



The **lwarp** package

v0.25 — 2017/03/22

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 X_YL^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	18
1	Updates	18
2	Introduction	21
2.1	Supported packages and features	22
3	Alternatives	27
3.1	Internet class	27
3.2	TeX4ht	27
3.3	Translators	27
3.4	AsciiDoc	28
3.5	Pandoc	28
3.6	Word Processors	28
3.7	Commercial Systems	29
3.8	Comparisons	29
4	Installation	30
4.1	Installing the lwarp package	30
4.2	Installing the lwarpmk utility	33
4.2.1	Using a local copy of lwarpmk	34
4.3	Installing additional utilities	35
5	Tutorial	37
5.1	Starting a new project	37
5.2	Compiling the print version with lwarpmk	41
5.3	Compiling the HTML version with lwarpmk	42
5.4	Generating the SVG images	43
5.5	Using MathJax for math	44
5.6	Changing the CSS style	45
5.7	Customizing the HTML output	46
5.8	Using latexmk	50
5.9	Using XeLaTeX or LuaLaTeX	51
5.10	Cleaning auxiliary files	52
5.11	Cleaning auxiliary and output files	52
5.12	Processing multiple projects in the same directory	52
5.13	Using the make utility	52
6	Additional details	53
6.1	Font and UTF-8 support	53
6.2	lwarp package loading and options	54
6.3	Selecting the operating system	54
6.4	Selecting actions for print or HTML output	54
6.5	Commands to be placed into the warpprint environment	55

6.6	Commands for a successful HTML conversion	56
6.7	Title page	57
6.8	HTML page meta descriptions	58
6.9	CSS	59
6.10	Special cases and limitations	59
6.10.1	Text formatting	59
6.10.2	Math	59
6.10.3	Ntheorem	61
6.10.4	Graphics	61
6.10.5	Xcolor	62
6.10.6	Tabular	62
6.10.7	Minipages	63
6.10.8	Mdframed	64
6.10.9	float, trivfloat, and/or algorithmicx together	65
6.10.10	caption and subcaption packages	65
6.10.11	subfig package	65
6.10.12	floatrow package	65
6.10.13	abstract package	66
6.10.14	cleveref and varioref packages	66
6.10.15	verse and memoir	66
6.10.16	siunitx package	67
6.10.17	newclude package	67
6.10.18	newtxmath package	67
6.10.19	babel package	67
6.10.20	enumitem package	68
6.10.21	enumerate package	68
7	Expanding the code base	68
7.1	Creating an lwarp version of a package	68
7.2	Testing lwarp	69
8	Troubleshooting	70
8.1	Using the <code>lwarp.sty</code> package	70
8.1.1	Debug tracing output	71
8.2	Compiling the <code>lwarp.dtx</code> file	72
9	Implementation	72
10	Stack depths	73
11	Source Code	74
12	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	75
13	Unicode Input Characters	75
14	Handling package options	75

15	Misplaced packages	78
16	Required packages	81
17	Loading Packages	86
18	Copying a file	88
19	Debugging messages	89
20	Remembering original formatting macros	89
21	Operating-System portability	90
21.1	Common portability code	91
21.2	Unix, Linux, and Mac OS	91
21.3	MS-Windows	91
22	Stacks	91
22.1	Assigning depths	92
22.2	Closing actions	92
22.3	Closing depths	93
22.4	Pushing and popping the stack	93
23	Data arrays	95
24	HTML entities	95
25	HTML filename generation	96
26	Homepage link	98
27	\PrintStack diagnostic tool	99
28	Closing stack levels	100
29	HTML tags, spans, divs, elements	100
29.1	Mapping L ^A T _E X Sections to HTML Sections	100
29.2	HTML tags	101
29.3	Block tags and comments	102
29.4	Div class and element class	103
29.5	Single-line elements	104
29.6	HTML5 semantic elements	105
29.7	High-level block and inline classes	105
29.8	Closing HTML tags	106
30	Paragraph handling	107
31	Paragraph start/stop handling	110

32	Page headers and footers	113
33	CSS	114
34	HTML meta description	114
35	Footnotes	115
35.1	Regular page footnotes	115
35.2	Minipage footnotes	115
35.3	Titlepage thanks	116
35.4	Regular page footnote implementation	116
36	Marginpars	117
37	Splitting HTML files	118
38	Sectioning	122
38.1	Book class commands	123
38.2	Sectioning support macros	123
38.3	\section and friends	128
39	Starting a new file	130
40	Starting HTML output	132
41	Ending HTML output	135
42	Titles and the titling package	137
42.1	Setting the title, etc.	138
42.2	Changes to HTML titlepage and titlingpage	140
42.3	Printing the title, etc. in HTML	141
42.4	Printing the title, etc. in print form	144
42.5	\maketitle for print output	145
42.6	\maketitle for HTML output	147
43	Abstract	150
44	Quote and verse	151
44.1	Citations and attributions	151
44.2	Quotes, quotations	151
44.3	Verse	152
45	Verbatim	153
46	Fancyvrb	153
47	Theorems	159

48	Lists	160
48.1	Itemize	160
48.2	Enumerate	161
48.3	Description	162
49	Tabular	163
49.1	Token lookahead	164
49.2	Booleans	165
49.3	Handling & and !	166
49.4	Handling \\	168
49.5	Variables	168
49.6	Parsing @, >, <, ! columns	170
49.7	Parsing 'l', 'c', or 'r' columns	172
49.8	Parsing 'p', 'm', or 'b' columns	173
49.9	Parsing 'D' columns	173
49.10	Parsing the column specifications	173
49.11	Starting a new row	177
49.12	Data opening tag	177
49.13	Midrules	179
49.14	Multicolumns	180
49.14.1	Parsing multicolumns	180
49.14.2	High-level multicolumn interface	183
49.14.3	Longtable captions	183
49.14.4	\tabledatamulticolumntag	186
49.15	Multirow	186
49.16	Utility macros inside a table	187
49.17	Checking for a new table cell	187
49.18	\mrowcell	188
49.19	New \tabular definition	189
49.20	Array	191
50	Cross-references	191
50.1	Setup	191
50.2	Zref setup	193
50.3	Labels	194
50.4	References	196
50.5	Hyper-references	198
51	Floats	201
51.1	Float captions	202
51.1.1	Caption inside a float environment	203
51.1.2	Caption and LOF linking and tracking	203
52	Table of Contents, LOF, LOT	207
52.1	Reading and printing the TOC	207
52.2	High-level TOC commands	209

52.3	Side TOC	209
52.4	Low-level TOC line formatting	211
53	Index	213
54	Math	215
54.1	Inline and display math	217
54.2	MathJax support	219
54.3	Equation environment	222
54.4	AMS Math environments	223
54.4.1	Support macros	223
54.4.2	Environment patches	225
55	Lateximages	229
56	center, flushleft, flushright	235
57	Siunitx	235
58	Graphics	236
58.1	\graphicspath	237
58.2	Length conversions and graphics options	237
58.3	\includegraphics	240
58.4	\rotatebox, \scalebox, \reflectbox	244
58.5	Null functions	246
59	Cleverref	247
60	Picture	248
61	Boxes and Minipages	248
61.1	Counters and lengths	249
61.2	Footnote handling	250
61.3	Minipage handling	250
61.4	Parbox, makebox, framebox, fbox, raisebox	254
62	Direct formatting	255
63	Skips, spaces, font sizes	258
64	\phantomsection	265
65	\LaTeX and other logos	265
65.1	HTML logos	265
65.2	Print logos	267
66	\AtBeginDocument, \AtEndDocument	267

67	lwarpmk.conf	269
68	Trademarks	270
2	lwarp-newproject.sty	271
69	lwarp-newproject	271
69.1	project_html.tex	273
69.2	project.lwarpmkconf	273
69.3	lwarp.css	274
69.4	lwarp_sagebrush.css	297
69.5	lwarp_formal.css	301
69.6	sample_project.css	306
69.7	lwarp_html.xdy	306
69.8	lwarp_mathjax.txt	307
69.9	Lwarpmk option	309
3	lwarp-abstract.sty	319
70	Abstract	319
4	lwarp-afterpage.sty	321
71	Afterpage	321
5	lwarp-algorithmicx.sty	322
72	Algorithmicx	322
6	lwarp-alltt.sty	322
73	Alltt	323
7	lwarp-amsthm.sty	324
74	AMSthm	324
8	lwarp-bookmark.sty	326
75	Bookmark	326

9	lwarp-booktabs.sty	328
76	Booktabs	328
10	lwarp-ccaption.sty	328
77	Ccaption	328
11	lwarp-changepage.sty	329
78	Changepage	329
12	lwarp-dcolumn.sty	329
79	Dcolumn	329
13	lwarp-draftwatermark.sty	330
80	Draftwatermark	330
14	lwarp-ellipsis.sty	330
81	Ellipsis	330
15	lwarp-emptypage.sty	331
82	Emptypage	331
16	lwarp-enumerate.sty	332
83	Enumerate	332
17	lwarp-epigraph.sty	333
84	Epigraph	333
18	lwarp-eso-pic.sty	334
85	Eso-pic	334

19	lwarp-everypage.sty	335
86	Everypage	335
20	lwarp-extramarks.sty	335
87	Extramarks	335
21	lwarp-fancyhdr.sty	336
88	Fancyhdr	336
22	lwarp-float.sty	337
89	Float and \newfloat	337
23	lwarp-floatrow.sty	339
90	Floatrow	339
24	lwarp-fontenc.sty	345
91	Fontenc	345
25	lwarp-fontspec.sty	345
92	Fontspec	345
26	lwarp-framed.sty	346
93	Framed	346
27	lwarp-ftnright.sty	349
94	Ftnright	349
28	lwarp-geometry.sty	349
95	Geometry	349

29	lwarp-graphics.sty	350
96	Graphics	350
30	lwarp-graphicx.sty	350
97	Graphicx	350
31	lwarp-hyperref.sty	351
98	Hyperref	351
32	lwarp-indentfirst.sty	354
99	Indentfirst	354
33	lwarp-inputenc.sty	354
100	Inputenc	354
34	lwarp-keyfloat.sty	355
101	Keyfloat	355
35	lwarp-layout.sty	355
102	Layout	356
36	lwarp-letterspace.sty	356
103	Letterspace	356
37	lwarp-lips.sty	357
104	Lips	357
38	lwarp-listings.sty	358
105	Listings	358

39	lwarp-longtable.sty	362
106	Longtable	362
40	lwarp-lscape.sty	364
107	Lscape	364
41	lwarp-ltcaption.sty	364
108	Ltcaption	364
42	lwarp-mcaption.sty	365
109	Mcaption	365
43	lwarp-mdframed.sty	366
110	Mdframed	366
44	lwarp-microtype.sty	373
111	Microtype	373
45	lwarp-multicol.sty	374
112	Multicol	374
46	lwarp-multirow.sty	376
113	Multirow	376
47	lwarp-nameref.sty	376
114	Nameref	376
48	lwarp-needspace.sty	377
115	Needspace	377

49	lwarp-newclude.sty	377
116	Newclude	377
50	lwarp-newunicodechar.sty	378
117	Newunicodechar	378
51	lwarp-nextpage.sty	378
118	Nextpage	378
52	lwarp-nowidow.sty	378
119	Nowidow	379
53	lwarp-ntheorem.sty	380
120	Ntheorem	380
54	lwarp-pagenote.sty	391
121	Pagenote	391
55	lwarp-parskip.sty	392
122	Parskip	392
56	lwarp-placeins.sty	392
123	Placeins	392
57	lwarp-ragged2e.sty	393
124	Ragged2e	393
58	lwarp-rotating.sty	394
125	Rotating	394

59	lwarp-setspace.sty	395
126	Setspace	395
60	lwarp-showidx.sty	395
127	Showidx	396
61	lwarp-showkeys.sty	396
128	Showkeys	396
62	lwarp-sidecap.sty	397
129	Sidecap	397
63	lwarp-subfig.sty	398
130	Subfig	398
64	lwarp-tabularx.sty	404
131	Tabularx	404
65	lwarp-textpos.sty	405
132	Textpos	405
66	lwarp-theorem.sty	406
133	Theorem	406
67	lwarp-threeparttable.sty	410
134	Threeparttable	410
68	lwarp-tikz.sty	411
135	Tikz	411

69	lwarp-titleps.sty	412
136	Titleps	412
70	lwarp-titlesec.sty	414
137	Titlesec	414
71	lwarp-titletoc.sty	415
138	Titletoc	415
72	lwarp-titling.sty	416
139	Titling	416
73	lwarp-tocloft.sty	417
140	Tocloft	417
74	lwarp-trivfloat.sty	422
141	Trivfloat	422
141.1	Combining \newfloat, \trivfloat, and algorithmicx	422
75	lwarp-verse.sty	424
142	Verse	424
76	lwarp-wallpaper.sty	426
143	Wallpaper	426
77	lwarp-wrapfig.sty	427
144	Wrapfig	427

78	lwarp-xcolor.sty	429
-----------	-------------------------	------------

145	Xcolor	429
-----	--------	-----

79	lwarp-xfrac.sty	432
-----------	------------------------	------------

146	Xfrac	432
-----	-------	-----

	Change History and Index	435
--	---------------------------------	------------

List of Figures

1	tutorial.tex listing	38
---	--------------------------------	----

List of Tables

1	L ^A T _E X–HTML generation — lwarp package — Supported functions	22
2	Additional supported packages	25
3	Required software programs	31
4	Files created in a new project	40
5	Section depths and HTML headings	73
6	Tabular baseline	174
7	Tabular HTML column conversions	174
8	Cross-referencing data structures	192
9	Float data structures	201

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 [435](#).

v0.25:

- Fix: Allows `graphicx` and `graphicsx` before `lwarp` because \TeX and \LaTeX 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 6.10.2. 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 $\text{X}\text{L}\text{A}\text{T}\text{E}\text{X}$ and $\text{L}\text{u}\text{a}\text{L}\text{A}\text{T}\text{E}\text{X}$. 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 placed 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 22.)

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, XeL ^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:	Supported, emulated as pagenotes per HTML page.
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. \mathcal{AMS} 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.
<code>multirow</code> :	Emulated.

lwarp Supported Functions — continued

Package or Macro	Status
<code>longtable:</code>	Emulated. Converted to a tabular. Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
<code>booktabs:</code>	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.
<code>threeparttable:</code>	Emulated
<code>graphics, graphicx:</code>	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.
<code>rotating:</code>	Emulated. All objects are displayed unrotated.
<code>Lists:</code>	Supported
<code>enumitem:</code>	Supported, although spacing is still controlled by CSS.
<code>Environments:</code>	Standard L ^A T _E X environments are supported.
<code>picture</code> and <code>tikz:</code>	Converted to an SVG image.
<code>minipage:</code>	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
<code>fancyvrb:</code>	Supported except for verbatim footnotes.
<code>framed, mdframed:</code>	Supported
<code>multicol:</code>	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
<code>siunitx:</code>	Supported except for <code>per-mode=fraction</code> .
<code>xfrac:</code>	Supported

lwarp Supported Functions — continued

Package or Macro	Status
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>fntcount</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 thin-unbreakable, unbreakable, <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</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 L ^A T _E X package is used.
Emulated:	The L ^A T _E X 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 deciding which path to take.

3.1 Internet class

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

3.2 TeX4ht

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

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

3.3 Translators

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

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

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

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

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

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

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

Prog **TeX2page** **TeX2page**: <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®
Prog	FrameMaker	FrameMaker®, Adobe® InDesign®, and Madcap Flare™.
Prog	InDesign	
	Prog	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.

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
```

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

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.

pdftcrop: 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

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

1. See <http://ctan.org/pkg/lwarp> for the lwarp package.
2. Download the zip archive `lwarp.zip` into your own `lwarp` directory.
3. Unpack `lwarp.zip`.
4. Locate the contents `lwarp.dtx` and `lwarp.ins`
5. Create the documentation:

Enter ⇒ `pdflatex lwarp.dtx` (several times)

6. Create the .sty files:

Enter ⇒ `pdflatex lwarp.ins`

7. Copy the .sty files somewhere such as the T_EX Live local tree found in the previous CTAN TDS section, under the subdirectory:

`<texlocal>/tex/latex/local/lwarp`

8. Copy the documentation `lwarp.pdf` to a `source` directory in the local tree, such as:

`<texlocal>/doc/local/lwarp`

9. Renew the cache:

Enter ⇒ `mktexlsr` — or —

Enter ⇒ `texhash`

Or, for Windows MiK_TE_X, 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.

Otherwise, continue with the following:

2. Locate the file **lwarpmk.lua**, which should be in the **scripts** directory of the TDS tree. On a T_EX Live or MiK_TE_X 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.)

3. 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 MiK_TE_X: 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 31.

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 ⇒ texdoc -l lwarp_tutorial.txt
```

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

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

⚠ Bad Formatting!

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

3. Compile the project:

```
Enter ⇒ pdflatex tutorial.tex (several times)
```

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

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 lwrap 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{lwrap-newproject}   % Possibly with the [lwrapmk] option.
\usepackage{lwrap}
% \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 Lwrap 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{lwrap_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.

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

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 L^AT_EX source.

While using MathJax has many advantages, it may not be able to represent complex expressions or spacing adjustments as well as L^AT_EX.

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 \LaTeX 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}`:

Ctrl <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.
Ctrl <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.
⚠	
Ctrl <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.

HTML filenames

Example HTML filenames:

Numbered HTML nodes:

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

```
\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 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 6.10.2.
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^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`, `\qquad`, `\hspace`, or `\hfill`. The result is similar in print and HTML. Do not use space commands at the start or end of the line.

\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 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 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 to pre-set the equation number format and value according to the current L^AT_EX values, and the MathJax cross-referencing system is ignored in favor of the L^AT_EX internal system, seamlessly integrating with the rest of the L^AT_EX code.

MathJax limitations Limitations when using MathJax include:

Prog **MathJax**

⁶See section 146 regarding fonts and fractions.

chapter numbers

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

<https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY>

subequations

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

footnotes in math

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

lateximage

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

siunitx

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

⚠ `siunitx` inside an equation

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

 \LaTeX macros

- MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.

custom MathJax macros

For an example of using custom \LaTeX macros with MathJax, see page 217.

6.10.3 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.

6.10.4 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 L^AT_EX, so expect some ugly results for scaling and rotating.

6.10.5 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.

6.10.6 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.
 - `\newcolumntype` is ignored; unknown column types are set to 1.

⚠ `\multirow &`
`\multicolumn`

⚠ `\multirow`

`\multirow with rules`

rule at last row

⚠ paragraphs

`\cmidrule width, trim`

`longtable headings`

⚠ `\warpprintonly`

⚠ S columns

- `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 \\}`

6.10.7 Minipages

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

6.10.8 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.

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

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 141.1.


6.10.10 `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`.


6.10.11 `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.

6.10.12 `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.

6.10.13 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.

6.10.14 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.

6.10.15 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}
```


<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>\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 `\TMLleftmargini` or `\TMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

6.10.16 siunitx package

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


6.10.17 newclude package

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

Ex:


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

6.10.18 newtxmath package

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

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

6.10.19 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}{:~}`

6.10.20 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.

6.10.21 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 T_EX can see it, then the user's new package will be seen by any documents using `lwarp`. (Remember `mktextlsr` 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 120 (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.

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.

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

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_TE_X, 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` $\{(\textit{badpackagename})\} \{(\textit{replacementpkgname})\}$

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{changepage}
100 \LWR@loadafter{dcolumn}
101 \LWR@loadafter{draftwatermark}
102 \LWR@loadafter{ellipsis}
103 \LWR@loadafter{emptypage}
104 \LWR@loadafter{enumerate}
105 \LWR@loadafter{epigraph}
106 \LWR@loadafter{eso-pic}
107 \LWR@loadafter{everypage}
108 \LWR@loadafter{extramarks}
109 \LWR@loadafter{fancyhdr}
110 \LWR@loadafter{floatrow}
111 \LWR@loadafter{float}
112 \LWR@loadafter{ftnright}
113 \LWR@loadafter{geometry}
114 % \LWR@loadafter{graphics}% pre-loaded by xunicode
115 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
116 \LWR@loadafter{hyperref}
117 \LWR@loadafter{indentfirst}
118 \LWR@loadafter{keyfloat}
119 \LWR@loadafter{layout}
120 \LWR@loadafter{letterspace}

```

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

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

Detects \TeX and \LuaTeX :

```
164 \RequirePackage{iftex}
165 \newif\ifxetexorluatex
166 \ifXeTeX
167   \xetexorluatextrue
168 \else
169   \ifLuaTeX
170     \xetexorluatextrue
171   \else
172     \xetexorluatexfalse
173   \fi
174 \fi

175 \end{warpall}
```

for HTML output: 176 `\begin{warpHTML}`

```
177 \ifxetexorluatex
178 % ^^A \usepackage[no-math]{fontspec}
```

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

```
179 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
180 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
181 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
182 \else
```

pdf \LaTeX only: Only pre-loaded if `pdf \LaTeX` is being used.

Pkg microtype

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

```

183 \RequirePackage {microtype}
184
185 \microtypesetup{
186 protrusion=false,
187 expansion=false,
188 tracking=false,
189 kerning=false,
190 spacing=false}
191
192 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

193 \fi

194 \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:

```

195 \begin{warpHTML}
196 \RequirePackage[paperheight=190in,paperwidth=20in,%
197 left=7in,right=7in,%
198 top=1in,bottom=1in,%
199 ]{geometry}
200 \end{warpHTML}

```

for HTML & PRINT:

```

201 \begin{warpall}

```

Pkg xparse

L^AT_EX3 command argument parsing

```

202 \RequirePackage{xparse}

203 \end{warpall}

```

for HTML output:

```

204 \begin{warpHTML}

```

Pkg **expl3**

L^AT_EX3 programming

205 \RequirePackage{expl3}

Pkg **getttitlestring**

Used to emulate \nameref.

206 \RequirePackage{getttitlestring}

Pkg **everyhook**

everyhook is used to patch paragraph handling.

207 \RequirePackage{everyhook}

208 \end{warpHTML}

for HTML & PRINT: 209 \begin{warpall}

Pkg **fancyvrb**

Used for Verbatim, verse.

210 \RequirePackage{fancyvrb}

211 \end{warpall}

for HTML output: 212 \begin{warpHTML}

Pkg **xifthen**

213 \RequirePackage{xifthen}

Pkg **xstring**

214 \RequirePackage{xstring}

Pkg **makeidx**

215 \RequirePackage{makeidx}

216 \makeindex

Pkg **calc**

217 \RequirePackage{calc}

Pkg **newfloat**

```
218 \RequirePackage{newfloat}
```

Pkg **caption**

```
219 \RequirePackage{caption}
```

Pkg **enumitem**

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

```
220 \RequirePackage{enumitem}
221 \setlist[itemize]{leftmargin=0em}
222 \setlist[enumerate]{leftmargin=0em}
223 \setlist[description]{leftmargin=0em}
```

Pkg **pagenote**

Used to convert footnotes into pagenotes.

```
224 \RequirePackage[continuous]{pagenote}
225 \makepagenote
226 \let\LWR@origmakepagenote\makepagenote
227 \let\LWR@origprintnotes\printnotes
228 \let\LWR@origpagenote\pagenote
229 \end{warpHTML}
```

for HTML & PRINT: 230 `\begin{warpall}`

Pkg **titling**

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

```
231 \RequirePackage{titling}

232 \end{warpall}
```

for HTML output: 233 `\begin{warpHTML}`

Pkg **zref**

Used for cross-references.

```
234 \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.

```
235 \PassOptionsToPackage{leqno,fleqn}{amsmath}
236 \RequirePackage{amsmath}
```

Pkg `environ`

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

```
237 \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.

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

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

```
239 \let\LWR@origpagestyle\pagestyle
240 \renewcommand*\pagestyle}[1]{}%
```

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

```
241 \let\LWR@origthispagestyle\thispagestyle
242 \renewcommand*\thispagestyle}[1]{}%
```

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

```
243 \let\LWR@origpagenumbering\pagenumbering
244 \renewcommand*\pagenumbering}[1]{}%
```

Pkg `xfrac`

Patched for HTML use. See section [146](#).

```
245 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
246 \RequirePackage{printlen}
```

247 \end{warpHTML}

17 Loading Packages

for HTML output: 248 \begin{warpHTML}

Remember the original \RequirePackage:

249 \let\LWR@origRequirePackage\RequirePackage

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

250 \newcommand*\LWR@requirepackagenames{}

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

Note that argument 4 is passed directly to \StrBetween.

```
251 \newcommand*\LWR@findword[3][,]{%
252   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
253 }
```

\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.

