
bookmark	327		
booktabs	329		
calc	85		
caption	86		
ccaption	329		
change page	330		
cleveref	249		
cmap	55		
comment	78		
cutwin	331		
dcolumn	332		
draftwatermark	333		
ellipsis	333		
emptypage	334		
endnotes	335		
enumerate	69, 336		
enumitem	69, 86		
environ	87		
epigraph	337		
eso-pic	338		
etoolbox	77		
everyhook	85		
everypage	339		
expl3	85		
extramarks	339		
fancyhdr	340		
fancyvrb	85		
float	341		
floatflt	343		
floatrow	344		

fontenc	54, 350	refcount	86
fontspec	54, 350	rotating	404
footmisc	351	setspace	405
footnote	352	showidx	406
footnotehyper	353	showkeys	406
framed	354	sidecap	407
ftnright	357	sidenotes	408
geometry	84, 357	siunitx	68, 238
getttitlestring	85	subfig	410
graphics	238, 358	tabularx	416
graphicx	238, 358	textcomp	55
hyperref	200, 359	textpos	417
ifplatform	78	theorem	418
indentfirst	362	threeparttable	422
inputenc	54, 362	tikz	423
keyfloat	363	titleps	87, 424
kvoptions	78	titlesec	426
layout	364	titletoc	427
letterspace	364	titling	86, 428
lips	365	tocloft	429
listings	366	trivfloat	434
lmodern	54	verse	436, 437
longtable	370	wallpaper	438
lscap	372	wrapfig	439
ltcaption	372	xcolor	441
lwrap	55	xfrac	87, 444
lwrap-newproject	272	xifthen	85
makeidx	85	xparse	84
marginfix	373	xstring	85
marginnote	374	zref	86
mcaption	374	page	
mdframed	375	inaccessible	47, 48
memoir	437	\pagebreak	4102
microtype	55, 83, 382	pagenote (package)	401
mparhack	383	\pagenumbering	246
multicol	384	\pageref	2754
multirow	188, 386	\pagerefPageFor	2753
nameref	386	\pagestyle	242
needspace	387	Pandoc (program)	29
newclude	68, 387	\paragraph	1273
newfloat	86	\parbox	3952
newtxmath	68	\parsemulticolumnalignment	2433
newunicodechar	54, 388	parskip (package)	402
nextpage	388	\part	1227
nowidow	389	pdfcrop (program) [requirement]	32
ntheorem	390	pdfLaTeX (program) [requirement]	32
pagenote	401	pdfseparate (program) [requirement]	32, 36
parskip	402	pdftocairo (program) [requirement]	32, 36
placeins	402		
ragged2e	403		

TtH (program)	28		
tutorial.tex (file)	38		
U			
underscore			
filename	47, 57		
Unicode			
enhanced coverage	54		
file & section names	124		
input characters	77		
selection	54		
\up	4000		
\upshape	3992		
\url	2799		
\UseLatexmk	47		
\UseVerbatim	1984		
usingOSWindows (boolean)	92		
UTF-8			
enhanced coverage	54		
file & section names	124		
selection	54		
utility			
programs	31		
V			
Verbatim (environment)	1840		
verbatim (environment)	1799		
VerbatimClass (environment)	1832		
\VerbatimHTMLWidth (length)	155		
verse (environment)	2		
verse (package)	436, 437		
viewport			
HTML meta tag	133		
vleftmargini (length)	67, 153, 436		
vleftskip (length)	67, 153, 436, 437		
W			
wallpaper (package)	438		
warning icon	76		
warpall (environment)	34, 56		
warpHTML (environment)	42, 50, 56		
warpHTML (option) [lwarp-newproject]	272		
warpHTML (option) [lwarp]	55, 79		
\warpHTMLonly	33, 50, 56		
warpingHTML (boolean)	78		
warpingprint (boolean)	78		
\warpOSWindows	55, 92, 388		
warpprint (environment)	35, 50, 56		
warpprint (option) [lwarp-newproject]	272		
warpprint (option) [lwarp]	55, 78		
\warpprintonly	32, 50, 56		
Windows (program)	55, 92		
Word (program)	29		
wrapfig (package)	439		
X			
xcolor (package)	441		
XeLaTeX			
detection	83		
file & section names	124		
\XeLaTeX	4164		
XeLaTeX (program) [requirement]	32		
\XeTeX	4164		
xfrac (package)	87, 444		
xifthen (package)	85		
xparse (package)	84		
xstring (package)	85		
Z			
zref (package)	86		