254 \newcommand*\LWR@lookforpackagename}[1]{

Find the n'th package name from the list:

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

See if the package name was found:

```
256 \IfStrEq{\LWR@strresult}{}
257 {}% no filename
258 {}% yes filename
```

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

```
259 \IfFileExists{lwarp-\LWR@strresult.sty}
260 {% latex_html_ file found
261 \StrSubstitute
262 {\LWR@requirepackagenames}
263 {\LWR@strresult}
```

```

264 {lwarp-\LWR@strresult}{\LWR@requirepackagenames}
265 }
266 {}% no latex_html_* file
267 }% yes filename
268 }

```

`\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.

```

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

```

Redirect up to nine names:

```

270 \renewcommand*{\LWR@requirepackagenames}{#2}
271 \LWR@lookforpackagename{1}
272 \LWR@lookforpackagename{2}
273 \LWR@lookforpackagename{3}
274 \LWR@lookforpackagename{4}
275 \LWR@lookforpackagename{5}
276 \LWR@lookforpackagename{6}
277 \LWR@lookforpackagename{7}
278 \LWR@lookforpackagename{8}
279 \LWR@lookforpackagename{9}

```

`\RequirePackage` depending on the options and version:

```

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

```

`\LWR@ProvidesPackagePass` [*(pkgnam)*] [*(version)*]

Provides a new package which passes its options to the original.

```

293 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
294 \PackageInfo{lwarp}{Using the lwarp version of package ‘#1’, including options,}%

```

```

295 \IfValueTF{#2}
296 {\ProvidesPackage{lwarp-#1}[#2]}
297 {\ProvidesPackage{lwarp-#1}}
298 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
299 \ProcessOptions\relax
300
301 \IfValueTF{#2}
302 {\LWR@origRequirePackage{#1}[#2]}
303 {\LWR@origRequirePackage{#1}}
304 }

```

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

Provides a new package which drops/discards its options.

```

305 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
306 \PackageInfo{lwarp}{Using the lwarp version of package ‘#1’, discarding options,}%
307 \IfValueTF{#2}
308 {\ProvidesPackage{lwarp-#1}[#2]}
309 {\ProvidesPackage{lwarp-#1}}
310 \DeclareOption*{}
311 \ProcessOptions\relax
312 }

313 \end{warpHTML}

```

18 Copying a file

for HTML output: 314 \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.

```

315 \newcommand*{\LWR@copyfile}[2]{%
316 \newwrite\copyfile % open the file to write to
317 \immediate\openout\copyfile=#2
318 \newread\file % open the file to read from
319 \openin\file=#1
320 \begingroup\endlinechar=-1
321 \makeatletter
322 \loop\unless\ifeof\file
323 \read\file to\fileline % Read one line and store it into \fileline
324 % \fileline\par % print the content into the pdf

```



```

325 % print the content:
326 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
327 \repeat
328 \closeout\copyfile
329 \endgroup
330 }

331 \end{warpHTML}

```

19 Debugging messages

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

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

`\tracinglwarp` Turns on the debug tracing messages.

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

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

```

334 \newcommand{\LWR@traceinfo}[1]{%
335 \ifbool{LWR@tracinglwarp}%
336 {\PackageInfo{lwarp}{#1 : }}%
337 {}%
338 }

```

20 Remembering original formatting macros

for HTML output: 339 `\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.

```

340 \let\LWR@origtextit\textit
341 \let\LWR@origtextbf\textbf
342 \let\LWR@origtexttt\texttt
343 \let\LWR@origtextsc\textsc
344 \let\LWR@origtextsf\textsf
345 \let\LWR@origtextrm\textrm

```

```

346 \let\LWR@origbfseries\bfseries
347 \let\LWR@origrmfamily\rmfamily
348 \let\LWR@origttfamily\ttfamily
349 \let\LWR@orignormalfont\normalfont
350
351 \let\LWR@origraggedright\raggedright
352 \let\LWR@origonecolumn\onecolumn
353
354 \let\LWR@origtextsuperscript\textsuperscript
355 \let\LWR@origtextsubscript\textsubscript
356
357 \let\LWR@origscriptsize\scriptsize
358
359 \let\LWR@orignewpage\newpage
360
361 \let\LWR@origminipage\minipage
362 \let\LWR@origendminipage\endminipage
363
364 \let\LWR@orignewline\newline
365
366 \let\LWR@origitem\item
367
368 \let\LWR@origpar\par
369
370
371 \let\LWR@origfootnote\footnote
372 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
373
374 \let\LWR@origclearpage\clearpage
375 \let\clearpage\relax
376 \let\cleardoublepage\relax
377 \end{warpHTML}

```

21 Operating-System portability

lwarp tries to detect which operating system is being used.

Prog MS-Windows Prog Windows \warpOSwindows	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> .
---	--

for HTML & PRINT: 378 \begin{warppall}

21.1 Common portability code

Bool `usingOSwindows` Set if `\warpOSwindows`.

```
379 \newbool{usingOSwindows}
380 \boolfalse{usingOSwindows}
```

21.2 Unix, Linux, and Mac OS

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

```
381 \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.

```
382 \newcommand*{\warpOSwindows}
383 {
384 \booltrue{usingOSwindows}
385 \renewcommand*{\OSPathSymbol}{\@backslashchar}
386 }
```

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

```
387 \ifwindows
388 \warpOSwindows
389 \fi

390 \end{warpall}
```

22 Stacks

for HTML output: 391 \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@closedepthone`, etc.

22.1 Assigning depths

initial depths for empty stack entries:

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

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

```
393 \newcommand*{\LWR@depthfinished}{-4}
394 \newcommand*{\LWR@depthpart}{-1}
395 \newcommand*{\LWR@depthchapter}{0}
396 \newcommand*{\LWR@depthsection}{1}
397 \newcommand*{\LWR@depthsubsection}{2}
398 \newcommand*{\LWR@depthsubsubsection}{3}
399 \newcommand*{\LWR@depthparagraph}{4}
400 \newcommand*{\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

```
402 \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.

```
403 \newcommand*{\LWR@closeone}{}% top of the stack
```

```

404 \newcommand*{\LWR@closetwo}{}
405 \newcommand*{\LWR@closethree}{}
406 \newcommand*{\LWR@closefour}{}
407 \newcommand*{\LWR@closefive}{}
408 \newcommand*{\LWR@closesix}{}
409 \newcommand*{\LWR@closeseven}{}
410 \newcommand*{\LWR@closeeight}{}
411 \newcommand*{\LWR@closenine}{}
412 \newcommand*{\LWR@closeten}{}
413 \newcommand*{\LWR@closeeleven}{}
414 \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}

```

```

415 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
416 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
417 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
418 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
419 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
420 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
421 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
422 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
423 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
424 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
425 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
426 \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.

```

427 \NewDocumentCommand{\pushclose}{m m}
428 {
429 \let\LWR@closetwelve\LWR@closeeleven
430 \let\LWR@closeeleven\LWR@closeten
431 \let\LWR@closeten\LWR@closenine
432 \let\LWR@closenine\LWR@closeeight
433 \let\LWR@closeeight\LWR@closeseven
434 \let\LWR@closeseven\LWR@closesix
435 \let\LWR@closesix\LWR@closefive
436 \let\LWR@closefive\LWR@closefour
437 \let\LWR@closefour\LWR@closethree
438 \let\LWR@closethree\LWR@closetwo
439 \let\LWR@closetwo\LWR@closeone
440 \let\LWR@closeone#1
441 \let\LWR@closedepthtwelve\LWR@closedeptheleven
442 \let\LWR@closedeptheleven\LWR@closedephten
443 \let\LWR@closedephten\LWR@closedepthnine
444 \let\LWR@closedepthnine\LWR@closedeptheight
445 \let\LWR@closedeptheight\LWR@closedepthseven
446 \let\LWR@closedepthseven\LWR@closedepthsix
447 \let\LWR@closedepthsix\LWR@closedepthfive
448 \let\LWR@closedepthfive\LWR@closedepthfour
449 \let\LWR@closedepthfour\LWR@closedepththree
450 \let\LWR@closedepththree\LWR@closedepthtwo
451 \let\LWR@closedepthtwo\LWR@closedepthone
452 \let\LWR@closedepthone#2
453 }

```

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

```

454 \newcommand*{\popclose}
455 {
456 \let\LWR@closeone\LWR@closetwo
457 \let\LWR@closetwo\LWR@closethree
458 \let\LWR@closethree\LWR@closefour
459 \let\LWR@closefour\LWR@closefive
460 \let\LWR@closefive\LWR@closesix
461 \let\LWR@closesix\LWR@closeseven
462 \let\LWR@closeseven\LWR@closeeight
463 \let\LWR@closeeight\LWR@closenine
464 \let\LWR@closenine\LWR@closeten
465 \let\LWR@closeten\LWR@closeeleven
466 \let\LWR@closeeleven\LWR@closetwelve
467 \let\LWR@closedepthone\LWR@closedepthtwo
468 \let\LWR@closedepthtwo\LWR@closedepththree
469 \let\LWR@closedepththree\LWR@closedepthfour
470 \let\LWR@closedepthfour\LWR@closedepthfive

```

```

471 \let\LWR@closedepthfive\LWR@closedepthsix
472 \let\LWR@closedepthsix\LWR@closedepthseven
473 \let\LWR@closedepthseven\LWR@closedeptheight
474 \let\LWR@closedeptheight\LWR@closedepthnine
475 \let\LWR@closedepthnine\LWR@closedepthten
476 \let\LWR@closedepthten\LWR@closedeptheleven
477 \let\LWR@closedeptheleven\LWR@closedepthtwelve
478 }

479 \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: 480 \begin{warpHTML}

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

```

481 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
482 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
483 }

```

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

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

```
485 \end{warpHTML}
```

24 HTML entities

for HTML output: 486 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
487 \let\LWR@origampersand\&
```

```

\HTMLentity  {\entitytag}

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

\HTMLUnicode  {\hex_unicode}

489 \newcommand*{\HTMLUnicode}[1]{\HTMLentity{\#x#1}}

\&

490 \renewcommand*{\&}{\HTMLentity{amp}}

\textless
\textgreater
491 \let\LWR@origtextless\textless
492 \renewcommand*{\textless}{\HTMLentity{lt}}
493
494 \let\LWR@origtextgreater\textgreater
495 \renewcommand*{\textgreater}{\HTMLentity{gt}}

496 \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: 497 \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.

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

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

```
499 \providecommand*{\HTMLFileName}{}
```

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

```
500 \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.

```
501 \newcommand*{\SetHTMLFileNumber}[1]{%
502 \setcounter{LWR@htmlfilenumber}{#1}%
503 }
```

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

Defaults to use section names in the filenames.

```
504 \newbool{FileSectionNames}
505 \booltrue{FileSectionNames}
```

```
506 \end{warpall}
```

for HTML output: 507 `\begin{warpHTML}`

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

```
508 \newcounter{LWR@htmlfilenumber}
509 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
510 \newcommand*{\LWR@htmlsectionfilename}[1]{%
511 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
512 \LWR@traceinfo{about to assign temp}%
513 \edef\LWR@tempone{#1}%
514 \LWR@traceinfo{about to compare with ??}%
515 \ifthenelse{\equal{\LWR@tempone}{??}}{%
516 {%
517 \LWR@traceinfo{found ??}%
518 }{%
519 \LWR@traceinfo{not found ??}%
520 }%
```

```

521 \LWR@traceinfo{about to compare with zero or empty}%
522 \ifthenelse{%
523   \equal{\LWR@tempone}{0}%
524   \OR \equal{\LWR@tempone}{}}%
525   \OR \equal{\LWR@tempone}{??}%
526 }%
527 {%
528 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
529 \HomeHTMLFileName.html%
530 }%

```

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`:

```

531 {%
532 \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
533 \ifthenelse{%
534   \equal{\HTMLFileName}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
535 }%
536 {%
537 \LWR@traceinfo{prefixing the index name with an underscore.}%
538 \_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

539 {\HTMLFileName#1.html}%
540 }%
541 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
542 }

```

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

Prints the filename for the given label

```

543 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
544 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
545 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
546 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
547 }

548 \end{warpHTML}

```

26 Homepage link

for HTML output: 549 \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.

```
550 \newcommand*{\LinkHome}{%
551 \LWR@subhyperrefclass{%
552 \HomeHTMLFileName.html}%
553 {Home}{linkhome}}%
554 }
```

`\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.

```
555 \newcommand*{\LWR@topnavigation}{
556 \LWR@html-elementclassline{nav}{topnavigation}{\LinkHome}
557 }
```

`\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.

```
558 \newcommand*{\LWR@botnavigation}{
559 \LWR@html-elementclassline{nav}{botnavigation}{\LinkHome}
560 }
```

```
561 \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: 562 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```
563 \newcommand*{\PrintStack}{
564 \LWR@startpars
565 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
566 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
567 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
568 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
569 }

570 \end{warpHTML}
```

28 Closing stack levels

for HTML output: 571 \begin{warpHTML}

Close one nested level:

```
572 \newcommand*{\LWR@closeoneprevious}{%
573
574 \LWR@closeone{
575
576 \popclose{
577 }
```

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

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

Close any pending paragraph:

```
579 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
580 \whiledo{\not\(\LWR@closedepthone<\#1\)}{\LWR@closeoneprevious}
581 }
```

```
582 \end{warpHTML}
```

29 HTML tags, spans, divs, elements

for HTML output: 583 \begin{warpHTML}

29.1 Mapping L^AT_EX Sections to HTML Sections

```
584 \newcommand*{\LWR@tagpart}{h2}
585 \newcommand*{\LWR@tagpartend}{/h2}
586 \newcommand*{\LWR@tagchapter}{h3}
587 \newcommand*{\LWR@tagchapterend}{/h3}
588 \newcommand*{\LWR@tagsection}{h4}
589 \newcommand*{\LWR@tagsectionend}{/h4}
590 \newcommand*{\LWR@tagsubsection}{h5}
591 \newcommand*{\LWR@tagsubsectionend}{/h5}
592 \newcommand*{\LWR@tagsubsubsection}{h6}
```

```

593 \newcommand*{\LWR@tagssubsubsectionend}{/h6}
594 \newcommand*{\LWR@tagparagraph}{span class="paragraph">{}}
595 \newcommand*{\LWR@tagparagraphend}{/span}
596 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
597 \newcommand*{\LWR@tagsubparagraphend}{/span}
598
599 \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.

```

600
601 \newcommand*{\LWR@htmltagc}[1]{%
602 {%
603 \protect\LWR@origttfamily%
604 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
605 }%
606 }

```

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.

```

607 \newenvironment*{LWR@nestspan}
608 {%
609 \addtocounter{LWR@spandepth}{1}%
610 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
611 }%
612 {\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`.

```

613 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
614 \LWR@ensuredoingapar%
615 \LWR@htmltagc{#1}%
616 \begin{LWR@nestspan}%
617 #2%

```

```

618 \LWR@htmltagc{/#1}%
619 \end{LWR@nestspan}%
620 }

```

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

```

621 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
622 \LWR@ensuredoingapar%
623 \LWR@subhtml elementclass{span}{#1}{#2}%
624 \begin{LWR@nestspan}%
625 #3%
626 \LWR@htmltagc{/span}%
627 \end{LWR@nestspan}%
628 }

```

```
\LWR@htmltag {<tag>}
```

print an HTML tag: <tag>

```

629 \newcommand*{\LWR@htmltagb}[1]{%
630 \LWR@htmltagc{#1}%
631 \endgroup%
632 }
633
634 \newcommand*{\LWR@htmltag}{%
635 \begingroup\catcode'\_ =12
636 \LWR@htmltagb%
637 }

```

29.3 Block tags and comments

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

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

```

638 \newcommand*{\LWR@htmlopencomment}{%
639 % \LWR@origtexttt{\LWR@origtextbf{\LWR@origtextless{!}{-}{-}}}%
640 \begingroup\LWR@origttfamily\LWR@origtextless{!}{-}{-}\endgroup%
641 }
642
643 \newcommand*{\LWR@htmlclosecomment}{%
644 % \LWR@origtexttt{\LWR@origtextbf{{-}{-}\LWR@origtextgreater{}}}%
645 \begingroup\LWR@origttfamily{-}{-}\LWR@origtextgreater{}\endgroup%
646 }

```

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

```
647 \newcommand{\LWR@htmlcomment}[1]{%
648 \LWR@htmlopencomment{}}%
649 \LWR@origtextrm{#1}%
650 \LWR@htmlclosecomment{}}
```

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

```
651 \newcommand{\LWR@htmlblockcommentb}[1]
652 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
653
654 \newcommand{\LWR@htmlblockcomment}
655 {%
656 \begingroup\catcode'\_ =12%
657 \LWR@htmlblockcommentb%
658 }
```

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

```
659 \newcommand*{\LWR@htmlblocktag}[1]{%
660 \LWR@stoppars%
661 \LWR@htmltag{#1}%
662 \LWR@startpars%
663 }
```

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

```
664 \NewDocumentCommand{\LWR@htmlblocktagcomment}{m +m}{%
665 \LWR@stoppars%
666 \LWR@htmltag{#1}\LWR@htmlcomment{#2}\LWR@orignewline%
667 \LWR@startpars%
668 }
```

29.4 Div class and element class

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

Factored and reused in several places.

```
669 \NewDocumentCommand{\LWR@subhtmlclass}{m m o}{%
670 \IfValueTF{#3}%
671 {% option
672 \ifthenelse{\equal{#3}{}}{}%
673 {\LWR@htmltag{#1 class="#2"}}}% empty option
```

```

674 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
675 }% option
676 {\LWR@htmltag{#1 class="#2"}}% no option
677 }

```

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

```

678 \NewDocumentCommand{\LWR@htmlelementclass}{m m o}{%
679 \LWR@stoppars%
680 \LWR@subhtmlelementclass{#1}{#2}[#3]%
681 \LWR@startpars%
682 }

```

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

```

683 \newcommand*{\LWR@htmlelementclassend}[2]{%
684 \LWR@stoppars%
685 \LWR@htmltag{/#1}%
686 \LWR@htmlcomment{End of #1 ‘‘#2’’}%
687 \LWR@startpars%
688 }

```

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

```

689 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
690 \LWR@htmlelementclass{div}{#1}[#2]%
691 }

```

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

```

692 \newcommand*{\LWR@htmldivclassend}[1]{%
693 \LWR@htmlelementclassend{div}{#1}%
694 }

```

29.5 Single-line elements

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

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

```

695 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
696 \LWR@stoppars
697 \LWR@subhtmlelementclass{#1}{#2}[#3]%

```



```

698 #4%
699 \LWR@htmltag{/#1}
700 \LWR@startpars
701 }

```

29.6 HTML5 semantic elements

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

```

702 \newcommand*\LWR@htmlelement}[1]{%
703 \LWR@htmlblocktag{#1}
704 }

```

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

```

705 \newcommand*\LWR@htmlelementend}[1]{%
706 \LWR@stoppars
707 \LWR@htmltag{/#1}
708 \LWR@startpars
709 }
710
711 \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:

```

712 \begin{warpHTML}
713 \NewDocumentEnvironment{BlockClass}{m o}
714 {
715 \LWR@htmldivclass{#1}[#2]
716 }
717 {
718 \LWR@htmldivclassend{#1}
719 }
720 \end{warpHTML}

```

for PRINT output: 721 \begin{warpprint}
 722 \NewDocumentEnvironment{BlockClass}{m o}{-}{-}
 723 \end{warpprint}

\BlockClassSingle {(\class)} {(\text)} A single-line div, without a paragraph tag for the line of text.

for HTML output: 724 \begin{warpHTML}
 725 \newcommand{\BlockClassSingle}[2]{%
 726 \LWR@html@element@classline{div}{#1}{#2}%
 727 }
 728 \end{warpHTML}

for PRINT output: 729 \begin{warpprint}
 730 \newcommand{\BlockClassSingle}[2]{#2}
 731 \end{warpprint}

\InlineClass {(\class)} [(\style)] {(\text)} High-level interface for inline span classes.

for HTML output: 732 \begin{warpHTML}
 733 \NewDocumentCommand{\InlineClass}{m o m}{%
 734 \LWR@html@spanclass{#1}{#2}{#3}%
 735 }
 736 \end{warpHTML}

for PRINT output: 737 \begin{warpprint}
 738 \NewDocumentCommand{\InlineClass}{m o m}{#3}
 739 \end{warpprint}

29.8 Closing HTML tags

for HTML output: 740 \begin{warpHTML}

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

```
741 \newcommand*{\LWR@printclosepart}  

742   {\LWR@htmlcomment{Closing part}}  

743 \newcommand*{\LWR@printclosechapter}  

744   {\LWR@htmlcomment{Closing chapter}}  

745 \newcommand*{\LWR@printclosesection}  

746   {\LWR@htmlcomment{Closing section}}  

747 \newcommand*{\LWR@printclosesubsection}  

748   {\LWR@htmlcomment{Closing subsection}}  

749 \newcommand*{\LWR@printclosesubsubsection}  

750   {\LWR@htmlcomment{Closing subsubsection}}  

751 \newcommand*{\LWR@printcloseparagraph}
```

```

752     {\LWR@htmlcomment{Closing paragraph}}
753 \newcommand*{\LWR@printclosesubparagraph}
754     {\LWR@htmlcomment{Closing subparagraph}}

```

Lists require closing HTML tags:

```

755 \newcommand*{\LWR@printcloselistitem}
756     {\LWR@htmltag{/li}}
757 \newcommand*{\LWR@printclosedescitem}
758     {\LWR@htmltag{/dd}}
759 \newcommand*{\LWR@printcloseitemize}
760     {\LWR@htmltag{/ul}}
761 \newcommand*{\LWR@printcloseenumerate}
762     {\LWR@htmltag{/ol}}
763 \newcommand*{\LWR@printclosedescription}
764     {\LWR@htmltag{/dl}}

765 \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: 766 `\begin{warpHTML}`

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

```

767 \newcounter{LWR@spandepth}
768 \setcounter{LWR@spandepth}{0}

```

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

```

769 \newbool{LWR@doingstartpars}
770 \boolfalse{LWR@doingstartpars}

```

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

```

771 \newbool{LWR@doingapar}

```

```
772 \global\boolfalse{LWR@doingapar}
```

`\LWR@ensuredoinapar` 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:

```
773 \newcommand*{\LWR@ensuredoinapar}{%
774 \ifbool{LWR@doingstartpars}%
775 {\global\booltrue{LWR@doingapar}}%
776 {}%
777 }
```

`\LWR@openparagraph`

```
778 \newcommand*{\LWR@openparagraph}
779 {%
```

See if paragraph handling is enabled:

```
780 \ifbool{LWR@doingstartpars}%
781 {% handling pars
```

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

```
782 \ifthenelse{%
783 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
784 \cnttest{\value{LWR@spandepth}}{>}{0}%
785 }% 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`.

```
786 {}% no nested par tags
```

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

```
787 {% yes nest par tags
788 \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Manually indent item list labels to avoid left margin intrusion:

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

See if are nested inside an item list:

```

789 \ifnumcomp{\@listdepth}{>}{0}%
790 {%

```

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

```

791 \LWR@orighspace{1in}%
792 }{}%

```

Now have started a paragraph.

```

793 \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.)

```

794 \let\par\LWR@closeparagraph%
795 }% end of yes nest par tags
796 }% end of handling pars
797 {}% not handling pars
798 }

```

\LWR@closeparagraph

```

799 \newcommand*{\LWR@closeparagraph}
800 {%

```

See if paragraph handling is enabled:

```

801 \ifbool{LWR@doingapar}%

```

If currently in paragraph mode:

```

802 {% handling pars

```

See if already started a lateximage or a span:

```

803 \ifthenelse{%
804 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
805 \cnttest{\value{LWR@spandepth}}{>}{0}%
806 }%

```

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

```

807 {% no nested par tags
808 \ifthenelse{%
809 \cnttest{\value{LWR@spandepth}}{>}{0}%

```

```

810 \AND%
811 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
812 }%
813 {%
814 \ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}%
815 }%
816 {}%
817 }% no nested par tags

```

If have not already started a lateximage or a span:

```
818 {% yes nest par tags
```

Print a closing tag:

```

819 \unskip%
820 \LWR@htmltagc{/LWR@tagregularparagraph}%

```

No longer doing a paragraph:

```

821 \global\boolfalse{LWR@doingapar}%
822 % Disable the special \env{minipage} \& \cs{hspace} interaction
823 % until a new minipage is found:
824 %   \begin{macrocode}
825 \global\boolfalse{LWR@minipagethispar}%
826 }% end of yes nest par tags
827 }% end of handling pars

```

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

```

828 {% not handling pars
829 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
830 {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
831 {}%
832 }% not handling pars

```

Finish with regular paragraph processing

```

833 \LWR@origpar%
834 }

```

```
835 \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: 836 `\begin{warpHTML}`

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

```
837 \newcommand*{\LWR@startpars}%
838 {%
```

See if currently handling HTML paragraphs:

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

If already in paragraph mode, do nothing.

```
840 {}%
```

If not currently in paragraph mode:

```
841 {%
```

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

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

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

```
843 \let\par\LWR@closeparagraph
844
845 }% an intentionally blank line
```

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

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

No `<par>` tag yet to undo:

```
847 \global\boolfalse{LWR@doingapar}%
848 }
```

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

```
849 \newcommand*{\LWR@stoppars}%  
850 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
852 {%
```

Print a closing tag:

```
853 \unskip%  
854 \LWR@htmltagc{/ \LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

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

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

```
856 \global\boolfalse{\LWR@minipagethispar}  
857  
858 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
859 {}%
```

See if currently allowing HTML paragraphs:

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

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

```
861 {%  
862 \ClearPreHook{par}%  
863 }%
```

Else: do nothing

```
864 {}%
```

no longer in paragraph mode

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


no <p> tag to undo:

```
866 \global\boolfalse{LWR@doingapar}%
867 }

868 \end{warpHTML}
```

32 Page headers and footers

for HTML & PRINT: 869 \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.

```
870 \newcommand{\LWR@firstpagetop}{} % for the home page alone
871 \newcommand{\LWR@pagetop}{} % for all other pages
872 \newcommand{\LWR@pagebottom}{}
873
874 \newcommand{\LWR@setfirstpagetopb}[1]{%
875 \renewcommand{\LWR@firstpagetop}{#1}
876 \catcode'\_ =8
877 }
```

\SetFirstPageTop {*{text and logos}*}

```
878 \newcommand{\SetFirstPageTop}{%
879 \catcode'\_ =12
880 \LWR@setfirstpagetopb
881 }

882 \newcommand{\LWR@setpagetopb}[1]{%
883 \renewcommand{\LWR@pagetop}{#1}
884 \catcode'\_ =8
885 }
```

\SetPageTop {*{text and logos}*}

```
886 \newcommand{\SetPageTop}{%
887 \catcode'\_ =12
888 \LWR@setpagetopb
889 }

890 \newcommand{\LWR@setpagebottomb}[1]{%
891 \renewcommand{\LWR@pagebottom}{#1}
892 \catcode'\_ =8
893 }
```

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

```
894 \newcommand{\SetPageBottom}{%
895 \catcode'\_ =12
896 \LWR@setpagebottomb
897 }

898 \end{warpall}
```

33 CSS

for HTML output: 899 `\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.

```
900 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

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

```
901 \newcommand*{\LWR@newcssb}[1]{%
902 \renewcommand*{\LWR@currentcss}{#1}
903 \catcode'\_ =8
904 }
905
906 \newcommand*{\NewCSS}{
907 \catcode'\_ =12
908 \LWR@newcssb
909 }
910 \end{warpHTML}
```

for PRINT output: 911 `\begin{warpprint}`
 912 `\newcommand*{\NewCSS}[1]{}`
 913 `\end{warpprint}`

34 HTML meta description

for HTML output: 914 `\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.

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

```

`\NewHTMLdescription` `{\langle New HTML meta description.\rangle}`

Assigns the HTML file's description meta tag

```
916 \newcommand{\NewHTMLdescription}[1]{%
917 \renewcommand{\LWR@currentHTMLdescription}{#1}
918 }
919
920 \end{warpHTML}

```

for PRINT output: `921 \begin{warpprint}`
`922 \newcommand{\NewHTMLdescription}[1]{}`
`923 \end{warpprint}`

35 Footnotes

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 L^AT_EX 2_ε printed documents, footnotes are placed at the bottom of the printed page. For HTML, footnotes will be replaced by page notes using the `pagenote` package. Footnotes will then appear at the bottom of the HTML 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: 924 \begin{warpHTML}

Delayed in case the user's preamble loaded and adjusted pagenote:

925 \AtEndPreamble{

\notedivision Prints the footnote heading at the end of the page.

926 \renewcommand*\notedivision{

927 {\BlockClassSingle{footnoteheader}{}}

To print the word “Notes:” at the top of the footnotes:

\renewcommand*\notedivision| \\
 {\BlockClassSingle{footnoteheader}{Notes:}}

\pagenotesubhead {\langle num \rangle} {\langle title \rangle}

Don't print page note headings in HTML page notes.

928 \renewcommand{\pagenotesubhead}[2]{}

\footnote [\langle id \rangle] {\langle text \rangle} Footnotes now become page notes.

929 \let\footnote\LWR@origpagenote

\notenuminnotes {\langle number \rangle}

Sets the page notes to use superscripts to match standard classes.

930 \renewcommand{\notenuminnotes}[1]{%

931 #1 }

\noteentry {\langle 1: note number \rangle} {\langle 2: id \rangle} {\langle 3: text \rangle} {\langle 4: pagenum \rangle}

```

932 \let\LWR@orignoteentry\noteentry
933 \renewcommand{\noteentry}[4]{%

```

`\@currentlabel` A label may be used inside a note, so the proper `\@currentlabel` must be set
`\cref@currentlabel` inside a group. Likewise for `cleveref`'s `\cref@currentlabel`.

```

934 \begingroup%
935 \renewcommand*\@currentlabel{#1}%
936 \renewcommand*\cref@currentlabel{[pagenote] [1] [] {#1}}%

```

Re-use `pageref`'s code, with an extra `endgroup`:

```

937 \LWR@orignoteentry{#1}{#2}{#3}{#4}%
938 \endgroup%
939 }

```

`\LWR@printpendingnotes` Prints the page notes at the end of the section.

```

940 \newcommand*\LWR@printpendingnotes>{

```

If there are footnotes to print, open a new `div` and print the notes.

```

941 \ifthenelse{\cnttest{\value{pagenote}}{>}{0}}{%
942 {
943 \begin{BlockClass}{footnotes}

```

Print any pending endnotes, allowing new endnotes to accumulate.

```

944 \LWR@origprintnotes*

```

Start over with new page notes.

```

945 \setcounter{pagenote}{0}
946 \end{BlockClass}
947 }
948 {}
949 }

```

```

950 }% \AtEndPreamble

```

```

951 \end{warpHTML}

```

36 Marginpars

for HTML output: 952 \begin{warpHTML}

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

```
953 \renewcommand{\marginpar}[2] [] {%
954 \LWR@htmlspanclass{marginpar}{#2}%
955 }

956 \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: 957 `\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.

```
958 \newcounter{FileDepth}
959 \setcounter{FileDepth}{-5}
```

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

```
960 \newbool{CombineHigherDepths}
961 \booltrue{CombineHigherDepths}

962 \end{warpall}
```

for HTML output: 963 `\begin{warpHTML}`

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

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

`\LWR@thisnewfilename` The filename being sanitized.

```
965 \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.

```
966 \newcommand*\LWR@filenamenoblanks}[1]{%
967 \begingroup
```

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

```
968 \renewcommand*\HTMLUnicode}[1]{
969 \renewcommand*\HTMLentity}[1]{##1}
970 \renewcommand*\LWR@htmltagc}[1]{
971 \DeclareExpandableDocumentCommand\InlineClass}{m o m}{##3}
```

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

```
972 \renewcommand*\&{and}
973 \renewcommand\textit}[1]{##1}
974 \renewcommand\textsc}[1]{##1}
975 \renewcommand\textsl}[1]{##1}
976 \renewcommand\textbf}[1]{##1}
977 \renewcommand\texttt}[1]{##1}
978 \renewcommand\textsf}[1]{##1}
979 \renewcommand\textrm}[1]{##1}
980 \renewcommand\textsuperscript}[1]{##1}
981 \renewcommand\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
982 \edef\LWR@thisnewfilename{#1}
983 \fullexpandarg
```

Convert spaces into hyphens:

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

Convert punctuation into hyphens:

```
985 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
986 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
987 \StrSubstitute{\LWR@thisnewfilename}%
988 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
989 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
```

```

990 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
991 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
992 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
993 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
994 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
995 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
996 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
997 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
998 \StrSubstitute{\LWR@thisnewfilename}%
999 {\textless}{-}[\LWR@thisnewfilename]
1000 \StrSubstitute{\LWR@thisnewfilename}%
1001 {\textgreater}{-}[\LWR@thisnewfilename]
1002 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
1003 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
1004 \StrSubstitute{\LWR@thisnewfilename}{\&}{-}[\LWR@thisnewfilename]
1005 \StrSubstitute{\LWR@thisnewfilename}{\|}{-}[\LWR@thisnewfilename]
1006 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
1007 \StrSubstitute{\LWR@thisnewfilename}%
1008 {\textbackslash}{-}[\LWR@thisnewfilename]
1009 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
1010 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
1011 %      "~{}" for babel
1012 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
1013 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
1014 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

1015 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
1016 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
1017 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
1018 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
1019 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
1020 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
1021 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
1022 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
1023 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
1024 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
1025 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
1026 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
1027 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
1028 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
1029 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
1030 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1031 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
1032 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

1033 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]

```



```

1034 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
1035 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
1036 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
1037 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1038 %      emdash
1039 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1040 %      emdash
1041 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
1042 \endgroup
1043 }

```

`\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.

```

1044 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```

1045 \LWR@htmlend{section}{tbody}
1046
1047 \LWR@printpendingnotes
1048
1049 \LWR@htmlend{footer}
1050
1051 \LWR@pagebottom
1052
1053 \LWR@htmlend{footer}

```

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

```

1054 \ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{

```

End of this HTML file:

```

1055 \LWR@stoppars
1056 \LWR@htmltag{/body}\LWR@orignewline
1057 \LWR@htmltag{/html}\LWR@orignewline
1058
1059 \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.

```

1060 \ifbool{FileSectionNames}%
1061 {\LWR@filenamoblanks{#1}}
1062 {\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.

```

1063 \LWR@htmlblockcomment{%
1064 \LWR@origtextttt{|Start file|}%
1065 \LWR@htmlsectionfilename{\LWR@thisfilename}}}%
1066 }

```

At the top of the starting file:

```

1067 \LWR@stoppars
1068
1069 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
1070
1071 \LWR@topnavigation
1072
1073 \LWR@html element{header}
1074
1075 \LWR@pagetop
1076
1077 \LWR@html elementend{header}
1078
1079 \ifcvoid{thetitle}{ }\LWR@printthetitle}
1080
1081 \LWR@sidetoc
1082
1083 \LWR@html elementclass{section}{textbody}
1084
1085 \LWR@stoppars
1086
1087 \setcounter{LWR@latestautopage}{\value{page}}
1088 }
1089 \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_YLaTeX and LuaLaTeX directly support accented section and file names.

for HTML output: 1090 `\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.



```
1091 \newbool{LWR@mainmatter}
1092 \DeclareDocumentCommand{\mainmatter}{}{%
1093 \booltrue{LWR@mainmatter}%
1094 }
```

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

```
1095 \DeclareDocumentCommand{\frontmatter}{}{%
1096 \boolfalse{LWR@mainmatter}%
1097 }
```

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

```
1098 \DeclareDocumentCommand{\backmatter}{}{%
1099 \boolfalse{LWR@mainmatter}
1100 }
```

38.2 Sectioning support macros

`\LWR@sectionnumber` `{\section type}`

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

```
1101 \newcommand*{\LWR@sectionnumber}[1]{%
1102 \InlineClass{sectionnumber}{#1} %
1103 }
```


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

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

Create an autosection tag.

```
1104 \newcommand*\LWR@createautosec[1]{%
1105 \LWR@htmltag{#1 id="autosec-\thepage"{}}%
1106 }
```


`\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.

```
1107 \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.

```
1108 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1109 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1110 \LWR@pushoneclose{#1}{#2}%
1111 }
```

`\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.

```
1112 \newcounter{LWR@prevFileDepth}
1113 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

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

The common actions for the high-level sectioning commands.

```
1114 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
1115 \LWR@traceinfo{LWR@section}%
1116 \LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
1117 \global\boolfalse{LWR@minipagethispar}%
```

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

```
1118 \LWR@orignewpage%
1119
```

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

```
1120 \LWR@traceinfo{LWR@section: testing about to start a new HTML file}%
1121 \IfBooleanTF{#1}{\% not starred
1122 \ifthenelse{%
1123 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
1124 \AND%
1125 \(%
1126 \NOT\boolean{CombineHigherDepths}\OR%
1127 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
1128 \)%
1129 }%
```

If so: start a new HTML file:

```
1130 {% new file
1131 \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
1132 \IfNoValueTF{#2}%
```

If no optional entry

```
1133 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
1134 {\LWR@newhtmlfile{#2}}%
1135 }% new file
```

Else: No new HTML file:

```
1136 {}% not new file
1137 }% not starred
```

Remember this section's name for `\nameref`:

```

1138 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
1139 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%

```

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

```

1140
1141 \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}
1142

```

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

```

1143 \ifthenelse{%
1144 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
1145 }%
1146 {\LWR@startpars}
1147 {}

```

Create the opening tag with an autosec:

```

1148 \LWR@createautosec{\csuse{LWR@tag#4}}%

```

If not starred, step counter and add to TOC:

```

1149 \IfBooleanTF{#1}%
1150 {}% starred
1151 {% not starred

```

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

```

1152 \ifthenelse{%
1153 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1154 }%
1155 {% if secnumdepth

```

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

```

1156 \LWR@traceinfo{LWR@section: about to test main matter}%
1157 \ifbool{LWR@mainmatter}%
1158 {%
1159 \LWR@traceinfo{LWR@section: yes mainmatter}%
1160 \refstepcounter{#4}%

```

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

```

1161 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1162 \addcontentsline{toc}{#4}%
1163 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1164 \IfValueTF{#2}{#2}{#3}%

```

```

1165 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1166 }% end of if main matter

```

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

```

1167 {% not main matter
1168 \LWR@traceinfo{LWR@section: no main matter}%
1169 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1170 }% end of not main matter
1171 }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

1172 {%
1173 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1174 }%

```

For part, print the section type:

```

1175 \ifbool{LWR@mainmatter}%
1176 {%
1177 \ifthenelse{%
1178 \(\cnttest{\csuse{LWR@depth#4}}{<=}}%
1179 {\value{secnumdepth}}\)\ AND
1180 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
1181 }%
1182 {\csuse{#4name}~{}}%
1183 {}%

```

Print the section number:

```

1184 \LWR@traceinfo{LWR@section: about to print section number}%
1185 \ifthenelse{%
1186 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1187 }%
1188 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1189 {}%
1190 \LWR@traceinfo{LWR@section: finished print section number}%
1191 }{}%
1192 }% end of not starred

```

Print the section name:

```

1193 #3

```

close the heading tag, such as /H2

```

1194 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a L^AT_EX label:

```
1195 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
1196 \ifthenelse{%
1197 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%
1198 {\LWR@startpars}%
1199 {}%
```

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.

```
1200 \IfBooleanTF{#1}{}{% not starred
1201 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
1202 }% not starred
1203 \LWR@traceinfo{LWR@section: done}%
1204 }
```

38.3 \section and friends

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

```
1205 \@ifundefined{chapter}
1206 {}
1207 {%
1208 \DeclareDocumentCommand{\part}{s o m}{%
1209 \LWR@stoppars%
1210
1211 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
1212
1213 \LWR@section{#1}{#2}{#3}{part}%
1214 }
1215 }
```

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

```
1216 \@ifundefined{chapter}
1217 {}
1218 {%
```



```

1219 \DeclareDocumentCommand{\chapter}{s o m}{%
1220 \LWR@traceinfo{chapter #3}%
1221 \LWR@stoppars%
1222 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
1223
1224 \LWR@traceinfo{chapter: about to LWR@section}%
1225 \LWR@section{#1}{#2}{#3}{chapter}%
1226 \LWR@traceinfo{chapter: done}%
1227 }
1228 }

```

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

```

1229 \DeclareDocumentCommand{\section}{s o m}{%
1230 \LWR@stoppars%
1231
1232 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
1233
1234 \LWR@section{#1}{#2}{#3}{section}%
1235 }

```

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

```

1236 \DeclareDocumentCommand{\subsection}{s o m}{%
1237 \LWR@stoppars%
1238
1239 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1240
1241 \LWR@section{#1}{#2}{#3}{subsection}%
1242 }

```

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

```

1243 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1244 \LWR@stoppars%
1245
1246 \LWR@startnewdepth{\LWR@depthsubsubsection}%
1247 {\LWR@printclosesubsubsection}%
1248
1249 \LWR@section{#1}{#2}{#3}{subsubsection}%
1250 }

```

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

```

1251 \DeclareDocumentCommand{\paragraph}{s o m}{%
1252 \LWR@stoppars%
1253

```

```

1254 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1255
1256 \LWR@section{#1}{#2}{#3}{paragraph}%
1257 }

```

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

```

1258 \DeclareDocumentCommand{\subparagraph}{s o m}{%
1259 \LWR@stoppars%
1260
1261 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
1262
1263 \LWR@section{#1}{#2}{#3}{subparagraph}%
1264 }

1265 \end{warpHTML}

```

39 Starting a new file

for HTML output: 1266 \begin{warpHTML}

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

```
1267 \newcommand*{\MetaLanguage}{en-US}
```

`\LWR@filestart` {*title__suffix*}

Creates the opening HTML tags.

```
1268 \newcommand*{\LWR@filestart}[1]{
```

Locally temporarily disable direct-formatting commands:

```

1269 \begingroup
1270 \renewcommand{\textit}[1]{##1}% not used in filenames
1271 \renewcommand{\textsc}[1]{##1}
1272 \renewcommand{\textsl}[1]{##1}
1273 \renewcommand{\textbf}[1]{##1}
1274 \renewcommand{\texttt}[1]{##1}
1275 \renewcommand{\textsf}[1]{##1}
1276 \renewcommand{\textrm}[1]{##1}
1277 \renewcommand{\textsuperscript}[1]{##1}
1278 \renewcommand{\textsubscript}[1]{##1}
1279 \renewcommand*{\HTMLUnicode}[1]{

```

```

1280 \renewcommand*{\HTMLentity}[1]{}
1281 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1282 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}

```

Create the page's HTML header:

```
1283 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
1284 \LWR@htmltag{html lang="\MetaLanguage"{}}\LWR@orignewline
```

Start of the meta data:

```
1285 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
1286 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

lwarp is the generator:

```

1287 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
1288     \LWR@orignewline

```

If there is a description, add it now:

```

1289 \ifdefempty{\LWR@currentHTMLdescription}{-}{-}{%
1290 \LWR@htmltag{%
1291 meta name="description" content="\LWR@currentHTMLdescription" /}%
1292     \LWR@orignewline
1293 }%

```

Mobile-friendly viewport:

```

1294 \LWR@htmltag{meta name="viewport" %
1295 content="width=device-width, initial-scale=1.0" /}%
1296     \LWR@orignewline

```

IE patch:

```

1297 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
1298 \LWR@htmltag{%
1299 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}}%
1300 \LWR@htmltag{/script}\LWR@orignewline
1301 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline

```

The page's title:

```

1302 \ifcvoid{thetitle}{}{%
1303 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
1304 }%

```

The page's stylesheet:

```

1305 \LWR@htmltag{%
1306 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1307 \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.

```

1308 \ifbool{mathjax}%
1309 {%
1310 \boolfalse{LWR@verbtags}
1311 \VerbatimInput{lwarp_mathjax.txt}%
1312 \booltrue{LWR@verbtags}
1313 \LWR@stoppars
1314 }% end of mathjax
1315 {%

```

End of the header:

```

1316 \LWR@htmltag{/head}\LWR@orignewline

```

Start of the body:

```

1317 \LWR@htmltag{body}\LWR@orignewline
1318 \endgroup
1319 }

```

```

1320 \end{warpHTML}

```

40 Starting HTML output

for HTML output: 1321 \begin{warpHTML}

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

```

1322 \newcommand*{\LWR@LwarpStart}
1323 {%

```

Expand and detokenize \HomeHTMLFileName and \HTMLFileName:

```
1324 \edef\LWR@strresult{\HomeHTMLFileName}
1325 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1326 \edef\LWR@strresult{\HTMLFileName}
1327 \edef\HTMLFileName{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
1328 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
1329 \LWR@origscriptsize%
```

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

```
1330 \LWR@origraggedright%
1331 \let\\\LWR@endoffline%
```

Spread the lines for pdftotext to read them well:

```
1332 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
1333 \setlength{\parindent}{0pt}
1334 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
1335 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1336 \setlength{\belowcaptionskip}{-3ex}
```

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

```
1337 \renewcommand{\ps@plain}{}%
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 1338 \renewcommand*\centering{}
1339 \renewcommand*\raggedleft{}
1340 \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:

```
1341 \let\LWR@origtabular\tabular
1342 \let\LWR@origendtabular\endtabular
1343 \let\tabular\LWR@tabular
1344 \let\endtabular\endLWR@tabular
```

Float captions:

```
1345 \let\LWR@origcaption\caption
```

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

```
1346 \let\LWR@origltx@label\ltx@label
1347 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
1348 \let\TeX\LWR@TeX
1349 \let\LaTeX\LWR@LaTeX
1350 \let\LuaTeX\LWR@LuaTeX
1351 \let\LuaLaTeX\LWR@LuaLaTeX
1352 \let\XeTeX\LWR@XeTeX
1353 \let\XeLaTeX\LWR@XeLaTeX
1354 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
1355 \let\rotatebox\LWR@rotatebox
1356 \let\scalebox\LWR@scalebox
1357 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
1358 \global\boolfalse{LWR@doingapar}
1359 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
1360 \LWR@filestart{}
1361 \LWR@htmltag{header}\LWR@orignewline
1362 \LWR@startpars
1363 \LWR@firstpagetop
1364 \LWR@stoppars
1365 \LWR@htmltag{/header}\LWR@orignewline
1366 \LWR@htmltag{section class="textbody"{} }
1367 \LWR@origpagestyle{empty}
```

Document and page settings:

```
1368 \mainmatter
1369 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section 42.6.

```
1370 \if@titlepage
1371     \thanksmarkseries{arabic}
1372 \else
1373     \thanksmarkseries{fnsymbol}
1374 \fi
```

Initial default patch for fancyvrb:

```
1375 \fvset{frame=none}%
```

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

```
1376 \catcode'\&=\active
```

Allow HTML paragraphs to begin:

```
1377 \LWR@startpars
1378 }
```

```
1379 \end{warpHTML}
```

41 Ending HTML output

for HTML output: 1380 \begin{warpHTML}

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

```
1381 \newcommand*\LWR@requesttoc[2]{%
1382 \ifbool{#1}
1383 {
1384 \expandafter\newwrite\csuse{tf@#2}
1385 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
1386 }{}
1387 }
```

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

```
1388 \newcommand*{\LWR@LwarpEnd}  
1389 {  
1390 \LWR@stoppars  
1391 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
1392 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
1393 \LWR@printpendingnotes
```

Create the footer:

```
1394 \LWR@htmlelement{footer}  
1395  
1396 \LWR@pagebottom  
1397  
1398 \LWR@htmlelementend{footer}
```

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

```
1399 \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

```
1400 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
1401 \LWR@htmltag{/body}\LWR@orignewline  
1402 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
1403 \LWR@orignewpage
```

For lateximage commands:

```
1404 \immediate\closeout\LWR@file  
1405 }
```

```
1406 \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:

<code>\maketitlehookaa</code>	<code>\maketitlehookaa</code> : Between “published” and the title.
<code>\maketitlehookaaa</code>	<code>\maketitlehookaaa</code> : Between the title and the subtitle.
<code>\prepublished</code>	<code>\prepublished</code> : Before the “published” field.
<code>\postpublished</code>	<code>\postpublished</code> : After the “published” field.
<code>\presubtitle</code>	<code>\presubtitle</code> : Before the subtitle.
<code>\postsubtitle</code>	<code>\postsubtitle</code> : After the subtitle.
<code>\printthanks</code>	<code>\printthanks</code> has been added to force the printing of thanks inside a <code>titlingpage</code> environment when <code>\maketitle</code> is not used.



No footnotes!

Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` for footnotes, do not use `\footnote`.

At the end of the `titlingpage` environment, footnote marks are forced to reset to zero.

Inside a `titlingpage` environment with the `article` document class, thanks marks will be `fnsymbol` instead of `arabic`. `arabic` is usually used when inside `titlepage` environments where the title page is on its own page, but is not automatically used inside a `titlingpage` environment.

To force the thanks marks to be `arabic`:

```

\begin{titlingpage}
\thanksmarkseries{arabic}
...

```

42.1 Setting the title, etc.

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

<pre> \published \title \subtitle \author \date </pre>	<pre> \@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. </pre>
<pre> \printpublished \printtitle \printsubtitle \printauthor \printdate </pre>	<pre> \printtitle, \printsubtitle, 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. </pre>
<pre> \thepublished \thetitle \thesubtitle \theauthor \thedate </pre>	<pre> \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: </pre>
<pre> \SetPageBottom </pre>	<pre> {\text} </pre>

```

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

```

<pre> \author \and </pre>	<pre> {\author} While using \maketitle, the author is treated as a single-column table and the \and feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as </pre>
---------------------------	--

```

\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}

```

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors

names separated by spaces. This fails when affiliations are included on their own table rows.

A solution, provide here, is to define a macro `\affiliation` which during `\maketitle` starts a new table row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to throw away its argument, thus printing only the author names when `\author` is later used inline.

```
\affiliation {{text}}
```

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

```
for HTML output: 1407 \begin{warpHTML}
                  1408 \newrobustcmd{\affiliation}[1]{\ \InlineClass{affiliation}{#1}}
                  1409 \end{warpHTML}
```

```
for PRINT output: 1410 \begin{warpprint}
                  1411 \newrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
                  1412 \end{warpprint}
```

The following are based on the original titling code:

```
for HTML & PRINT: 1413 \begin{warpall}
```

```
\author {{text}}
```

Redefined to nullify `\affiliation`, etc. before printing the authors inline.

`\@author` retains the entire author with its `\thanks`, while `\theauthor` will have `\thanks` removed and `\and` simplified.

```
1414 \renewcommand{\author}[1]{%
1415 \gdef\@author{#1}
1416 \begingroup
1417   \renewcommand{\thanks}[1]{ }
1418   \renewcommand{\and}{\unskip, }
1419   \renewcommand{\thanksmark}[1]{ }
1420   \renewcommand{\thanksgap}[1]{ }
1421   \renewcommand{\affiliation}[1]{ }
1422   \protected@xdef\theauthor{#1}
1423 \endgroup}
```

```
\published {{text}}
```

```
1424 \newcommand{\published}[1]{%
1425 \gdef\@published{#1}
```

```

1426 \begingroup
1427   \renewcommand{\thanks}[1]{%
1428     \renewcommand{\thanksmark}[1]{%
1429       \renewcommand{\thanksgap}[1]{%
1430         \protected@xdef\thepublished{#1}
1431       }
1432     }
1433   \newcommand{\@published}{}
1434   \newcommand{\thepublished}{}

\subtitle {(\text)}

1435 \newcommand{\subtitle}[1]{%
1436   \gdef\@subtitle{#1}
1437   \begingroup
1438     \renewcommand{\thanks}[1]{%
1439       \renewcommand{\thanksmark}[1]{%
1440         \renewcommand{\thanksgap}[1]{%
1441           \protected@xdef\thesubtitle{#1}
1442         }
1443       }
1444     \newcommand{\@subtitle}{}
1445     \newcommand{\thesubtitle}{}

1446 \end{warpall}

```

42.2 Changes to HTML titlepage and titlingpage

for HTML output: 1447 \begin{warpHTML}

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

```

1448 \renewenvironment*{titlepage}
1449   {\BlockClass{titlepage}\LWR@subminipage}
1450   {\LWR@endsubminipage\endBlockClass}

```

Env **titlingpage**

```

1451 \renewenvironment*{titlingpage}
1452 {%

```

Start an HTML titlepage div:

```

1453 \begin{titlepage}

```

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

```
1454 \LWR@maketitlesetup
1455 \let\maketitle\LWR@titlingmaketitle
1456 }
1457 {
```

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

```
1458 \end{titlepage}
```

Reset the footnote counter:

```
1459 \@bscontmark
1460 }
```

```
1461 \end{warpHTML}
```

for HTML & PRINT: 1462 `\begin{warpall}`

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

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

```
1463 \newcommand*{\printthanks}{\@thanks}
```

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

```
1464 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1465 \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: 1466 `\begin{warpHTML}`

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

```
1467 \newcommand{\prepublished}[1]{%
1468 \def\@bspublished{\BlockClass{published}#1}%
1469 }
1470
```

```
1471 \newcommand{\postpublished}[1]{%
1472 \def\@bspostpublished{#1\endBlockClass}%
1473 }
1474
1475 \renewcommand{\pretitle}[1]{%
1476 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}}%
1477 }
1478
1479 \renewcommand{\posttitle}[1]{%
1480 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
1481 }
1482
1483 \newcommand{\presubtitle}[1]{%
1484 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
1485 }
1486
1487 \newcommand{\postsubtitle}[1]{%
1488 \def\@bspostsubtitle{#1\endBlockClass}%
1489 }
1490
1491 \renewcommand{\preauthor}[1]{%
1492 \def\@bspreauthor{\BlockClass{author}#1}%
1493 }
1494
1495 \renewcommand{\postauthor}[1]{%
1496 \def\@bspostauthor{#1\endBlockClass}%
1497 }
1498
1499 \renewcommand{\predate}[1]{%
1500 \def\@bspredate{#1\BlockClass{titledate}}}%
1501 }
1502
1503 \renewcommand{\postdate}[1]{%
1504 \def\@bspostdate{\endBlockClass#1}%
1505 }
1506
1507 \prepublished{\begin{center}}
1508 \postpublished{\par\end{center}}
1509
1510 \pretitle{\begin{center}}
1511 \posttitle{\par\end{center}}
1512
1513 \presubtitle{\begin{center}}
1514 \postsubtitle{\par\end{center}}
1515
1516 \preauthor{\begin{center}}%
1517 \begin{tabular}[t]{c}%
1518 }
1519 \postauthor{\end{tabular}\par\end{center}}
1520
```

```

1521 \predatex{\begin{center}}
1522 \postdatex{\par\end{center}}

```

\printpublished

```

1523 \newcommand*\printpublished{
1524 \ifthenelse{\equal{\thepublished}{}}
1525 {}
1526 {
1527 \begin{BlockClass}{published}
1528 \@published
1529 \end{BlockClass}
1530 }
1531 }

```

\printtitle

```

1532 \newcommand*\printtitle
1533 {
1534 \LWR@stoppars
1535 \LWR@htmltag{h1}%
1536 \@title%
1537 \LWR@htmltag{/h1}
1538 \LWR@startpars
1539 }

```

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

```

1540 \newcommand*\LWR@printthetitle
1541 {
1542 \LWR@stoppars
1543 \LWR@htmltag{h1}%
1544 \thetitle%
1545 \LWR@htmltag{/h1}
1546 \LWR@startpars
1547 }

```

\printsubtitle

```

1548 \newcommand*\printsubtitle{
1549 \ifthenelse{\equal{\thesubtitle}{}}
1550 {}
1551 {
1552 \begin{BlockClass}{subtitle}
1553 \@subtitle
1554 \end{BlockClass}

```

```
1555 }
1556 }
```

`\printauthor`

```
1557 \newcommand*{\printauthor}{
1558 \begin{BlockClass}{author}
1559 \begin{tabular}{c}\@author\end{tabular}
1560 \end{BlockClass}
1561 }
```

`\printdate`

```
1562 \newcommand*{\printdate}{%
1563 \begin{BlockClass}{titledate}
1564 \@date
1565 \end{BlockClass}
1566 }
```

```
1567 \end{wrapHTML}
```

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: 1568 `\begin{wrapprint}`

`\printpublished`

```
1569 \newcommand*{\printpublished}{\Large\scshape\@published}
```

`\printtitle`

```
1570 \newcommand*{\printtitle}{\Huge\@title}
```

`\printsubtitle`

```
1571 \newcommand*{\printsubtitle}{\Large\itshape\@subtitle\bigskip}
```

`\printauthor`

```
1572 \newcommand*{\printauthor}
1573 {\large\begin{tabular}{t}{c}\@author\end{tabular}}
```


\printdate

```
1574 \newcommand*{\printdate}{\small\textit{\@date}}
```

42.5 \maketitle for print output

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

```
1575 \providecommand{\maketitle}{}
1576 \if@titlepage
1577   \renewcommand{\maketitle}{\begin{titlepage}%
1578     \let\footnotesize\small
1579     \let\footnoterule\relax
1580     \let \footnote \thanks
1581     \@bsmarkseries
1582     \def\@makefnmark{\rlap{\@textsuperscript{%
1583       \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1584       \long\def\@makefntext##1{\makethanksmark ##1}
1585     \null\vfil
1586     \vskip 60\p@
1587     \vspace*{\droptitle}
1588     \maketitlehooka
1589     \ifcseempty{@published}
1590   {}
1591   {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1592   {\@bsprettitle \@title \@bsposttitle}
1593   \ifcseempty{@subtitle}
1594 {}
1595 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1596   \maketitlehookb
1597   {\@bspreauthor \@author \@bspostauthor}
1598   \maketitlehookc
1599   {\@bspredate \@date \@bspostdate}
1600   \maketitlehookd
1601   \par
1602   \@thanks
1603   \vfil\null
1604   \end{titlepage}%
1605   \@bscontmark % \setcounter{footnote}{0}%
1606   %% \bsmtitleempty
1607 } % end titlepage defs
1608 \else
1609   \renewcommand{\maketitle}{\par
1610     \begingroup
1611     \@bsmarkseries
1612     \def\@makefnmark{\rlap{\@textsuperscript{%
1613       \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1614       \long\def\@makefntext##1{\makethanksmark ##1}
1615     \null\vfil
1616     \vskip 60\p@
1617     \vspace*{\droptitle}
1618     \maketitlehooka
1619     \ifcseempty{@published}
1620   {}
1621   {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1622   {\@bsprettitle \@title \@bsposttitle}
1623   \ifcseempty{@subtitle}
1624 {}
1625 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1626   \maketitlehookb
1627   {\@bspreauthor \@author \@bspostauthor}
1628   \maketitlehookc
1629   {\@bspredate \@date \@bspostdate}
1630   \maketitlehookd
1631   \par
1632   \@thanks
1633   \vfil\null
1634   \end{titlepage}%
1635   \@bscontmark % \setcounter{footnote}{0}%
1636   %% \bsmtitleempty
1637 } % end titlepage defs
1638 \fi
```

```

1614     \long\def\@makefntext##1{\makethanksmark ##1}
1615     \if@twocolumn
1616         \ifnum \col@number=\@ne
1617             \@maketitle
1618         \else
1619             \twocolumn[\@maketitle]%
1620         \fi
1621     \else
1622         \newpage
1623         \global\@topnum\z@
1624         \@maketitle
1625     \fi
1626     \thispagestyle{plain}\@thanks
1627 \endgroup
1628 \bscontmark % \setcounter{footnote}{0}%
1629 %%% \bsmttitleempty
1630 } % end non-titlepage
1631
1632 \def\@maketitle{%
1633     \newpage
1634     \null
1635     \vskip 2em%
1636     \vspace*{\droptitle}
1637     \maketitlehooka
1638     \ifcsempy{@published}
1639 {}
1640 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1641     {\@bspretitle \@title \@bsposttitle}
1642     \ifcsempy{@subtitle}
1643 {}
1644 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1645     \maketitlehookb
1646     {\@bspreadauthor \@author \@bspostauthor}
1647     \maketitlehookc
1648     {\@bspredate \@date \@bspostdate}
1649     \maketitlehookd
1650     \par
1651     \vskip 1.5em}
1652 \fi
1653
1654 \providecommand{\maketitlehookaa}{}
1655
1656 \providecommand{\maketitlehookaaa}{}
1657
1658 \newcommand{\prepublished}[1]{%
1659 \def\@bsprepublished{#1}%
1660 }
1661
1662 \newcommand{\postpublished}[1]{%
1663 \def\@bspostpublished{#1}%

```

```

1664 }
1665
1666 \newcommand{\presubtitle}[1]{%
1667 \def\@bspresubtitle{#1}%
1668 }

```

`\presubtitle` Hook after printing the subtitle.

```

1669 \newcommand{\postsubtitle}[1]{%
1670 \def\@bspostsubtitle{#1}%
1671 }

```

Initial settings:

```

1672 \if@titlepage
1673 \prepublished{
1674 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1675 \begin{center}}
1676 \postpublished{\par\end{center}\vskip 2em}
1677
1678 \presubtitle{\unskip\begin{center}\unskip}
1679 \postsubtitle{\par\end{center}\vskip 2em}
1680 \else
1681 \prepublished{\begin{center}}
1682 \postpublished{\par\end{center}\vskip 0.5em}
1683
1684 \presubtitle{\begin{center}\unskip}
1685 \postsubtitle{\par\end{center}\vskip 0.5em}
1686 \fi

```

1687 \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: 1688 \begin{warppHTML}

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

```
1689 \newcommand*{\LWR@maketitlesetup}{%
```

Select which kind of footnote marks to use:

```
1690 \@bsmarkseries
1691 \@mpbsmarkseries
```

Redefine the footnote mark:

```
1692 \def\@makefnmark{\textsuperscript{\thefootnote}}%

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

Redefine the footnote text:

```
1693 \long\def\@makefntext##1{%
```

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

```
1694 \makethanksmark \LWR@origspace{1in}

\makethanksmark ⇒ \thanksfootmark ⇒ \tmark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text and a closing paragraph tag:

```
1695 ##1\LWR@htmltagc{/LWR@tagregularparagraph}%
1696 }%
1697 }
```

`\@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`.

```
1698 \def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
1699 \mathsection\or \mathparagraph\or \text{\HTMLUnicode{2016}}\or
1700 **\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.

```
1701 \renewcommand*{\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
1702 \begin{titlepage}
```

Set up special patches:

1703 \LWR@maketitlesetup

Typeset the title, etc:

1704 \@maketitle

Immediately generate any `\thanks` footnotes:

1705 \@thanks

Close the HTML titlepage div:

1706 \end{titlepage}

Reset the footnote counter:

1707 \@bscontmark

1708 }

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

```

1709 \DeclareDocumentCommand{\@maketitle}{-}{%
1710 \maketitlehooka
1711 \ifcsemtyp{@published}
1712 {}
1713 {\@bsprepublished \@published \@bspublished}\maketitlehookaa}
1714 {\@bspublished \@title \@bspublished}
1715 \ifcsemtyp{@subtitle}
1716 {}
1717 {\maketitlehookaaa{\@bspublished \@subtitle \@bspublished}}
1718 \maketitlehookb
1719 {\@bspublished \@author \@bspublished}
1720 \maketitlehookc
1721 {\@bspublished \@date \@bspublished}
1722 \maketitlehookd
1723 }

```

```
1724 \providecommand{\maketitlehookaa}{}

```

```
1725 \providecommand{\maketitlehookaaa}{}

```

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

```
1726 \newcommand*{\LWR@titlingmaketitle}{%
```

Typeset the title, etc:

```
1727 \@maketitle
```

Immediately generate any \thanks footnotes:

```
1728 \@thanks
```

```
1729 }
```

```
\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.

```
1730 \let\LWR@origthanksmarkseries\thanksmarkseries
```

```
1731 \renewcommand{\thanksmarkseries}[1]{%
```

```
1732 \def\@mpbsmarkseries{%
```

```
1733 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}
```

```
1734 \LWR@origthanksmarkseries{#1}
```

```
1735 }
```

```
1736 \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: 1737 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the **babel** package.

```
1738 \providecommand*{\abstractname}{Abstract}
```

Env **abstract**

```
1739 \DeclareDocumentEnvironment{abstract}{}%
```

```
1740 {
```

```
1741 \BlockClass{abstract}
```

```

1742 \BlockClassSingle{abstracttitle}{\abstractname}
1743 }
1744 {
1745 \endBlockClass
1746 }

1747 \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:

```

1748 \begin{warpHTML}
1749 \newcommand{\attribution}[1]{%
1750 \InlineClass{attribution}{--\,#1}}% emdash
1751 \end{warpHTML}

```

for PRINT output:

```

1752 \begin{warpprint}
1753 \newcommand{\attribution}[1]{\textsc{--\,#1}}
1754 \end{warpprint}

```

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

for HTML output:

```

1755 \begin{warpHTML}
1756 \newcommand{\citetitle}[1]{%
1757 \InlineClass{citetitle}{--\,#1}}% emdash
1758 \end{warpHTML}

```

for PRINT output:

```

1759 \begin{warpprint}
1760 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
1761 \end{warpprint}

```

44.2 Quotes, quotations

for HTML output:

```

1762 \begin{warpHTML}

```

Env quote

```

1763 \renewenvironment*{quote}
1764 {\LWR@htmlblocktag{blockquote}}
1765 {\LWR@htmlblocktag{/blockquote}}
1766
1767 \renewenvironment*{quotation}
1768 {\LWR@htmlblocktag{blockquotation}}
1769 {\LWR@htmlblocktag{/blockquotation}}

1770 \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> is if there is a wide <code>\flagverse</code> in use, such as the word “Chorus”, in which case the value of <code>\HTMLleftmargini</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.

for HTML & PRINT: 1771 `\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.

```

1772 \newlength{\HTMLvleftskip}

```



```
1773 \setlength{\HTMLvleftskip}{1em}
```

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

```
1774 \newlength{\HTMLleftmargini}
1775 \setlength{\HTMLleftmargini}{4.5em}

1776 \end{warpall}
```

45 Verbatim

for HTML output: 1777 \begin{warpHTML}

Env `verbatim`

```
1778 \AfterEndPreamble{
1779 \AtBeginEnvironment{verbatim}{\LWR@atbeginverbatim{verbatim}\unskip\vspace*{-\baselineskip}}
1780 \AfterEndEnvironment{verbatim}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
1781 }

1782 \end{warpHTML}
```

46 Fancyvrb

for HTML & PRINT: 1783 \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.

```
1784 \newlength{\VerbatimHTMLWidth}
1785 \setlength{\VerbatimHTMLWidth}{4in}
1786 \end{warpall}
```

for HTML output: 1787 \begin{warpHTML}

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

```
1788 \newbool{LWR@verbtags}
1789 \booltrue{LWR@verbtags}
```

For `\VerbatimFootnotes`:

```
1790 \renewcommand{\VerbatimFootnotes}{
1791 \PackageError{lwarp}
1792 {Verbatim footnotes are not yet supported by lwarp.}
1793 {This may be improved when lwarp uses LaTeX footnotes instead of pagenotes.}
1794 }
```

`\LWR@atbeginverbatim` `{\class}`

Encloses a verbatim environment with the given CSS class.

```
1795 \newcommand*{\LWR@atbeginverbatim}[1]
1796 {%
```

Avoid excessive space between lines:

```
1797 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1798 \LWR@stoppars%
```

Create a new `pre` of the given class:

```
1799 \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.

```
1800 \LWR@origttfamily%
```

Do not produce HTML tags for `\hspace` inside a verse `par`. Restore plain L^AT_EX `\hspace` functionality:

```
1801 \let\hspace\LWR@orighspace%
1802 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
1803 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
1804 \unskip%
```

At the end of the environment, close the `pre`:

```

1805 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
1806
1807 }{}%

```

Resume regular paragraph handling:

```

1808 \LWR@startpars%
1809 }

```

`\LWR@Verbatimclass` Holds the class of the following verbatim.

```

1810 \newcommand*{\LWR@Verbatimclass}{fancyvrb}

```

Env `VerbatimClass` `{(class)}` [*Verbatim options*]

Creates a `Verbatim` enclosed in a DIV of the given class.

```

1811 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1812 {%
1813 \renewcommand*{\LWR@Verbatimclass}{#1}%
1814 \LWR@origVerbatim[#2]%
1815 }
1816 {\endVerbatim}

```

After the preamble is loaded, after any patches to `Verbatim`:

```

1817 \AfterEndPreamble{

```

Remember the original definition of `Verbatim`:

```

1818 \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.

```

1819 \renewcommand*{\Verbatim}{%
1820 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
1821 \LWR@origVerbatim%
1822 }

```

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.

```

1823 \newcommand*{\LWR@fvstartnone}{%
1824 \LWR@traceinfo{fvstartnone}%
1825 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
1826 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
1827 }
1828
1829 \newcommand*{\LWR@fvendnone}{%
1830 \LWR@traceinfo{fvendnone}%
1831 \hbox to\z@{\LWR@afterendverbatim}%
1832 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
1833 }
1834
1835 \newcommand*{\LWR@fvstartsingle}{%
1836 \LWR@traceinfo{fvstartsingle}%
1837 \LWR@fvstartnone%
1838 \FV@BeginListFrame@Single%
1839 }
1840
1841 \newcommand*{\LWR@fvendsingle}{%
1842 \LWR@traceinfo{fvendsingle}%
1843 \FV@endListFrame@Single%
1844 \LWR@fvendnone%
1845 }
1846
1847 \newcommand*{\LWR@fvstartline}{%
1848 \LWR@traceinfo{fvstartline}%
1849 \LWR@fvstartnone%
1850 \FV@BeginListFrame@Lines%
1851 }
1852
1853 \newcommand*{\LWR@fvendline}{%
1854 \LWR@traceinfo{fvendline}%
1855 \FV@endListFrame@Lines%
1856 \LWR@fvendnone%
1857 }

```

The following patches select the start/left/right/end behaviors depending on **frame**.
Original code is from the fancyvrb package.

```

1858 \def\FV@Frame@none{%
1859 \let\FV@BeginListFrame\LWR@fvstartnone%
1860 \let\FV@LeftListFrame\relax%
1861 \let\FV@RightListFrame\relax%
1862 \let\FV@endListFrame\LWR@fvendnone%
1863
1864 \def\FV@Frame@single{%
1865 \let\FV@BeginListFrame\LWR@fvstartsingle%
1866 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1867 \let\FV@RightListFrame\FV@RightListFrame@Single%

```

```

1868 \let\FV@EndListFrame\LWR@fvendsingle}
1869
1870 \def\FV@Frame@lines{%
1871 \let\FV@BeginListFrame\LWR@fvstartline%
1872 \let\FV@LeftListFrame\relax%
1873 \let\FV@RightListFrame\relax%
1874 \let\FV@EndListFrame\LWR@fvendline}
1875
1876 \def\FV@Frame@topline{%
1877 \let\FV@BeginListFrame\LWR@fvstartline%
1878 \let\FV@LeftListFrame\relax%
1879 \let\FV@RightListFrame\relax%
1880 \let\FV@EndListFrame\LWR@fvendnone}
1881
1882 \def\FV@Frame@bottomline{%
1883 \let\FV@BeginListFrame\LWR@fvstartnone%
1884 \let\FV@LeftListFrame\relax%
1885 \let\FV@RightListFrame\relax%
1886 \let\FV@EndListFrame\LWR@fvendline}
1887
1888 \def\FV@Frame@leftline{%
1889 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
1890 \ifx\FancyVerbFillColor\relax%
1891 \let\FV@FrameFillLine\relax%
1892 \else%
1893 \@tempdima\FV@FrameRule\relax%
1894 \multiply\@tempdima-\tw@%
1895 \edef\FV@FrameFillLine{%
1896 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
1897 \kern-\number\@tempdima sp}}%
1898 \fi%
1899 \let\FV@BeginListFrame\LWR@fvstartnone%
1900 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1901 \let\FV@RightListFrame\relax%
1902 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the `fancyvrb` package.

```

1903 \def\FV@SingleFrameLine#1{%
1904 \hbox to\z@{%
1905 % \kern\leftmargin
1906 \ifnum#1=\z@\relax
1907 \let\FV@Label\FV@LabelBegin
1908 \else
1909 \let\FV@Label\FV@LabelEnd
1910 \fi
1911 \ifx\FV@Label\relax
1912 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%

```

```

1913     \else
1914         \ifnum#1=\z@
1915 %         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
1916         \ifx\FV@LabelPositionTopLine\relax
1917 \else
1918     \LWR@htmltagc{div class="fancyvrblabel"}
1919 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
1920 \LWR@htmltagc{/div}
1921 \fi
1922     \else
1923 %         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
1924         \ifx\FV@LabelPositionBottomLine\relax
1925 \else
1926     \LWR@htmltagc{div class="fancyvrblabel"}
1927 \LWR@origtextrm{\FV@LabelEnd}
1928 \LWR@htmltagc{/div}
1929 \fi
1930     \fi
1931
1932     \fi
1933     \hss
1934 }
1935 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

1936 \def\FV@ListProcessLine#1{%
1937     \hbox to \hsize{%
1938 %         \kern\leftmargin
1939         \hbox to \VerbatimHTMLWidth {%
1940             \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
1941 \FV@LeftListNumber%
1942 %         \FV@LeftListFrame
1943         \FancyVerbFormatLine{#1}%
1944         \hss%
1945 %         \FV@RightListFrame
1946         \FV@RightListNumber%
1947 }%
1948     \hss% required to avoid underfull hboxes
1949 }
1950 }

```

Env BVerbatim

```

1951 \AtBeginEnvironment{BVerbatim}
1952 {
1953 \LWR@atbeginverbatim{bverbatim}
1954

```

```

1955 }
1956
1957 \AfterEndEnvironment{BVerbatim}
1958 {
1959 \leavevmode\par\vspace{-\baselineskip}
1960 \LWR@afterendverbatim
1961 }

```

Env **LVerbatim** No changes required.

End of the modifications to make at the end of the preamble:

```

1962 } % \AfterEndPreamble

```

\UseVerbatim $\{\langle text \rangle\}$

No changes required.

```

1963 \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: 1964 **\begin{warpHTML}**

\@begintheorem $\{\langle name \rangle\}$ $\{\langle number \rangle\}$

```

1965 \renewcommand{\@begintheorem}[2]{%
1966 \BlockClass{theoremcontents}
1967 \InlineClass{theoremlabel}{#1\ #2\ }
1968 }

```

\@opargbegintheorem $\{\langle name \rangle\}$ $\{\langle number \rangle\}$ $\{\langle oparg \rangle\}$

```

1969 \renewcommand{\@opargbegintheorem}[3]{%
1970 \BlockClass{theoremcontents}
1971 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
1972 }

```

`\@endtheorem`

```

1973 \renewcommand*\@endtheorem}{%
1974 \endBlockClass% theoremcontents
1975 }

```

```

1976 \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: 1977 \begin{warpHTML}

```

1978 \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.

```

1979 \newcommand*\LWR@itemizeitem}{%
1980 \LWR@stoppars%
1981 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
1982 \LWR@htmltag{li}%
1983 \LWR@startpars%
1984 \LWR@origitem%
1985 }

```


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*]

```
1986 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
1987
1988 \newcommand*{\LWR@itemizestart}{%
1989 \LWR@stoppars%
1990 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
1991 \LWR@htmltag{ul style="list-style-type:none">{}}%
1992 \LWR@startpars%
1993 \let\item\LWR@itemizeitem%
1994 }
1995
1996 \AtEndEnvironment{itemize}{\LWR@itemizeend}
1997
1998 \newcommand*{\LWR@itemizeend}{%
1999 \LWR@stoppars%
2000 \LWR@closeprevious{\LWR@depthlistitem}%
2001 \LWR@closeoneprevious{}}%
2002 \LWR@startpars%
2003 }
```

48.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env **enumerate** [*enumitem options*]

```
2004 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
2005
2006 \newcommand*{\LWR@enumeratestart}{%
2007 \LWR@stoppars%
2008 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
2009 \LWR@htmltag{ul style="list-style-type:none">{}}%
2010 \LWR@startpars%
2011 \let\item\LWR@itemizeitem%
2012 }
```

```

2013
2014
2015 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
2016
2017 \newcommand*{\LWR@enumerateend}{\%
2018 \LWR@stoppars%
2019 \LWR@closeprevious{\LWR@depthlistitem}%
2020 \LWR@closeoneprevious{}}%
2021 \LWR@startpars%
2022 }

```

48.3 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```

2023 \newcommand*{\LWR@descitem}[1][]{\%
2024 {%
2025 \LWR@stoppars%
2026 \LWR@setlatestname{#1}%
2027 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%
2028 \LWR@origitem[]%

```

Be sure the label doesn't print to the left of the rest of the file:

```

2029 \LWR@origspace{1in}
2030 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
2031 \LWR@orignewline%
2032 \LWR@htmltag{dd}%
2033 \LWR@startpars%
2034 }

```

Env `description` [*enumitem options*]

```

2035 \AtBeginEnvironment{description}{\LWR@descriptionstart}
2036
2037 \newcommand*{\LWR@descriptionstart}{\%
2038 \LWR@stoppars%
2039 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
2040 \LWR@htmltag{dl}%
2041 \LWR@startpars%
2042 \let\item\LWR@descitem%
2043 }
2044
2045 \AtEndEnvironment{description}{\LWR@descriptionend}
2046
2047 \newcommand*{\LWR@descriptionend}{\%

```

```

2048 \LWR@stoppars%
2049 \LWR@closeprevious{\LWR@depthlistitem}%
2050 \LWR@closeoneprevious{}%
2051 \LWR@startpars%
2052 }

```

```
\newlist {<name>} {<type>} {<maxdepth>}
```

```
\renewlist {<name>} {<type>} {<maxdepth>}
```

For `enumitem` lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```

2053 \let\LWR@orignewlist\newlist
2054
2055 \renewcommand*\newlist{[3]}{
2056 \LWR@orignewlist{#1}{#2}{#3}%
2057 \AtBeginEnvironment{#1}{\csuse{LWR@#2start}}%
2058 \AtEndEnvironment{#1}{\csuse{LWR@#2end}}%
2059 }

2060 \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.
- `\newcolumntype` is ignored; unknown column types are set to `l`.
- `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.)

 `\multirow &`
`\multicolumn`

-  `\multirow`


`\multirow with rules`


`rule at last row`

 paragraphs

`\cmidrule width, trim`

`longtable headings`

 `\warpprintonly`

 S columns

- 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 \\}
```

49.1 Token lookahead

Used by `\LWR@futurenonospacelet` to look at the next token.

for HTML output: 2061 `\begin{warpHTML}`

`\LWR@mynexttoken`

2062 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonSPACElet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonSPACElet`

```
2063 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
2064 \afterassignment\LWR@fnsLone\let\nexttoken= }
2065 \def\LWR@fnsLone{\expandafter\futurelet\LWR@cs\LWR@fnsLtwo}
2066 \def\LWR@fnsLtwo{%
2067 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfnsLthree%
2068 \else\let\next=\nexttoken\fi\next}
2069 \def\@BTfnsLthree{\afterassignment\LWR@fnsLone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
2070 \newcommand*{\LWR@getmynexttoken}{%
2071 % nothing must follow this next line
2072 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumnTag
2073 }
```

49.2 Booleans

Bool `LWR@startedrow` True if should print a row tag before this column.

```
2074 \newbool{LWR@startedrow}
2075 \boolfalse{LWR@startedrow}
```

Bool `LWR@doinghline` True if the next row will have an hline above it.

```
2076 \newbool{LWR@doinghline}
2077 \boolfalse{LWR@doinghline}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
2078 \newbool{LWR@doingtbrule}
2079 \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.

```
2080 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
2081 \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.

```
2082 \newbool{LWR@intabularmetadata}
```

```
2083 \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.

```
2084 \newcommand*{\LWR@closetabledatacell}{%
2085 \global\booltrue{LWR@intabularmetadata}%
2086 \ifbool{LWR@exitingtabular}{}%
2087 {% not exiting tabular
2088 \ifbool{LWR@skippingmrowcell}{}%
2089 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents:

```
2090 \unskip%
2091 \LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}%
2092 \LWR@getexparray{LWR@colatspec}{\theLWR@tablecolspos}%
2093 \LWR@getexparray{LWR@colbangspec}{\theLWR@tablecolspos}%
```

Close paragraphs:

```
2094 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
2095 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
2096 \unskip\LWR@htmltag{/td}\LWR@orignewline%
2097 }% not skipping mrowcell
2098 }% not exiting tabular
2099 \global\boolfalse{LWR@skippingmrowcell}%
2100 }
```

`LWR@tabulardepth` tracks whether `&` is being used inside a `tabular`.

```
2101 \newcounter{LWR@tabulardepth}
2102 \setcounter{LWR@tabulardepth}{0}
2103
```

When not used inside a `tabular`, `&` performs its original function as recorded here (with catcode 4).

```
2104 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2105 \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.

```
2106 \catcode'\&=\active
2107
2108 \gdef&{%
2109 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2110 {%
```

If not skipping a multirow cell, close the current data cell.

```
2111 \unskip%
2112 \LWR@closetabledatacell%
```

Move to the next column.

```
2113 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
2114 \LWR@getmynexttoken%
2115 }%
```

If not inside a `tabular`, performs the original action:

```
2116 {\LWR@origampmacro}%
2117 }
2118 \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`

```
2119 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
2120 {%
2121 \LWR@closetabledatacell%
```

Finish the previous row:

```
2122 \LWR@htmltag{/tr}\LWR@orignewline
2123 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
2124 \global\boolfalse{\LWR@startedrow}
```

Additional setup:

```
2125 \global\boolfalse{\LWR@doinghline}%
2126 \global\boolfalse{\LWR@doingtbrule}%
2127 \LWR@clearmidrules%
```

Start at first column:

```
2128 \setcounter{\LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
2129 \LWR@getmynexttoken%
2130 }
```

49.5 Variables


```

2131 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2132 \newcommand*{\LWR@pposition}{}
2133 \newcommand*{\LWR@pleft}{}
2134 \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.

```
2135 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
2136 \newcommand*{\LWR@strresult}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
2137 \newcommand*{\LWR@origcolspec}{}

```

`Ctr` `LWR@tablecolwidth` Holds the width of the table specification.

(This is not the total # columns.)

```
2138 \newcounter{LWR@tablecolwidth}

```

`Ctr` `LWR@tablecolspos` Where are currently looking into the table column specification.

```
2139 \newcounter{LWR@tablecolspos}

```

`Ctr` `LWR@tabletotalcols` Holds the final number of table columns.

```
2140 \newcounter{LWR@tabletotalcols}

```

`Ctr` `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
2141 \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.

```
2142 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2143 \LWR@traceinfo{at column}%
2144 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
2145 \LWR@traceinfo{about to read the next token:}%
2146 \expandarg%
2147 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2148 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
2149 \LWR@traceinfo{have now read the next token}%
2150 \ifthenelse{\cinttest{\value{LWR@tabletotalcols}}=0}{
2151 {% left edge of the table:
2152 \LWR@traceinfo{at the left edge}%
2153 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
2154 }%
2155 {% not at the left edge:
2156 \LWR@traceinfo{not at the left edge}%
2157 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2158 \LWR@traceinfo{at \theLWR@tabletotalcols: %
2159 \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}!}%
2160 \let\LWR@strresult\relax%
2161 \booltrue{LWR@validtablecol}%
2162 }%
2163 }
```

`\LWR@parsebangcolumn`

```
2164 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
2165 \LWR@traceinfo{bang column}%
2166 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```

2167 \LWR@traceinfo{about to read the next token:}%
2168 \expandarg%
2169 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2170 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2171 \LWR@traceinfo{have now read the next token}%
2172 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
2173 {% left edge of the table:
2174 \LWR@traceinfo{at the left edge}%
2175 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@strresult}%
2176 }%
2177 {% not at the left edge:
2178 \LWR@traceinfo{not at the left edge}%
2179 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2180 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colbangspec(\theLWR@tabletotalcols)!}%
2181 }%
2182 \let\LWR@strresult\relax%
2183 \booltrue{LWR@validtablecol}%
2184 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```

2185 \newcommand*{\LWR@parsebeforecolumn}{%

```

Move to the next token after the '>':

```

2186 \addtocounter{LWR@tablecolspos}{1}%

```

Read the next token, expanding once into \LWR@strresult:

```

2187 \expandarg%
2188 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2189 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

2190 \LWR@setexparray{LWR@colbefore-spec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2191 \let\LWR@strresult\relax%
2192 \booltrue{LWR@validtablecol}%
2193 }

```

\LWR@parseaftercolumn Handles <{text} columns.

```

2194 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '<':

```
2195 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
2196 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
2197 \expandarg%
2198 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2199 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
2200 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2201 \let\LWR@strresult\relax%
2202 \booltrue{LWR@validtablecol}%
2203 }
```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```
2204 \newcommand*{\LWR@parseskipcolumn}{%
2205 \booltrue{LWR@validtablecol}%
2206 }
```

49.7 Parsing 'l', 'c', or 'r' columns

`\LWR@parsenormalcolumn` $\{\langle thiscolumn \rangle\}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
2207 \newcommand*{\LWR@parsenormalcolumn}[1]{%
2208 \appto\LWR@tablecolspec{#1}%
2209 \addtocounter{LWR@tabletotalcols}{1}%
2210 \addtocounter{LWR@tabletotalcolsnext}{1}%
2211 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%
2212 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{\relax}%
2213 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\relax}%
2214 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
2215 \booltrue{LWR@validtablecol}%
2216 }
```

49.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```
2217 \newcommand*\LWR@parsepcolumn}[1]{%
```

Converts to the given column type.

```
2218 \LWR@parsenormalcolumn{#1}%
```

skips the following width

```
2219 \addtocounter{LWR@tablecolspos}{1}%
2220 }
```

49.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```
2221 \newcommand*\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
2222 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
2223 \addtocounter{LWR@tablecolspos}{3}%
2224 }
```

49.10 Parsing the column specifications



HTML CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 6 shows the L^AT_EX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

```
2225 \newbool{LWR@validtablecol}
```

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 7: Tabular HTML column conversions

-
- l, r, c:** Converted to table cells without paragraph tags.
Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.
- p:** Converted to table cells with paragraph tags. Ref: Table 6, \LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
- m:** With paragraph tags, CSS `vertical-align:middle`.
- b:** With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.
- P, M, B:** Horizontally-centered versions.
- S:** Converted to 'r'. From the `siunitx` package.
- D:** Converted to 'c'. From the `dcolumn` package.
- @, !, >, <:** One each, in that order.
- Unknown:** Converted to 'l'.
- \newcolumn:** Currently treated as unknown.
-

`\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`.

```
2226 \newcommand*{\LWR@parsetablecols}[1]{%
2227 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
2228 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
2229 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
2230 \setcounter{LWR@tabletotalcols}{0}%
2231 \setcounter{LWR@tabletotalcolsnext}{1}%
2232 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%
2233 \LWR@setexparray{LWR@colatspec}{1}{\relax}%
2234 \LWR@setexparray{LWR@colatspec}{2}{\relax}%
2235 \LWR@setexparray{LWR@colatspec}{3}{\relax}%
2236 \LWR@setexparray{LWR@colbangspec}{leftedge}{\relax}%
2237 \LWR@setexparray{LWR@colbangspec}{1}{\relax}%
2238 \LWR@setexparray{LWR@colbangspec}{2}{\relax}%
2239 \LWR@setexparray{LWR@colbangspec}{3}{\relax}%
2240 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
2241 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
2242 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
2243 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
2244 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
2245 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%
```

Starting at the first column specification:

```
2246 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
2247 \LWR@traceinfo{about to StrLen}%
2248 \noexpandarg%
2249 \StrLen{#1}[\LWR@strresult]%
2250 \fullexpandarg%
```

```
2251 \LWR@traceinfo{finished StrLen}%
2252 \setcounter{LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:


```
2253 \whileof{\not\value{LWR@tablecolspos}>\value{LWR@tablecolswidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
2254 \noexpandarg%
2255 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
2256 \fullexpandarg%
```

Not yet found a valid column type

```
2257 \boolfalse{LWR@validtablecol}%
```

 Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
2258 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
2259 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
2260 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
2261 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
2262 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
2263 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
2264 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
2265 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
2266 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
2267 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
2268 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
2269 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%
```

From the dcolumn package:

```
2270 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
2271 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%
```

Many people define centered versions “P”, “M”, and “B”:
`\newcolumn{P}[1]{>{\centering\arraybackslash}p{#1}}`

```
2272 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{%
```



```

2273 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
2274 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to a p column:

```

2275 \ifbool{\LWR@validtablecol}{}{}%
2276 \LWR@parsenormalcolumn{1}%
2277 }%
2278 \addtocounter{\LWR@tablecolspos}{1}%
2279 }%
2280 }%

```

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.

```

2281 \newcommand*{\LWR@maybenewtablerow}
2282 {%
2283 \ifbool{\LWR@startedrow}%
2284 {}% started the row
2285 {}% not started the row

```

Remember that now have started the row:

```

2286 \global\booltrue{\LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

2287 \global\booltrue{\LWR@intabularmetadata}%
2288 \ifbool{\LWR@doinghline}%
2289 {\LWR@htmltag{tr class="hline"{} }\LWR@orignewline}%
2290 {}% not doing hline
2291 \ifbool{\LWR@doingtbrule}%
2292 {\LWR@htmltag{tr class="tbrule"{} }\LWR@orignewline}%
2293 {\LWR@htmltag{tr} \LWR@orignewline}%
2294 }% end of not doing hline
2295 }% end of not started the row
2296 }

```

49.12 Data opening tag

`\LWR@tabledatasinglecolumnntag` Print a table data opening tag with style for alignment

```

2297 \newcommand*{\LWR@tabledatasinglecolumnntag}%
2298 {%
2299 \LWR@maybenewtablerow%

```

If have found the end of tabular command, do not create the next data cell:

```

2300 \ifbool{\LWR@exitingtabular}{}%
2301 {% not exiting tabular

```

Fetch the current column's alignment character into \LWR@strresult:

```

2302 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

2303 \LWR@htmltag{td class="td%

```

append this column's spec:

```

2304 \LWR@strresult%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```

2305 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
2306 "{}}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

2307 \ifthenelse{%
2308 \equal{\LWR@strresult}{p}\OR%
2309 \equal{\LWR@strresult}{m}\OR%
2310 \equal{\LWR@strresult}{b}\OR%
2311 \equal{\LWR@strresult}{P}\OR%
2312 \equal{\LWR@strresult}{M}\OR%
2313 \equal{\LWR@strresult}{B}\OR%
2314 \equal{\LWR@strresult}{X}%
2315 }%
2316 {% allow pars
2317 \LWR@startpars%
2318 \global\booltrue{\LWR@tableparcell}%
2319 }% allow pars
2320 }% no pars

```

Print the @ and ! contents before first column, and then the > contents:

```

2321 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
2322 {%
2323 \LWR@getexparray{\LWR@colatspec}{leftedge}%
2324 \LWR@getexparray{\LWR@colbangspec}{leftedge}%

```

```

2325 }% left edge
2326 {}% not left edge
2327 \LWR@getexparray{LWR@colbeforespec}{\theLWR@tablecolspos}%
2328 \global\boolfalse{LWR@intabularmetadata}%
2329 }% not exiting tabular
2330 }%

```

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.

Ctrl LWR@midrulecounter Indexes across the LWR@midrules data array.

```

2331 \newcounter{LWR@midrulecounter}

```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \.

Clears all LWR@midrules markers for this line.

```

2332 \newcommand*{\LWR@clearmidrules}
2333 {%
2334 \setcounter{LWR@midrulecounter}{1}%
2335 \whiledo{%
2336 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
2337 }%
2338 {%
2339 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{\relax}%
2340 \addtocounter{LWR@midrulecounter}{1}%
2341 }%
2342 }

```

\LWR@subcmidrule [*width*] [*trim*] [*leftcolumn*] [*rightcolumn*]

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

2343 \newcommand*{\LWR@subcmidrule}[4]{%
2344 \setcounter{LWR@midrulecounter}{#3}%
2345 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
2346 {%
2347 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2348 \addtocounter{LWR@midrulecounter}{1}%
2349 }% end of the whiledo
2350 }

```

\LWR@docmidrule [*width*] [*trim*] [*leftcolumn-rightcolumn*]

Marks LWR@midrules array elements to be “Y” from left to right columns.

```
2351 \NewDocumentCommand{\LWR@docmidrule}{o d() >\SplitArgument{1}{-}m}%
2352 {\LWR@subcmidrule{#1}{#2}#3}
```

49.14 Multicolumns

49.14.1 Parsing multicolumns

```
2353 \newcounter{LWR@tablemulticolwidth}
2354 \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.

```
2355 \newcommand*{\LWR@printmccoltype}[1]{%
2356 \LWR@traceinfo{lw@printmccoltype -#1-}%
```

Get one token of the column spec:

```
2357 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
2358 \IfStrEq{\LWR@strresult}{l}{l}{}%
2359 \IfStrEq{\LWR@strresult}{c}{c}{}%
2360 \IfStrEq{\LWR@strresult}{r}{r}{}%
2361 \IfStrEq{\LWR@strresult}{p}{p}{}%
2362 \IfStrEq{\LWR@strresult}{m}{m}{}%
2363 \IfStrEq{\LWR@strresult}{b}{b}{}%
2364 \IfStrEq{\LWR@strresult}{P}{P}{}%
2365 \IfStrEq{\LWR@strresult}{M}{M}{}%
2366 \IfStrEq{\LWR@strresult}{B}{B}{}%
2367 \IfStrEq{\LWR@strresult}{S}{r}{}%
2368 \IfStrEq{\LWR@strresult}{X}{p}{}%
2369 \LWR@traceinfo{lw@printmccoltype done}%
2370 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
2371 \newcommand*{\LWR@multicolpartext}{%
2372 \LWR@startpars%
2373 \LWR@multicoltext%
2374 \LWR@stoppars%
2375 }
```

`\LWR@multicolother` $\{\langle colspec \rangle\}$ For @, >, <, print the next token without paragraph tags:

```
2376 \newcommand*{\LWR@multicolother}[1]{%
2377 \addtocounter{\LWR@tablemulticolspos}{1}%
2378 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
2379 \LWR@strresult%
```

A valid column data type was found:

```
2380 \booltrue{\LWR@validtablecol}%
2381 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
2382 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
2383 \booltrue{\LWR@validtablecol}%
2384 }
```

`\LWR@printmccoldata` $\{\langle colspec \rangle\}$ Print the data for any valid column type found.

```
2385 \newcommand*{\LWR@printmccoldata}[1]{%
2386 \LWR@traceinfo{\LWR@printmccoldata -#1}%
```

Not yet found a valid column type:

```
2387 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
2388 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
2389 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
2390 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
2391 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
2392 \IfStrEq{\LWR@strresult}{D}{}%
2393 \addtocounter{\LWR@tablemulticolspos}{3}% skip parameters
2394 \LWR@multicoltext%
2395 }{}%
2396 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
2397 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
2398 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
```

```

2399 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
2400 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
2401 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
2402 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
2403 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}{}%
2404 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
2405 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
2406 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
2407 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
2408 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%

```

If an invalid column type:

```

2409 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%

```

Tracing:

```

2410 \LWR@traceinfo{lw@printmccoldata done}%
2411 }

```

`\parsemulticolumnalignment` $\{ \langle 1: \textit{colspec} \rangle \} \{ \langle 2: \textit{printresults} \rangle \}$

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or @, >, <, is a token list which will NOT match l, c, r, or p.

```

2412 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2413 \setcounter{LWR@tablemulticolspos}{1}%
2414 \StrLen{#1}[\LWR@strresult]%
2415 \setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

2416 \whiledo{%
2417 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolswidth}%
2418 }%
2419 {%

```

Execute the assigned print function for each token in the column spec:

```

2420 #2{#1}%

```

Move to the next token in the column spec:

```

2421 \addtocounter{LWR@tablemulticolspos}{1}%
2422 }%
2423 }

```

49.14.2 High-level multicolumn interface

`\LWR@domulticolumn` $\{\langle 1: numcols \rangle\} \{\langle 2: colspec \rangle\} \{\langle 3: text \rangle\}$

```
2424 \newcommand{\LWR@multicoltext}{}
2425
2426 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2427 \LWR@traceinfo{lw@domulticolumn -#1- -#2-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
2428 \renewcommand{\LWR@multicoltext}{%
2429 #3%
2430 \booltrue{\LWR@validtablecol}%
2431 }%
```

Row processing:

```
2432 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
2433 \LWR@htmltag{td colspan="#1"
2434 class="td%
```

Print the column type:

```
2435 \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”.

```
2436 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
2437 "%
2438 }% end of the opening table data tag
2439 \global\boolfalse{\LWR@intabularmetadata}%
2440 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2441 }
```

49.14.3 Longtable captions

`Bool` `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
2442 \newbool{LWR@starredlongtable}
2443 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
2444 \providecommand*{\LTcapttype}{table}
```

```
\LWR@longtabledatacaptiontag * [(\toc entry)] {\caption}
```

```
2445 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2446 {%
```

Remember the latest name for \nameref:

```
2447 \IfValueTF{#2}{% optional given?
2448 \ifthenelse{\equal{#2}{}}{% optional empty?
2449 {\LWR@setlatestname{#3}}% empty
2450 {\LWR@setlatestname{#2}}% given and non-empty
2451 }% optional given
2452 {\LWR@setlatestname{#3}}% no optional
```

create a multicolumn across all the columns

```
2453 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
2454 % \IfBooleanTF{#1}% star?
2455 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
2456 % {\IfValueTF{#2}{\LWR@origcaption{#2}{#3}}{\LWR@origcaption{#3}}}
2457 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
2458 {% yes star
2459 \LWR@htmlblocktag{figcaption}%
2460 #3%
2461 \LWR@htmlblocktag{/figcaption}%
2462 }%
2463 {% No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
2464 \ifbool{LWR@starredlongtable}%
2465 {%
2466 \ifthenelse{\equal{#2}{}}{% TOC entry
2467 }%
2468 {%
2469 \refstepcounter{\LTcapttype}%
```



```

2470 \protected@edef\@currentlabel{%
2471 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2472 }%
2473 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

2474 \LWR@htmlblocktag{figcaption}%
2475 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
2476 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

2477 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

2478 {}%

```

If the optional caption was given, but might only be []:

```

2479 {% TOC entry not empty
2480 \IfNoValueTF{#2}% No TOC entry?

```

The optional caption is []:

```

2481 {% No TOC entry
2482 \addcontentsline%
2483 {\csuse{ext@\LTcapttype}}%
2484 {\LTcapttype}%
2485 {%
2486 \protect\numberline%
2487 {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2488 {\ignorespaces #3\protect\relax}%
2489 }%
2490 }% end of No TOC entry

```

The optional caption has text enclosed:

```

2491 {% yes TOC entry
2492 \addcontentsline%
2493 {\csuse{ext@\LTcapttype}}%
2494 {\LTcapttype}%
2495 {%
2496 \protect\numberline%
2497 {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2498 {\ignorespaces #2\protect\relax}%
2499 }%
2500 }% end of yes TOC entry

```

```

2501 }% end of TOC entry not empty
2502 }% end of no star
2503 }% end of \LWR@domulticolumn
2504
2505 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
2506 \addtocounter{LWR@tablecolspos}{-1}
2507
2508 }

```

49.14.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` $\{\langle numcols \rangle\}$ $\{\langle alignment \rangle\}$ $\{\langle text \rangle\}$

```

2509 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
2510 {%
2511 \LWR@domulticolumn{#1}{#2}{#3}%
2512 \addtocounter{LWR@tablecolspos}{#1}%
2513 \addtocounter{LWR@tablecolspos}{-1}%
2514 }

```

49.15 Multirow

Pkg `multirow`

`\LWR@tabledatamultirowtag` $\{\langle numrows \rangle\}$ $[\langle bigstruts \rangle]$ $\{\langle width \rangle\}$ $[\langle fixup \rangle]$ $\{\langle text \rangle\}$

```

2515 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
2516 {%
2517 \LWR@maybenewtablerow%

```

Print the start of a new table data cell:

```

2518 \LWR@htmltag{td rowspan="#1" class="td%

```

Append this column's spec:

```

2519 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%

```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```

2520 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
2521 "{}}

```

While printing the text, redefine `\` to generate a new line

```

2522 \begingroup \let\\\LWR@endofline #5 \endgroup
2523 \LWR@stoppars%
2524 \global\boolfalse{\LWR@intabularmetadata}%
2525 }%

```

49.16 Utility macros inside a table

```

2526 \newcommand*{\LWR@donothing}{}
2527 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}
2528 \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:

```

2529 \newbool{\LWR@exitingtabular}
2530 \newcommand*{\LWR@tabledatacolumnntag}%
2531 {%

```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```

2532 \let\mynext\LWR@tabledatasinglecolumnntag%

```

If exiting the tabular:

```

2533 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2534 {\booltrue{\LWR@exitingtabular}}}%

```

`longtable` can have a caption in a cell

```

2535 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2536 {\let\mynext\LWR@donothing}}%

```

Look for other things which would not start a table cell:

```

2537 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2538 {\let\mynext\LWR@donothing}}%
2539 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2540 {\let\mynext\LWR@donothing}}%

```

if come to an `\mrowcell`, this is a cell to be skipped over

```

2541 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2542 {\let\mynext\LWR@donothing}{}}%
2543 %
2544 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2545 {\let\mynext\LWR@donothing}{}}%
2546 %
2547 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2548 {%
2549 \let\mynext\LWR@donothing}{}}%
2550 %
2551 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2552 {\let\mynext\LWR@donothing}{}}%
2553 %
2554 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2555 {\let\mynext\LWR@donothing}{}}%
2556 %
2557 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2558 {\let\mynext\LWR@donothing}{}}%
2559 %
2560 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2561 {\let\mynext\LWR@donothing}{}}%
2562 %
2563 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}{%
2564 {\let\mynext\LWR@donothing}{}}%
2565 %
2566 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}{%
2567 {\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:


```

2568 \mynext%
2569 }

2570 \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: 2571 `\begin{warpall}`
 2572 `\newcommand*{\mrowcell}{}%`
 2573 `\end{warpall}`

49.19 New `\tabular` definition

for HTML output: 2574 `\begin{warpHTML}`

Env `LWR@tabular` [$\langle\textit{vertposition}\rangle$] [$\langle\textit{colspecs}\rangle$]

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```
2575 \newenvironment*{LWR@tabular}[2] []
2576 {%
2577 \LWR@traceinfo{tabular started}%
2578 \begingroup%
2579 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
2580 \global\boolfalse{LWR@startedrow}%
```

Not yet doing an hline:

```
2581 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2582 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2583 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```
2584 \global\booltrue{LWR@intabularmetadata}%
2585 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2586 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
2587 \LWR@stoppars%
```

Track column `#` for setting text-align:

```
2588 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2589 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
2590 \let\\LWR@tabularendofline%
```

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:

```
2591 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
2592 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
2593 \NewDocumentCommand{\cmidrule}{o d() m}%
2594 {\LWR@docmidrule{##1}{##2}{##3}\LWR@getmynexttoken}%
2595 \RenewDocumentCommand{\cline}{m}%
2596 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2597 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2598 \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`.

```
2599 \let\multicolumn\LWR@tabledatamulticolumntag%
2600 \let\multirow\LWR@tabledatamultirowtag%
2601 \renewcommand*{\mrowcell}{\global\booltrue\LWR@skippingmrowcell}}%
2602 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```
2603 \global\boolfalse\LWR@tableparcell}%
2604 \global\boolfalse\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```
2605 \LWR@getmynexttoken%
2606 }%
```

Ending the environment:

```
2607 {%
2608 \LWR@closetabledatacell%
2609 \LWR@htmlblocktag{/tr}%
2610 \LWR@htmlblocktag{/table}%
2611 \global\boolfalse\LWR@intabularmetadata}%

```

```

2612 \addtocounter{LWR@tabulardepth}{-1}%
2613 \endgroup%
2614 }

2615 \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: 2616 \begin{warpHTML}

50.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
2617 \newcommand*{\@currentlabelname}{}

```

`\LWR@stripperperiod` $\{ \langle text \rangle \} [\langle . \rangle]$

Removes a trailing period.

```
2618 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

`\LWR@setlatestname` $\{ \langle object name \rangle \}$

Removes `\label`, strips any final period, and remembers the result.

```
2619 \newcommand*{\LWR@setlatestname}[1]{%

```

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.	

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `getttitlestring`.

```
2620 \GetTitleStringExpand{#1}%
2621 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
2622 \edef\@currentlabelname{%
2623 \expandafter\LWR@stripperperiod\@currentlabelname%
2624 \ltx@empty.\ltx@empty\@nil%
2625 }%
2626 }
```

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`:

```
2627 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2628 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2629 \zref@newprop{zLWR@htmlfilenumber}{%
2630 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}%
2631 }%
```

Additional properties for `lateximages`:

```
2632 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2633 \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`:

```
2634 \zref@addprop{special}{zLWR@name}
2635 \zref@addprop{special}{zLWR@htmlfilenumber}
2636 \zref@addprop{special}{zLWR@lateximagedepth}
2637 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
2638 \newcommand*{\LWR@spref}[2]{%
2639 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```
2640 \newcommand*{\LWR@nameref}[1]{%
2641 \LWR@spref{#1}{z\LWR@name}%
2642 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```
2643 \newcommand*{\LWR@htmlfileref}[1]{%
2644 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
2645 \LWR@spref{#1}{z\LWR@htmlfilenumber}%
2646 }
```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the lateximagedepth for this label:

```
2647 \newcommand*{\LWR@lateximagedepthref}[1]{%
2648 \LWR@spref{#1}{z\LWR@lateximagedepth}%
2649 }
```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the lateximagenumber for this label:

```
2650 \newcommand*{\LWR@lateximagenumberref}[1]{%
2651 \LWR@spref{#1}{z\LWR@lateximagenumber}%
2652 }
```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```
2653 \newcommand*{\LWR@splabel}[1]{%
2654 \LWR@setlatestname{\@currentlabelname}%
2655 \zref@labelbylist{#1}{special}}
```

50.3 Labels

`\LWR@subsublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

```
2656 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```

2657 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
2658 {}%
2659 {% not lateximage

```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

2660 \ifbool{LWR@doingstartpars}%
2661 {% pars allowed
2662 \ifbool{LWR@doingapar}
2663 {% par started
2664 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2665 }% par started
2666 {% par not started
2667 \LWR@stoppars%
2668 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2669 \LWR@startpars%
2670 }% par not started
2671 }% pars allowed
2672 {% pars not allowed
2673 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2674 }% pars not allowed
2675 }% not lateximage
2676 }

```

`\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`.

`clevereref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```

2677 \NewDocumentCommand{\LWR@newlabel}{m o}{%
2678 \LWR@traceinfo{LWR@newlabel: !#1!}%
2679 % \@bsphack%

```

Create a traditional LaTeX label, as modified by `cleveref`:

```

2680 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

2681 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2682 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
2683 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
2684 \LWR@splabel{#1}%
2685 \LWR@subsublabel{#1}%

```

```

2686 % \@esphack%
2687 \LWR@traceinfo{LWR@newlabel: done}%
2688 }

```

50.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

2689 \newcommand*{\LWR@startref}[1]
2690 {%
2691 \edef\LWR@lhref{\LWR@lateximagedepthref{#1}}
2692 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

2693 \LWR@htmltag{a href="%
2694 \LWR@traceinfo{LWR@startref B}%
2695 \LWR@htmlrefsectionfilename{#1}%
2696 \LWR@traceinfo{LWR@startref C}%
2697 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

2698 \LWR@traceinfo{LWR@startref D: !#1!}%
2699 \ifthenelse{\equal{\LWR@lhref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

2700 {%
2701 \LWR@traceinfo{LWR@startref D0: ??}
2702 ??}%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

2703 {%
2704 \LWR@traceinfo{LWR@startref D1: \LWR@lhref}%
2705 \ifthenelse{\cnttest{\LWR@lhref}{>}{0}}%
2706 {%
2707 \LWR@traceinfo{LWR@startref D2: \LWR@lhref}%
2708 lateximage\LWR@lateximagenumberref{#1}%

```

```

2709 }%
2710 {%
2711 \LWR@traceinfo{LWR@startref D3}%
2712 #1%
2713 }%
2714 }%
2715 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

2716 "{}}%
2717 \LWR@traceinfo{LWR@startref F}%
2718 }

```

`\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)”.

```

2719 \NewDocumentCommand{\LWR@subnewref}{m m}{%
2720 \LWR@traceinfo{LWR@subnewref #1 #2}%
2721 \LWR@startref{#1}%
2722 \LWR@origref{#2}%
2723 \LWR@htmltag{/a}%
2724 }

```

`\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.

```

2725 \NewDocumentCommand{\LWR@newref}{s m}{%
2726 \LWR@traceinfo{LWR@newref #2}%
2727 \IfBooleanTF{#1}%
2728 {\LWR@origref{#2}}%
2729 {\LWR@subnewref{#2}{#2}}%
2730 }

```

`\pagerefPageFor` Text for starred page references.

```

2731 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` * $\{\langle label \rangle\}$ Create an internal document reference, or just the unlinked number if starred, per hyperref.

```

2732 \NewDocumentCommand{\LWR@newpageref}{s m}{%

```

```

2733 \IfBooleanTF{#1}%
2734 {(\pagerefPageFor\LWR@origref{#2})}%
2735 {(\cpageref{#2})}%
2736 }

```

`\nameref` $\{\langle label \rangle\}$

```

2737 \newcommand*{\nameref}[1]{%
2738 \LWR@traceinfo{nameref A}%
2739 \LWR@startref{#1}%
2740 \LWR@traceinfo{nameref B}%
2741 \LWR@nameref{#1}%
2742 \LWR@traceinfo{nameref C}%
2743 \LWR@htmltag{/a}%
2744 \LWR@traceinfo{nameref D}%
2745 }

```


`\Nameref` $\{\langle label \rangle\}$ In print, adds the page number. In HTML, does not.

```


2746 \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.

```

2747 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2748 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.

```

Create a link with a text name:

`\LWR@subhyperref` $\{\langle URL \rangle\} \{\langle text \rangle\}$

```

2749 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2750 \LWR@htmltag{a href="{#1}" target="_{blank}"\LWR@orignewline}#2\LWR@htmltag{/a}%

```

```
2751 \LWR@ensuredoingapar%
2752 }
```

`\LWR@subhyperrefclass` $\{\langle URL \rangle\} \{\langle text \rangle\} \{\langle htmlclass \rangle\}$

```
2753 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2754 \LWR@htmltag{a href="{#1}"
2755 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
2756 \LWR@ensuredoingapar%
2757 }
```

`\href` $[\langle options \rangle] \{\langle URL \rangle\} \{\langle text \rangle\}$

Create a link with accompanying text:

```
2758 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
2759 \LWR@subhyperref{#2}{#3}%
2760 \endgroup%
2761 \LWR@ensuredoingapar%
2762 }
2763
2764 \newcommand{\href}{%
2765 \LWR@ensuredoingapar%
2766 \begingroup%
2767 \catcode'\_ =12
2768 \LWR@hrefb%
2769 }
```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```
2770 \newcommand*\LWR@nolinkurlb{[1]{#1\endgroup\LWR@ensuredoingapar}
2771
2772 \newcommand{\nolinkurl}{%
2773 \LWR@ensuredoingapar%
2774 \begingroup\catcode'\_ =12
2775 \LWR@nolinkurlb%
2776 }
```

`\url` $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link:

```
2777 \newcommand*\LWR@urlb{[1]{%
2778 \href{#1}{#1}%
2779 \endgroup%
2780 }
```

```
2780 \LWR@ensuredoingapar%
2781 }
2782
2783 \newcommand{\url}{%
2784 \LWR@ensuredoingapar%
2785 \begingroup\catcode'\_ =12
2786 \LWR@urlb%
2787 }
```

`\LWR@subinlineimage` [`<alttag>`] [`<class>`] [`<filename>`] [`<extension>`] [`<style>`]

```
2788 \newcommand*{\LWR@subinlineimage}[5][]{%
2789 \ifthenelse{\equal{#1}{}}{%
2790 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
2791 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
2792 }

2793 \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: 2794 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
2795 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2796 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
2797 \addtocounter{LWR@thisfloat}{1}%
2798 \booltrue{LWR@freezethisfloat}%
```

```
2799 \begingroup
```

Settings while inside the environment:

```
2800 \LWR@origraggedright
```

Open an HTML figure tag:

```
2801 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
2802 \renewcommand*{\@capttype}{#1}
2803 \caption@settype{#1}
2804 \LWR@startpars
2805 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
2806 \let\@float\LWR@floatbegin
2807 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
2808 \newcommand*{\LWR@floatend}{%
2809 \LWR@stoppars%
2810 \LWR@htmllementend{figure}%
2811 \endgroup%
2812 \boolfalse{LWR@freezethisfloat}%
2813 \LWR@startpars%
2814 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

2815 `\let\end@float\LWR@floatend`
 2816 `\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.

2817 `\newcounter{LWR@thisfloat}`

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

2818 `\newbool{LWR@freezethisfloat}`
 2819 `\boolfalse{LWR@freezethisfloat}`

`\LWR@maybeinthisfloat`

2820 `\newcommand*{\LWR@maybeinthisfloat}{%`
 2821 `\ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%`
 2822 `}`

`\@capttype` Remembers which float type is in use.

2823 `\newcommand*{\@capttype}{}`

51.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

2824 `\AtBeginDocument{\providecommand*\CaptionSeparator}{:~}}`

`\@makecaption` `{\langle name and num \rangle}{\langle text \rangle}`

Prints the float type and number, the caption separator, and the caption text.

2825 `\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.

```
2826 \newcounter{LWR@nextautofloat}
2827 \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.

```
2828 \newcommand*{\LWRsetnextfloat}[2]{%
2829 \setcounter{LWR@nextautopage}{#1}%
2830 \setcounter{LWR@nextautofloat}{#2}%
2831 }
```

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.

```
2832 \newcounter{LWR@latestautopage}% updated each new HTML file
2833 \setcounter{LWR@latestautopage}{1}
```

```
2834 \let\LWR@origcaption@begin\caption@begin
2835 \let\LWR@origcaption@end\caption@end
2836 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2837 \newcommand{\LWR@caption@begin}
2838 {
2839 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
2840 \begingroup%
```

The `caption` code was not allowing the closing `par` tag:

```
2841 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
2842 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
2843 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
2844 \LWR@htmlblocktag{figcaption}%
2845 \LWR@origcaption@begin%
2846 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
2847 \newcommand{\LWR@caption@end}
2848 {%
2849 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
2850 \vspace*{\baselineskip}%
```

Closing tag:

```
2851 \LWR@htmlblocktag{/figcaption}%
2852 \endgroup%
2853 % \leavevmode% avoid bad space factor (0) error
2854 \LWR@traceinfo{\LWR@caption@end: done}%
2855 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
2856 \AtBeginDocument{
2857 \let\caption@begin\LWR@caption@begin
2858 \let\caption@end\LWR@caption@end
2859 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
2860 \let\LWR@origcaptionlistentry\captionlistentry
2861
2862 \renewcommand*{\captionlistentry}{%
2863 \LWR@maybeinthisfloat%
2864 \LWR@ensuredoingapar%
2865 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2866 \LWR@origcaptionlistentry%
2867 }
```

```

2868
2869 \def\LWR@LTcaptionlistentry{%
2870 \LWR@ensuredoingapar%
2871 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2872 \bgroup
2873 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
2874 {\egroup\LWR@LT@captionlistentry}}%
2875 \def\LWR@LT@captionlistentry#1{%
2876 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autofloat before each float's entry. No changes if writing `.toc`. For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

```

2877 \let\LWR@origaddcontentsline\addcontentsline
2878
2879 \renewcommand*{\addcontentsline}[3]{%
2880 \ifthenelse{\equal{#1}{toc}}{ }{%
2881 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
2882 \addtocontents{\@nameuse{ext@#2}}{%
2883 \protect\LWRsetnextfloat%
2884 {\arabic{LWR@latestautopage}}%
2885 {\arabic{LWR@thisfloat}}%
2886 }% addtocontents
2887 }% not toc
2888 \LWR@origaddcontentsline{#1}{#2}{#3}%
2889 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

2890 \AtBeginDocument{
2891 \let\LWR@origcaptionof\captionof
2892
2893 \renewcommand*{\captionof}{%
2894 \LWR@maybeinthisfloat%
2895 \LWR@stoppars
2896 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2897 \LWR@origcaptionof%
2898 }
2899 }

2900 \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: 2901 `\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.

```
2902 \newcommand*{\LWR@myshorttoc}[1]{
2903 \LWR@ensuredoingapar
```

Only if the file exists:

```
2904 \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.

```
2905 \begingroup
2906 % \ifxetexorluatex%
2907 % \else
2908 % \inputencoding{latin1}% currently disabled
2909 % \fi
2910 \makeatletter
```

Read in the TOC file:

```
2911 \@input{\jobname.#1}
2912 % \makeatother
2913 \endgroup
2914 }%
2915 {}%
2916 }
```

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
2917 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
2918 \@ifundefined{chapter}
2919 {\LWR@closeprevious{\LWR@depthsection}}
2920 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
2921 \LWR@printpendingnotes
```

Place the list into its own chapter (if defined) or section:

```
2922 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
2923 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
2924 \LWR@myshorttoc{#1}
```

Close the nav:

```
2925 \LWR@htmlelementclassend{nav}{#1}
2926 }
```

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
2927 \let\LWR@orig@starttoc\@starttoc
2928
2929 \renewcommand{\@starttoc}[1]{
```



```

2930 \LWR@html elementclass{nav}{#1}
2931 \LWR@orig@starttoc{#1}
2932 \LWR@html elementclassend{nav}{#1}
2933 }

```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

2934 \let\LWR@origtableofcontents\tableofcontents
2935 \let\LWR@origlistoffigures\listoffigures
2936 \let\LWR@origlistoftables\listoftables
2937 \renewcommand*{\tableofcontents}{%

```

Copy the `.toc` file to `.sidetoc` for printing the `sidetoc`. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

2938 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
2939 \LWR@printpendingnotes%
2940 \LWR@origtableofcontents%
2941 }
2942 \renewcommand*{\listoffigures}{\LWR@printpendingnotes\LWR@origlistoffigures}
2943 \renewcommand*{\listoftables}{\LWR@printpendingnotes\LWR@origlistoftables}

```

52.2 High-level TOC commands

`\listof` `{<type>}` `{<title>}`

Emulate the `\listof` command from the `float` package (section 89). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```

2944 \NewDocumentCommand{\listof}{m +m}{%
2945 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
2946 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
2947 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
2948     \jobname.\csuse{ext@#1}\relax
2949 }

```

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.

2950 `\end{warpHTML}`

for HTML & PRINT: 2951 `\begin{warpall}`

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

2952 `\newcounter{SideTOCDepth}`

2953 `\setcounter{SideTOCDepth}{1}`

`\sidetocname` Holds the default name for the sidetoc.

2954 `\newcommand{\sidetocname}{Contents}`

2955 `\end{warpall}`

for HTML output: 2956 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

2957 `\newcommand*{\LWR@sidetoc}{`

2958 `\LWR@stoppars`

2959

The entire sideTOC is placed into a `nav` of class `sidetoc`.

2960 `\LWR@htmlclass{nav}{sidetoc}`

2961

2962 `\setcounter{tocdepth}{\value{SideTOCDepth}}`

2963

The title is placed into a `div` of class `sidetoctitle`, and may contain paragraphs.

2964 `\begin{BlockClass}{sidetoctitle}`

2965 `\sidetocname`

2966 `\end{BlockClass}`

The table of contents is placed into a div of class `sidetoccontents`.

```
2967 \begin{BlockClass}{sidetoccontents}
2968 \LinkHome
2969
2970 \LWR@myshorttoc{sidetoc}
2971 \end{BlockClass}
2972 \LWR@html-elementclassend{nav}{sidetoc}
2973 }
```

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:

```
2974 \renewcommand*\numberline[1]{%
2975 \LWR@sectionnumber{#1}%
2976 }
```

`\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

```
2977 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to `tocdepth`:

```
2978 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
2979 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```

2980 \LWR@subhyperrefclass{%
2981 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
2982 \LWR@stoppars%
2983 }
2984 {}
2985 }

```

Ctrl **lofdepth** TOC depth for figures.

```

2986 \newcounter{lofdepth}
2987 \setcounter{lofdepth}{1}

```

Ctrl **lotdepth** TOC depth for tables.

```

2988 \newcounter{lotdepth}
2989 \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

```

2990 \newcommand{\hypertocfloat}[5]{%
2991 \LWR@startpars

```

If some float-creation package has not yet defined the float type's **lofdepth** counter, etc, define it here:

```

2992 \@ifundefined{c@#3depth}{%
2993 \newcounter{#3depth}%
2994 \setcounter{#3depth}{1}%
2995 }{}%

```

Respond to **lofdepth**, etc.:

```

2996 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
2997 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{%
2998 \LWR@startpars%

```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```

2999 \LWR@subhyperrefclass{%
3000 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
3001 \#autofloat-\arabic{LWR@nextautofloat}}%
3002 {#4}{toc#2}%
3003 \LWR@stoppars%
3004 }{}%
3005 }

```

Automatically called by \contentsline:

```

3006 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
3007 \DeclareDocumentCommand{\l@chapter}{m m}
3008 {\hypertoc{0}{chapter}{#1}{#2}}
3009 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
3010 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
3011 \renewcommand{\l@subsubsection}[2]
3012 {\hypertoc{3}{subsubsection}{#1}{#2}}
3013 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
3014 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
3015 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
3016 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

3017 \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: 3018 \begin{warpHTML}

\LWR@indexsection Controls whether the index will be in a section or a chapter, depending on the documentclass.

```

3019 \@ifundefined{chapter}
3020 {\newcommand*\LWR@indexsection{\section{\indexname}}}
3021 {\newcommand*\LWR@indexsection{\chapter{\indexname}}}

```

\printindex

```

3022 \let\LWR@origprintindex\printindex
3023
3024 \renewcommand*{\printindex}
3025 {
3026 \LWR@indexsection
3027 \LWR@startpars
3028 \LWR@origprintindex
3029 }

```

Env **theindex**

```

3030 \renewenvironment*{theindex}{%
3031 \let\item\LWR@indexitem%
3032 \let\subitem\LWR@indexsubitem%
3033 \let\subsubitem\LWR@indexsubsubitem%
3034 }{}

```

\LWR@indexitem

```

3035 \newcommand{\LWR@indexitem}{
3036
3037 \InlineClass{indexitem}{}
3038 }

```

\LWR@indexitem

```

3039 \newcommand{\LWR@indexsubitem}{
3040
3041 \InlineClass{indexsubitem}{}
3042 }

```

\LWR@indexitem

```

3043 \newcommand{\LWR@indexsubsubitem}{
3044
3045 \InlineClass{indexsubsubitem}{}
3046 }

```

\hyperindexref $\{\langle autosecnumber \rangle\}$

\hyperindexref{web address} is inserted into *.ind by the xindy style file
lwarp_html.xdy

```

3047 \newcommand*{\hyperindexref}[1]{
3048 \LWR@htmltag{a href="%

```

Create an HTML reference to the autosection:

```
3049 \LWR@htmlrefsectionfilename{autopage-#1}\#autosec-#1"{}%}
```

Add the section number and section name, then close the tag:

```
3050 \LWR@origref{autopage-#1}\enskip\LWR@nameref{autopage-#1}%
3051 \LWR@htmltag{/a}%
3052 }
```

```
3053 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```
3054 \begin{warpprint}
3055 \newcommand{\hyperindexref}[1]{#1}
3056 \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 146 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: 3057 `\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.

```
3058 \let\LWR@origtextdollar\$
3059
3060 \renewcommand*{\$}{\%
3061 \LWR@htmltagc{span}%
3062 \LWR@origtextdollar%
3063 \LWR@htmltagc{/span}%
3064 }
```

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
3065 \newcounter{LWR@externalfilecnt}
```

54.1 Inline and display math

```
3066 \let\LWR@origdollar=$
3067 \let\secondorigdollar=$% balance for editor syntax highlighting

3068 \let\LWR@origopenparen\(  
3069 \let\LWR@origcloseparen\)
```

`$` Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
`$$`

```

3070 \begingroup
3071 \catcode'\$=\active%
3072 \protected\def$\{@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

3073 \gdef\LWR@doubledollar$#1$$#{
3074 \ifbool{mathjax}%

```

For MathJax, print the math between `\[` and `\]`:

```

3075 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

3076 {% not mathjax
3077
3078 \begin{lateximage}%
3079 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
3080 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
3081 \end{lateximage}
3082
3083 }
3084 }%

```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

3085 \gdef\LWR@singledollar#1$#{%
3086 \ifbool{mathjax}%

```

For MathJax, print the math between `\(` and `\)`:

```

3087 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

3088 {% not mathjax
3089 \begin{lateximage}%
3090 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
3091 \LWR@origdollar#1\LWR@origdollar%
3092 \end{lateximage}%
3093 }%
3094 }%

```

`\(` Redefine to the above dollar macros.

`\)`

```

3095 \gdef\(#1\){$#1$}
3096 \gdef\[#1\]{$$#1$$}

```

```
3097
3098 \endgroup
```

Remove the old `math` and `displaymath` environments:

```
3099 \let\math\relax
3100 \let\endmath\relax
3101 \let\displaymath\relax
3102 \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`.

```
3103 \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`.

```
3104 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}
```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```
3105 \AtBeginDocument{\catcode'\$=\active}
```

54.2 MathJax support

Ctr **LWR@nextequation** Used to add one to compute the next equation number.

```
3106 \newcounter{LWR@nextequation}
```

\LWR@syncmathjax Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\($\)$ ” characters. They are printed to HTML output, not interpreted by \LaTeX .

```
3107 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
3108 \ifcsdef{thechapter}{
3109 \BlockClassSingle{hidden}{
3110 \textbackslash(
```

```

3111 \textbackslash{}seteqsection \{\thechapter\}
3112 \textbackslash)
3113 }
3114 }
3115 {}% not using chapters
3116

```

MathJax doesn't allow setting the equation number to 1:

```

3117 \ifthenelse{\cnttest{\value{equation}}>0}
3118 {

```

Tell MathJax that the next set of equations begins with the current L^AT_EX equation number, plus one.

```

3119 \setcounter{LWR@nextequation}{\value{equation}}
3120 \addtocounter{LWR@nextequation}{1}

```

Place the MathJax command inside “\(>” and “\)>” characters, to be printed to HTML, not interpreted by L^AT_EX.

```

3121 \BlockClassSingle{hidden}{
3122 \textbackslash(
3123 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
3124 \textbackslash)>
3125 }
3126 }{}% not eq > 1
3127 }

```

`\LWR@restoremathlatexformatting` While producing math, use regular L^AT_EX formatting instead of HTML tags.

```

3128 \newcommand*{\LWR@restoremathlatexformatting}{%
3129 \let\hspace\LWR@origspace%
3130 \let\rule\LWR@origrule%
3131 \let\,\LWR@origcomma% disable HTML short unbreakable space
3132 \let\textit\LWR@origtextit%
3133 \let\textbf\LWR@origtextbf%
3134 \let\texttt\LWR@origtexttt%
3135 \let\textsc\LWR@origtextsc%
3136 \let\textsf\LWR@origtextsf%
3137 \let\textrm\LWR@origtextrm%
3138 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}}%
3139 \let\textsuperscript\LWR@origtextsuperscript%
3140 \let\textsubscript\LWR@origtextsubscript%
3141 \let~\LWR@origtilde%
3142 \let\enskip\LWR@origenskip%
3143 \let\quad\LWR@origquad%
3144 \let\qquad\LWR@origqquad%
3145 }

```

`\LWR@hidelatexequation` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the L^AT_EX version of the equation inside an HTML comment.

```
3146 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
3147 \LWR@stoppars
3148 \LWR@htmlopencomment
3149
```

Start the L^AT_EX math environment inside the HTML comment:

```
3150 \begingroup
3151 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their L^AT_EX meanings.

```
3152 \LWR@restoremathlatexformatting
```

See `\LWR@htmlmathlabel` in section 54.4.1.

Print the contents of the equation:

```
3153 #2
```

End the L^AT_EX math environment inside the HTML comment:

```
3154 \csuse{\LWR@origend#1}
3155 \endgroup
3156
```

Close the HTML comment and resume HTML paragraph handling:

```
3157 \LWR@htmlclosecomment
3158 \LWR@startpars
3159 }
```

`\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 L^AT_EX.

```
3160 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3161
```

Enclose the MathJax environment inside printed “\(" and “\)” characters.

```
3162 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
3163 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
3164 \textbackslash{}end\{#1\}
```

```
3165
```

```
3166 }
```

54.3 Equation environment

Remember existing `equation` environment:

```
3167 \let\LWR@origequation\equation
```

```
3168 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
3169 \let\equation\relax
```

```
3170 \let\endequation\relax
```

Env **equation** The new `equation` environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the \LaTeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
3171 \NewEnviron{equation}{%
```

```
3172
```

```
3173 \ifbool{mathjax}
```

MathJax output:

```
3174 {
```

Print commands to synchronize MathJax's equation number and format to the current \LaTeX chapter/section and equation number:

```
3175 \LWR@syncmathjax
```

Print the \LaTeX math inside an HTML comment:

```
3176 \LWR@hidelatexequation{equation}{\BODY}
3177 }
```

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`.

```
3178 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3179 \begin{lateximage}[({\theequation}) \textbackslash{begin}{equation}\}]%
3180 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
3181 \textbackslash{end}{equation}\}]% alt tag
```

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
3182 \LWR@origequation
3183 \BODY% contents collected by NewEnviron
3184 \LWR@origendequation
3185 \end{lateximage}%
3186 }
3187
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3188 }[\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 multiline-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 multiline environment and not an equation.

```
3189 \newbool{LWR@amsmultline}
3190 \boolfalse{LWR@amsmultline}
```

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

`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`.

```
3191 \newcommand*\LWR@htmlmathlabel[1]{%
3192 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3193 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
3194 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3195 \ifbool{\LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the \LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
3196 \LWR@htmlclosecomment%
3197 \LWR@origltx@label{#1}%
3198 \LWR@htmlopencomment%
3199 }% text
3200 }% mathjax
3201 {%
3202 \LWR@origltx@label{#1}%
3203 }%
3204 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
3205 \newcommand*\LWR@beginhideamsmath{%
3206 \LWR@stoppars
3207
3208 \LWR@htmlopencomment
3209
3210 \begingroup
3211 \LWR@restoremathlatexformatting
3212 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.


```

3213 \newcommand*{\LWR@endhideamsmath}{
3214 \endgroup
3215
3216 \LWR@htmlclosecomment
3217
3218 \LWR@startpars
3219 }

```

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} 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`

```

3220 \BeforeBeginEnvironment{multline}{
3221 \ifbool{mathjax}
3222 {
3223 \LWR@syncmathjax
3224 \booltrue{LWR@amsmultline}
3225 \LWR@beginhideamsmath
3226 }
3227 {
3228 \lateximage[LWRAMSMATHBODY]
3229 }
3230 }
3231
3232 \AfterEndEnvironment{multline}{
3233
3234 \ifbool{mathjax}
3235 {
3236 \LWR@endhideamsmath
3237 \boolfalse{LWR@amsmultline}
3238 \LWR@addmathjax{multline}{\the\@envbody}
3239 }
3240 {\endlateximage}
3241
3242 }

```

Env **multline***

```

3243 \BeforeBeginEnvironment{multline*}{
3244 \ifbool{mathjax}
3245 {
3246 \LWR@syncmathjax
3247 \booltrue{LWR@amsmultline}
3248 \LWR@beginhideamsmath
3249 }
3250 {
3251 \lateximage[LWRMSMATHBODY]
3252 }
3253 }
3254
3255 \AfterEndEnvironment{multline*}{
3256
3257 \ifbool{mathjax}
3258 {
3259 \LWR@endhideamsmath
3260 \boolfalse{LWR@amsmultline}
3261 \LWR@addmathjax{multline*}{\the\@envbody}
3262 }
3263 {\endlateximage}
3264
3265 }
3266

```

Env **gather**

```

3267 \BeforeBeginEnvironment{gather}{
3268 \ifbool{mathjax}
3269 {
3270 \LWR@syncmathjax
3271 \boolfalse{LWR@amsmultline}
3272 \LWR@beginhideamsmath
3273 }
3274 {
3275 \lateximage[LWRMSMATHBODY]
3276 }
3277 }
3278
3279 \AfterEndEnvironment{gather}{
3280
3281 \ifbool{mathjax}
3282 {
3283 \LWR@endhideamsmath
3284 \LWR@addmathjax{gather}{\the\@envbody}
3285 }
3286 {\endlateximage}

```

```
3287
3288 }
```

Env **gather***

```
3289 \BeforeBeginEnvironment{gather*}{
3290 \ifbool{mathjax}
3291 {
3292 \LWR@syncmathjax
3293 \boolfalse{LWR@amsmultline}
3294 \LWR@beginhideamsmath
3295 }
3296 {
3297 \lateximage[LWR@SMATHBODY]
3298 }
3299 }
3300
3301 \AfterEndEnvironment{gather*}{
3302
3303 \ifbool{mathjax}
3304 {
3305 \LWR@endhideamsmath
3306 \LWR@addmathjax{gather*}{\the\@envbody}
3307 }
3308 {\endlateximage}
3309
3310 }
```

Env **align**

```
3311 \BeforeBeginEnvironment{align}{
3312 \ifbool{mathjax}
3313 {
3314 \LWR@syncmathjax
3315 \boolfalse{LWR@amsmultline}
3316 \LWR@beginhideamsmath
3317 }
3318 {
3319 \lateximage[LWR@SMATHBODY]
3320 }
3321 }
3322
3323 \AfterEndEnvironment{align}{
3324
3325 \ifbool{mathjax}
3326 {
3327 \LWR@endhideamsmath
3328 \LWR@addmathjax{align}{\the\@envbody}
3329 }
```

```

3330 {\endlateximage}
3331
3332 }

```

Env **align***

```

3333 \BeforeBeginEnvironment{align*}{
3334 \ifbool{mathjax}
3335 {
3336 \LWR@syncmathjax
3337 \boolfalse{LWR@amsmultline}
3338 \LWR@beginhideamsmath
3339 }
3340 {
3341 \lateximage[LWR@SMATHBODY]
3342 }
3343 }
3344
3345 \AfterEndEnvironment{align*}{
3346
3347 \ifbool{mathjax}
3348 {
3349 \LWR@endhideamsmath
3350 \LWR@addmathjax{align*}{\the\@envbody}
3351 }
3352 {\endlateximage}
3353
3354 }

```

Env **flalign**

```

3355 \BeforeBeginEnvironment{flalign}{
3356 \ifbool{mathjax}
3357 {
3358 \LWR@syncmathjax
3359 \boolfalse{LWR@amsmultline}
3360 \LWR@beginhideamsmath
3361 }
3362 {
3363 \lateximage[LWR@SMATHBODY]
3364 }
3365 }
3366
3367 \AfterEndEnvironment{flalign}{
3368
3369 \ifbool{mathjax}
3370 {
3371 \LWR@endhideamsmath
3372 \LWR@addmathjax{flalign}{\the\@envbody}

```

```

3373 }
3374 {\endlateximage}
3375
3376 }

```

Env `flalign*`

```

3377 \BeforeBeginEnvironment{flalign*}{
3378 \ifbool{mathjax}
3379 {
3380 \LWR@syncmathjax
3381 \boolfalse{LWR@amsmultline}
3382 \LWR@beginhideamsmath
3383 }
3384 {
3385 \lateximage[LWR@SMATHBODY]
3386 }
3387 }
3388
3389 \AfterEndEnvironment{flalign*}{
3390
3391 \ifbool{mathjax}
3392 {
3393 \LWR@endhideamsmath
3394 \LWR@addmathjax{flalign*}{\the\@envbody}
3395 }
3396 {\endlateximage}
3397
3398 }

3399 \end{warpHTML}

```

55 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment starting on the preceeding page and ending 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.

Since the `\lateximage` is placed onto its own PDF page, which `pdftotext` will read and try to convert to text, an HTML open comment is placed onto the bottom of the current page, but it must appear below any footnotes on the current page, so a special page style (`LWR@commentfooterstyle`) is used.

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.

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: 3400 `\begin{warpHTML}`

`LWR@commentfooterstyle`

```
3401 \newpagestyle{LWR@commentfooterstyle}{
3402 \sethead{}{}{}
3403 \setfoot{\LWR@htmlopencomment}{}{}
3404 }
```

Ctrl LWR@lateximagenumber Sequence the images.

```
3405 \newcounter{LWR@lateximagenumber}
3406 \setcounter{LWR@lateximagenumber}{0}
```

Ctrl LWR@lateximagedepth Do not create `\lateximage` inside of `\lateximage`.

```
3407 \newcounter{LWR@lateximagedepth}
3408 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
3409 \ifcsdef{\LWR@file}{}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3410 \edef\LWR@hashmark{\string#} % for use in \write
3411 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl LWR@nextpage Used to count ahead to the next page:

```
3412 \newcounter{LWR@nextpage}

3413 \end{warpHTML}
```

for HTML & PRINT: 3414 `\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.

```
3415 \newcommand*{\LateximageFontSizeName}{large}
```

```
3416 \end{warpall}
```

for HTML output: 3417 \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...

```
3418 \newcommand{\LWR@HTMLsanitize}[1]{%
3419 \protect\StrSubstitute{\detokenize{#1}}{%
3420 {\detokenize{&}}}%
3421 {\detokenize{&}}[\LWR@strresult]%
3422 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3423 {\detokenize{<}}}%
3424 {\detokenize{<}}[\LWR@strresult]%
3425 [\LWR@strresult]%
3426 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3427 {\detokenize{>}}}%
3428 {\detokenize{>}}[\LWR@strresult]%
3429 [\LWR@strresult]%
3430 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3431 {\detokenize{##}}}%
3432 {\detokenize{##}}[\LWR@strresult]%
3433 [\LWR@strresult]%
3434 \LWR@strresult%
3435 }
```

```
\LWR@HTMLsanitizeexpand {\langle text \rangle}
```

This version expands the argument before sanitizing it.

```
3436 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3437 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
3438 {\detokenize{&}}}%
3439 {\detokenize{&}}[\LWR@strresult]%
3440 [\LWR@strresult]%
3441 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3442 {\detokenize{<}}}%
3443 {\detokenize{<}}[\LWR@strresult]%
3444 [\LWR@strresult]%
3445 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
3446 {\detokenize{>}}}%
3447 {\detokenize{>}}[\LWR@strresult]%
3448 [\LWR@strresult]%
3449 }
```

```

3449 \LWR@strresult%
3450 }

```

Env `lateximage` [*<alttag>*]

```

3451 \NewDocumentEnvironment{lateximage}{0{image}}{%
3452 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

3453 {%
3454 \addtocounter{LWR@lateximagedepth}{1}%
3455 }%

```

Otherwise, this is the outer-most lateximage:

```

3456 {% start of outer-most lateximage

```

Starting a new lateximage:

```

3457 \addtocounter{LWR@lateximagenumber}{1}%

```

While inside a lateximage, do not use mathjax:

```

3458 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

3459 \LWR@ensuredoingapar%

```

Next file:

```

3460 \addtocounter{LWR@externalfilecnt}{1}%

```

Figure out what the next page number will be:

```

3461 \setcounter{LWR@nextpage}{\thepage}%
3462 \addtocounter{LWR@nextpage}{1}%

```

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:

```

3463 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
3464 class="lateximagesource"{} } \LWR@orignewline

```

Write instructions to the `lateximages.txt` file:

```

3465 \immediate\write\LWR@file{|\theLWR@nextpage|\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.

```
3466 \LWR@origthispagestyle{\LWR@commentfooterstyle}%
3467 \addtocounter{\LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
3468 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
3469 \LWR@origminipage{6in}%
3470 \csuse{\LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
3471 \let\hspace\LWR@origspace%
3472 \let\rule\LWR@origrule%
3473 \let\,\LWR@origcomma% disable HTML short unbreakable space
3474 \let\textit\LWR@origtextit%
3475 \let\textbf\LWR@origtextbf%
3476 \let\texttt\LWR@origtexttt%
3477 \let\textsc\LWR@origtextsc%
3478 \let\textsf\LWR@origtextsf%
3479 \let\textrm\LWR@origtextrm%
3480 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3481 \let\textsuperscript\LWR@origtextsuperscript%
3482 \let\textsubscript\LWR@origtextsubscript%
3483 \let~\LWR@origtilde%
3484 \let\enskip\LWR@origenskip%
3485 \let\quad\LWR@origquad%
3486 \let\qquad\LWR@origqquad%
3487 \let\tabular\LWR@origtabular%
3488 \let\endtabular\LWR@origendtabular%
3489 \let\newline\LWR@orignewline%
3490 }% end of outer-most lateximage
3491 }% end of \begin{lateximage}
3492 {% start of \end{lateximage}
3493 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
3494 {%
3495 \addtocounter{\LWR@lateximagedepth}{-1}%
3496 }%
```

if this is the outer-most lateximage:

```
3497 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
3498 \LWR@origendminipage%
3499 \LWR@orignewpage%
3500 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```
3501 \LWR@htmlclosecomment{}\LWR@orignewline%
```

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` `multline`, `gather`, or `align` environments.

```
3502 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
3503 {%
3504 \LWR@subinlineimage[%
3505 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
3506 ]%
3507 {lateximage}%
3508 {lateximages\OSPathSymbol{lateximage-\theLWR@externalfilecnt}}%
3509 {svg}%
3510 {}%
3511 }%
3512 {%
3513 \LWR@subinlineimage[#1]{lateximage}%
3514 {lateximages\OSPathSymbol{lateximage-\theLWR@externalfilecnt}{svg}}}%
3515 }%
3516 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
3517 \LWR@ensuredoingapar%
```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
3518 \LWR@htmltag{/span}%
3519 \LWR@htmlcomment{End of lateximage}%
3520 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```

3521 \addtocounter{LWR@lateximagedepth}{-1}%
3522 }% end of outer-most lateximage
3523 }%
3524 \end{warpHTML}

```

for PRINT output: 3525 \begin{warpprint}
 3526 \newenvironment{lateximage}[1][\begin{group}]{\end{group}}
 3527 \end{warpprint}

56 center, flushleft, flushright

for HTML output: 3528 \begin{warpHTML}

Env center Replace center functionality with CSS tags:

```

3529 \renewenvironment*{center}
3530 {\BlockClass{center}}
3531 {\endBlockClass}

```

Env flushright

```

3532 \renewenvironment*{flushright}
3533 {\BlockClass{flushright}}
3534 {\endBlockClass}

```

Env flushleft

```

3535 \renewenvironment*{flushleft}
3536 {\BlockClass{flushleft}}
3537 {\endBlockClass}

```

```

3538 \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: 3539 \begin{warpHTML}

Options for siunitx:

```

3540 \PassOptionsToPackage{
3541 detect-mode=true,
3542 per-mode=symbol,% fraction is not seen by pdftotext
3543 text-celsius = {\HTMLentity{deg}C},
3544 text-degree = {\HTMLentity{deg}},
3545 }{siunitx}

3546 \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: 3547 `\begin{warpHTML}`

58.1 `\graphicspath`

`\graphicspath` $\{\langle path \rangle\}$

```
3548 \newcommand*\thisgraphicspath{{}}
3549 \renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{#1}}
```

`\DeclareGraphicsExtensions` $\{\langle list \rangle\}$

`\DeclareGraphicsRule` $\{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\} \{\langle \rangle\}$

```
3550 \renewcommand*\DeclareGraphicsExtensions[1]{%
3551 \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.

```
3552 \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.

```
3553 \newlength{\LWR@igwidth}
3554 \newlength{\LWR@igheight}
3555 \newcommand*\LWR@igwidthstyle{}
3556 \newcommand*\LWR@igheightstyle{}
3557 \newcommand*\LWR@igorigin{}
3558 \newcommand*\LWR@igangle{}
3559 \newcommand*\LWR@igxscale{1}
3560 \newcommand*\LWR@igyscale{1}
3561 \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:

```
3562 \define@key{igraph}{width}{%
3563 \setlength{\LWR@igwidth}{#1}%
3564 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3565 {%
```

Default to use the converted fixed length given:

```
3566 \uselengthunit{PT}%
3567 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
3568 \IfEndWith{#1}{ex}%
3569 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
3570 {}% not ex
3571 \IfEndWith{#1}{em}%
3572 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
3573 {}% not em
3574 \IfEndWith{#1}{\}%
3575 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
3576 {}% not percent
3577 \IfEndWith{#1}{px}%
3578 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
3579 {}% not px
3580 }{}% end of length > Opt
3581 }
```

If an optional height was given, set an HTML style:

```
3582 \define@key{igraph}{height}{%
3583 \setlength{\LWR@igheight}{#1}%
3584 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
3585 {%
```

Default to use the converted fixed length given:

```
3586 \uselengthunit{PT}%
3587 \renewcommand*{\LWR@igheightstyle}{%
3588 height:\rndprintlength{\LWR@igheight} %
3589 }%
```

If ex or em dimensions were given, use those instead:

```
3590 \IfEndWith{#1}{ex}%
3591 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
3592 {}% not ex
3593 \IfEndWith{#1}{em}%
3594 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
3595 {}% not em
3596 \IfEndWith{#1}{\}%
3597 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
3598 {}% not percent
3599 \IfEndWith{#1}{px}%
3600 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
```

```

3601 {}% not px
3602 {}% end of length > 0pt
3603 }

```

Handle origin key:

```

3604 \define@key{igraph}{origin}{%
3605 \renewcommand*{\LWR@igorigin}{#1}%
3606 }

```

Handle angle key:

```

3607 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

3608 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
3609

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

3610 \define@key{igraph}{scale}{%
3611 \renewcommand*{\LWR@igxscale}{#1}%
3612 \renewcommand*{\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

3613 \define@key{igraph}{bb}{}
3614 \define@key{igraph}{bblly}{}
3615 \define@key{igraph}{bbllx}{}
3616 \define@key{igraph}{bburx}{}
3617 \define@key{igraph}{bbury}{}
3618 \define@key{igraph}{natwidth}{}
3619 \define@key{igraph}{natheight}{}
3620 \define@key{igraph}{hiresbb}{}
3621 \define@key{igraph}{viewport}{}
3622 \define@key{igraph}{trim}{}
3623 \define@key{igraph}{totalheight}{}
3624 \define@key{igraph}{keepaspectratio}{}
3625 \define@key{igraph}{clip}{}
3626 \define@key{igraph}{draft}{}
3627 \define@key{igraph}{type}{}
3628 \define@key{igraph}{ext}{}
3629 \define@key{igraph}{read}{}
3630 \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.

```
3631 \newcommand*{\LWR@rotstyle}[2]{%
3632   #1transform:rotate(-#2deg);
3633 }
```

\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.

```
3634 \newcommand*{\LWR@scalestyle}[3]{%
3635   #1transform:scale(#2,#3);
3636 }
```

58.3 \includegraphics

Bool **LWR@infloatrow** Used to compute `\linewidth`.

```
3637 \newbool{LWR@infloatrow}
3638 \boolfalse{LWR@infloatrow}

3639 \newcommand*{\LWR@imageextension}{}
3640 \newcommand*{\LWR@expgraphicsfilename}{}

```

\LWR@includegraphicsb $[\langle options \rangle] \{\langle filename \rangle\}$

```
3641
3642 \newcommand*{\LWR@includegraphicsb}[2] []
3643 {%

```

Start the image tag on a new line, allow PDF output word wrap:

```
3644 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
3645 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3646 \ifbool{LWR@infloatrow}%

```



```

3647 {}
3648 {% not in a minipage or a floatrow:
3649 \setlength{\linewidth}{6in}%
3650 \setlength{\textwidth}{6in}%
3651 \setlength{\textheight}{9in}%
3652 }%
3653 }{}%

```

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.

```

3654 \edef\LWR@expgraphicsfilename{#2}
3655 \renewcommand*{\LWR@imageextension}{}%
3656 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
3657 {\renewcommand*{\LWR@imageextension}{.jpg}}{}%
3658 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
3659 {\renewcommand*{\LWR@imageextension}{.JPG}}{}%
3660 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
3661 {\renewcommand*{\LWR@imageextension}{.png}}{}%
3662 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
3663 {\renewcommand*{\LWR@imageextension}{.PNG}}{}%
3664 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
3665 {\renewcommand*{\LWR@imageextension}{.gif}}{}%
3666 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
3667 {\renewcommand*{\LWR@imageextension}{.GIF}}{}%
3668 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
3669 {\renewcommand*{\LWR@imageextension}{.svg}}{}%
3670 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
3671 {\renewcommand*{\LWR@imageextension}{.SVG}}{}%
3672 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
3673 {\renewcommand*{\LWR@imageextension}{.svgz}}{}%
3674 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
3675 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also `\detokenize .pdf` and `.svg` comparison strings.

```

3676 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
3677 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
3678 %
3679 \StrSubstitute{\LWR@strresult}%
3680 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

3681 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```
3682 \setlength{\LWR@igwidth}{Opt}%
3683 \setlength{\LWR@igheight}{Opt}%
3684 \renewcommand*\LWR@igwidthstyle{}%
3685 \renewcommand*\LWR@igheightstyle{}%
3686 \renewcommand*\LWR@igorigin{}%
3687 \renewcommand*\LWR@igangle{}%
3688 \renewcommand*\LWR@igxscale{1}%
3689 \renewcommand*\LWR@igyscale{1}%
3690 \renewcommand*\LWR@igclass{inlineimage}%
```

Options are in #1

```
3691 \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.

```
3692 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
3693 {% start of href
3694 \LWR@htmltag{% start of image tags
3695 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3696 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline
```

Only include a style tag if a width, height, angle, or scale was given:

```
3697 \ifthenelse{
3698 \NOT\equal{\LWR@igwidthstyle}{ } \OR
3699 \NOT\equal{\LWR@igheightstyle}{ } \OR
3700 \NOT\equal{\LWR@igorigin}{ } \OR
3701 \NOT\equal{\LWR@igangle}{ } \OR
3702 \NOT\equal{\LWR@igxscale}{1} \OR
3703 \NOT\equal{\LWR@igyscale}{1}
3704 }%
3705 {\LWR@origtilde{} style="%
3706 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
3707 {\LWR@igwidthstyle;}%
3708 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
3709 {\LWR@igheightstyle;}%
3710 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
3711 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{ }%
3712 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
3713 {%
3714 \LWR@rotstyle{-ms-}{\LWR@igangle}%
3715 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
```

```

3716 \LWR@rotstyle{}\LWR@igangle%
3717 }}{}%
3718 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
3719 \NOT\equal{\LWR@igyyscale}{1}}%
3720 {\LWR@scalestyle{-ms-}\LWR@igxscale}\LWR@igyyscale}%
3721 \LWR@scalestyle{-webkit-}\LWR@igxscale}\LWR@igyyscale}%
3722 \LWR@scalestyle{}\LWR@igxscale}\LWR@igyyscale}}{}%
3723 " \LWR@orignewline}}{}%

```

Set the class:

```

3724 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
3725 }% end of image tags
3726 }% end of href
3727 \endgroup

```

Return to small-sized output:

```

3728 \LWR@origscriptsize
3729 }

```

`\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.

```

3730 \renewcommand*\includegraphics{
3731 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

3732 \LWR@ensuredoingapar%
3733 \begin{group}%
3734 \LWR@includegraphicsb%
3735 }

```

```

3736 \end{warpHTML}

```

for PRINT output: For print output, accept and then discard the new `class` key:

```

3737 \begin{warpprint}
3738 \define@key{Gin}{class}{}
3739 \end{warpprint}

```

58.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 3740 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
3741 \newcommand*\LWR@rotboxorigin{}
```

\LWR@originname {\langle letter \rangle}

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
3742 \newcommand*\LWR@originname[1]{%
3743 \ifthenelse{\equal{#1}{t}}{top}{}%
3744 \ifthenelse{\equal{#1}{b}}{bottom}{}%
3745 \ifthenelse{\equal{#1}{c}}{center}{}%
3746 \ifthenelse{\equal{#1}{l}}{left}{}%
3747 \ifthenelse{\equal{#1}{r}}{right}{}%
3748 }
```

\LWR@originnames {\langle letters \rangle}

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```
3749 \newcommand*\LWR@originnames[1]{%
3750 \StrChar{#1}{1}[\LWR@strresult]%
3751 \LWR@originname{\LWR@strresult}%
3752 \StrChar{#1}{2}[\LWR@strresult]%
3753 \LWR@originname{\LWR@strresult}%
3754 }
```

Handle the origin key for \rotatebox:

```
3755 \define@key{krotbox}{origin}{%
3756 \renewcommand*\LWR@rotboxorigin{#1}%
3757 }
```

These keys are ignored:

```
3758 \define@key{krotbox}{x}{}
3759 \define@key{krotbox}{y}{}
3760 \define@key{krotbox}{units}{}

```

\rotatebox [*\langle keyval list \rangle*] {\langle angle \rangle} {\langle text \rangle}

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
3761 \NewDocumentCommand{\LWR@rotatebox}{0}{ m +m}{%
```

Reset the origin to “none-given”:

```
3762 \renewcommand*\LWR@rotboxorigin{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
3763 \setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
3764 \LWR@htmltagc{span style="display: inline-block;
```

If an origin was given, translate and print the origin information:

```
3765 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
3766 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}}%
```

Print the rotation information:

```
3767 \LWR@rotstyle{-ms-}{#2} %
3768 \LWR@rotstyle{-webkit-}{#2} %
3769 \LWR@rotstyle}{#2} %
3770 "{}\LWR@orignewline%
```

Print the text to be rotated:

```
3771 \begin{LWR@nestspan}%
3772 #3%
```

Close the span:

```
3773 \LWR@htmltagc{/span}%
3774 \end{LWR@nestspan}%
3775 }
```

`\scalebox` $\{\langle h-scale \rangle\} [\langle v-scale \rangle] \{\langle text \rangle\}$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
3776 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
3777 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
3778 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3779 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3780 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3781 "{}}%
```

Print the text to be scaled:

```
3782 \begin{LWR@nestspan}%
3783 #3%
```

Close the span:

```
3784 \LWR@htmltagc{/span}%
3785 \end{LWR@nestspan}%
3786 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
3787 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
3788 \end{warpHTML}
```

58.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.

for HTML output: 3789 `\begin{warpHTML}`

`\resizebox` $\{\langle h-length \rangle\} \{\langle v-length \rangle\} \{\langle text \rangle\}$

Simply prints its text argument.

```
3790 \renewcommand{\resizebox}[3]{#3}
3791 \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 192 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: 3792 `\begin{warptHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
3793 \AtEndPreamble{
3794 \RequirePackage{cleveref}
3795 \crefname{pagenote}{note}{notes}
3796 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
3797 \AfterEndPreamble{
```

```
\@@setcref {<kindofref>} {<label>}
```

```
3798 \renewcommand*\@@setcref[2]{#1{\ref{#2}}{}}{}}
```

```
\@@setcrefrange {<text>} {<label>} {<label>}
```

```
3799 \renewcommand*\@@setcrefrange[3]{%
3800 #1{\ref{#2}}{\ref{#3}}{}}{}}{}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
3801 \newcommand*\cpagerefFor{for}
```

`\@@setcpageref` $\{\langle typeofref \rangle\} \{\langle label \rangle\}$, where `typeofref` is “page” or “pages”

```
3802 \renewcommand*\@@setcpageref}[2]{%
3803 #1{\cpagerefFor\ \cref{#2}}{}{}%
3804 }

3805 \renewcommand{\@@setpagerefrange}[3]{%
3806 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}
3807 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
3808 \let\LWR@origlabel\label
3809 \let\label\LWR@newlabel
3810 \let\LWR@origref\ref
3811 \let\ref\LWR@newref% \end{ syntax highlighting
3812 \let\LWR@origpageref\pageref
3813 \let\pageref\LWR@newpageref
3814
3815
3816
3817 \end{warpHTML}
```

60 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 3818 `\begin{warpHTML}`



Env `picture`

```
3819 \BeforeBeginEnvironment{picture}{\lateximage}
3820
3821 \AfterEndEnvironment{picture}{\endlateximage}

3822 \end{warpHTML}
```

61 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

placement	Minipages and parboxes will be placed side-by-side in HTML unless you place a <code>\newline</code> 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 <code>inline-block</code> .
side-by-side	Side-by-side minipages may be separated by <code>\quad</code> , <code>\qquad</code> , <code>\enskip</code> , <code>\hspace</code> , <code>\hfill</code> , or a <code>\rule</code> . When inside a <code>center</code> 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 <code></code> . An HTML <code><div></code> cannot appear inside a <code></code> . While in a <code></code> , minipages and parboxes are ignored. Use <code>\newline</code> or <code>\par</code> for an HTML break.
size	When using <code>\linewidth</code> , <code>\textwidth</code> , and <code>\textheight</code> , widths and heights are scaled proportionally to a 6×9 inch text area.
no-width minipages	A minipage of width exactly <code>\linewidth</code> is automatically given no HTML width.
full-width minipages	A new macro <code>\minipagefullwidth</code> requests that the next minipage be generated without an HTML <code>width</code> 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 ^A T _E X PDF output they do not. Use a <code>flushleft</code> or similar environment in the child minipage to force a text alignment.
for HTML output:	3823 <code>\begin{warpHTML}</code>

61.1 Counters and lengths

Ctr	<code>LWR@minipagedepth</code>	Used to only reset the line width at the outermost minipage.
	3824 <code>\newcounter{LWR@minipagedepth}</code> 3825 <code>\setcounter{LWR@minipagedepth}{0}</code>	
Len	<code>\WR@minipagewidth</code>	Used to convert the width into printable units.
	3826 <code>\newlength{\LWR@minipagewidth}</code>	
Len	<code>\WR@minipageheight</code>	Used to convert the height into printable units.
	3827 <code>\newlength{\LWR@minipageheight}</code>	

Remember the original definitions:

```
3828 \let\LWR@origminipage\minipage
3829 \let\LWR@origendminipage\endminipage
```

61.2 Footnote handling

Also see section 35 for other forms of footnotes.

`\LWR@nullrule` [*raise*] {*width*} {*height*}

Used to cancel the footnotesep rule during HTML output. This rule was causing occasional end paragraph tags to be inserted for some (so far) undetermined reason, as well as adding purposeless code to the HTML output.

```
3830 \NewDocumentCommand{\LWR@nullrule}{o m m}{}
```

`\LWR@makefntext` {*text*}

Adds a paragraph closing tag to the end of each footnote.

```
3831 \newcommand{\LWR@makefntext}[1]{%
3832 \begingroup%
3833 \let\rule\LWR@nullrule%
3834 \textsuperscript{\@thefnmark} #1%
3835 \endgroup%
3836 \LWR@htmltagc{/\LWR@tagregularparagraph}%
3837 }
```

`\LWR@makefnmark` Creates an HTML superscript footnote mark:

```
3838 \newcommand*{\LWR@makefnmark}{\textsuperscript{\@thefnmark}}
```

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:

```
3839 \def\LWR@endminipage{%
3840   \par
3841   \unskip
3842   \ifvoid\@mpfootins\else
```

```

3843      \vskip\skip\@mpfootins
3844      \normalcolor
3845 \LWR@htmldivclass{mpfootnotes}
3846      \unvbox\@mpfootins
3847 \LWR@htmldivclassend{mpfootnotes}
3848      \fi
3849      \@minipagefalse
3850 \color@endgroup
3851 \egroup
3852 \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.

```

3853 \newcommand*{\LWR@subminipage}{%
3854 \LWR@stoppars
3855 \LWR@origminipage{6in}

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

3856 \LWR@origraggedright%

```

Minipage footnotes appear the bottom of the minipage.

```

3857 \let\footnote\LWR@origfootnote%
3858 \let\@makefnmark\LWR@makefnmark%
3859 \let\@makefnmark\LWR@makefnmark%

```

Resume paragraph tag handling for the contents of the minipage:

```

3860 \LWR@startpars%
3861 }

```

`\LWR@endsubminipage` Closes the subminipage.

```

3862 \newcommand*{\LWR@endsubminipage}{%
3863 \LWR@stoppars%
3864 \LWR@endminipage% The following empty line is required:
3865
3866 }

```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```

3867 \newbool{LWR@minipagefullwidth}
3868 \boolfalse{LWR@minipagefullwidth}

```

`\minipagefullwidth` Requests that the next minipage have no `width` tag in HTML:

for HTML output: 3869 `\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`
 3870 `\end{warpHTML}`

for PRINT output: 3871 `\begin{warpprint}`
 3872 `\newcommand*{\minipagefullwidth}{}`
 3873 `\end{warpprint}`

for HTML output: 3874 `\begin{warpHTML}`

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

3875 `\newbool{LWR@minipagethispar}`
 3876 `\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.

3877 `\RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}`
 3878 {%

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

3879 `\LWR@traceinfo{starting minipage of width #4}%`
 3880 `\uselengthunit{in}%`
 3881 `\setlength{\LWR@minipagewidth}{#4}%`
 3882 `\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{0}{%}`
 3883 `\addtolength{\LWR@minipagewidth}{3em}% room for frames`
 3884 `\setlength{\linewidth}{6in}%`
 3885 `\setlength{\textwidth}{6in}%`
 3886 `\setlength{\textheight}{9in}%`
 3887 `}{0}{%}`
 3888 `\LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}`
 3889 `\addtocounter{LWR@minipagedepth}{1}%`
 3890 `\setlength{\LWR@minipageheight}{\textheight}% default unless specified`
 3891 `\IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{0}{%}`

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.

```
3892 \LWR@stoppars%
```

Create the <div> tag with optional alignment style:

```
3893 \LWR@traceinfo{minipage: creating div class}%
3894 \LWR@htmltag{div class="minipage" style="%
3895 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
3896 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
3897 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
3898 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
3899 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
3900 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
3901 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%
```

Print the width and optional height styles:

```
3902 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
3903 \uselengthunit{PT}%
3904 \ifbool{\LWR@minipagefullwidth}%
3905 {\boolfalse{\LWR@minipagefullwidth}}%
3906 {%
3907 \ifthenelse{\lengthtest{#4}=\linewidth}%
3908 {}%
3909 {width:\rndprintlength{\LWR@minipagewidth} ; }%
3910 }%
3911 \LWR@traceinfo{minipage: about to print the height}%
3912 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
3913 "{}}%
```

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.

```
3914
3915 \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.

```
3916 \setlength{\linewidth}{#4}% the original width
3917 \setlength{\textwidth}{6in}%
3918 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3919 \LWR@origraggedright%
3920 \let\footnote\LWR@origfootnote%
3921 \let\@makefnmark\LWR@makefnmark%
3922 \let\@makefnmark\LWR@makefnmark%
```

Resume paragraph tag handling for the contents of the minipage:

```
3923 \LWR@startpars%
3924 \LWR@traceinfo{minipage: finished starting the minipage}%
3925 }
```

End the environment with L^AT_EX processing and closing tag:

```
3926 {%
3927 \LWR@stoppars%
3928 \LWR@endminipage% The following empty line is required:
3929
3930 \LWR@htmldivclassend{minipage}%
3931 \vspace{1\baselineskip}% required for subcaption
3932 \addtocounter{LWR@minipagedepth}{-1}%
3933 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
3934 \global\booltrue{LWR@minipagethispar}%
3935 }
```

61.4 Parbox, makebox, framebox, fbox, raisebox

`\parbox` [*⟨pos⟩*] [*⟨height⟩*] [*⟨inner-pos⟩*] {*⟨width⟩*} {*⟨text⟩*}

A parbox uses the minipage code:

```
3936 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
3937 {
3938 \LWR@traceinfo{parbox of width #4}%
3939 \begin{minipage}[#1][#2][#3]{#4}
3940 #5
3941 \end{minipage}
3942 }
```

`\makebox` [*⟨width⟩*] [*⟨pos⟩*] {*⟨text⟩*}

Width and position are ignored.

```
3943 \RenewDocumentCommand{\makebox}{o o m}{%
3944 \mbox{#3}
3945 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

Width and position are ignored.

```
3946 \RenewDocumentCommand{\framebox}{o o m}{%
3947 \fbox{#3}
3948 }
```

`\fbox` {*<text>*}

```
3949 \let\LWR@origfbox\fbox
3950 %
3951 \renewcommand*{\fbox}[1]{%
3952 \InlineClass{framebox}{#1}%
3953 }
```

`\raisebox` {*<raiselen>*} [*<height>*] [*<depth>*] {*<text>*}

```
3954 \RenewDocumentCommand{\raisebox}{m o o m}{%
3955 #4%
3956 }

3957 \end{wrapHTML}
```

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: 3958 `\begin{wrapHTML}`

`\emph` {*<text>*}

```
3959 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}
```

`\textmd` {*<text>*}

```
3960 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}
```

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

3961 `\renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}`

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

3962 `\renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}`

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

3963 `\renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}`

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

3964 `\renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}`

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

3965 `\renewcommand{\textup}[1]{\LWR@htmlspan{textup}{#1}}`

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

3966 `\renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}`

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

3967 `\renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}`

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

3968 `\renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}`

`\mdseries`

3969 `\renewcommand*\mdseries{}`

`\bfseries`

3970 `\renewcommand*\bfseries{}`

`\rmfamily`

3971 `\renewcommand*\rmfamily{}`

`\sffamily`

3972 `\renewcommand*\sffamily{}`

`\ttfamily`

3973 `\renewcommand*\ttfamily{}`

`\upshape`

3974 `\renewcommand*\upshape{}`

`\itshape`

3975 `\renewcommand*\itshape{}`

`\scshape`

3976 `\renewcommand*\scshape{}`

`\scshape`

3977 `\renewcommand*\normalfont{}`

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

3978 `\renewcommand{\sp}[1]{\text{^{#1}}}`

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

3979 `\renewcommand{\sb}[1]{\text{_{#1}}}`

`\textsuperscript` $\langle text \rangle$

3980 `\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}`

`\textsubscript` $\langle text \rangle$

3981 `\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.

```
3982 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup` $\{\langle text \rangle\}$ Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
3983 \AtBeginDocument{\let\fup\textsuperscript}
```

`\hfill`

```
3984 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
3985 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
3986 \renewcommand*{\dotfill}{\dots}
```

```
3987 \end{warpHTML}
```

63 Skips, spaces, font sizes

for HTML output: 3988 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
3989 \let\LWR@origcomma\,
3990 \let\LWR@origtilde~
3991 \let\LWR@origenskip\enskip
3992 \let\LWR@origquad\quad
3993 \let\LWR@origqqquad\qqquad
3994 \let\LWR@orighspace\hspace
3995 \let\LWR@origrule\rule
```

Direct-formatting space commands become HTML entities:

```
3996 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
3997 \renewcommand*{~}{\HTMLentity{nbsp}}
3998 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
3999 \let\LWR@orignormalsize\normalsize
4000 \let\LWR@origsmall\small
4001 \let\LWR@origfootnotesize\footnotesize
4002 \let\LWR@origscriptsize\scriptsize
4003 \let\LWR@origtiny\tiny
4004 \let\LWR@origlarge\large
4005 \let\LWR@origLarge\Large
4006 \let\LWR@origLARGE\LARGE
4007 \let\LWR@orighuge\huge
4008 \let\LWR@origHuge\Huge
4009 \renewcommand*{\normalsize}{}
4010 \renewcommand*{\small}{}
4011 \renewcommand*{\footnotesize}{}
4012 \renewcommand*{\scriptsize}{}
4013 \renewcommand*{\tiny}{}
4014 \renewcommand*{\large}{}
4015 \renewcommand*{\Large}{}
4016 \renewcommand*{\LARGE}{}
4017 \renewcommand*{\huge}{}
4018 \renewcommand*{\Huge}{}
4019
4020 \renewcommand*{\onecolumn}{}
4021
4022 \renewcommand{\twocolumn}[1][]{
4023
4024 #1
4025
4026 }
```

`\newline` Uses HTML `
` tag

```
4027 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
4028 \let\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@tabularendofline`.

```
4029 \NewDocumentCommand{\LWR@endofline}{s o}
4030 {%
4031 \newline%
4032 }
```

`\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, `lwrap` 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. `lwrap` 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` 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. `lwrap` 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.

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.

```
4033 \newcommand*{\LWR@minipagestartpars}{%
4034 \ifbool{LWR@minipagethispar}%
4035 {%
4036 \LWR@startpars%
4037 }{}%
4038 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
4039 \newcommand*{\LWR@minipagestoppars}{%
4040 \ifbool{LWR@minipagethispar}%
4041 {%
4042 \LWR@stoppars%
4043 }{}%
4044 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
4045 \renewcommand*{\quad}{%
4046 \LWR@minipagestoppars%
4047 \HTMLUnicode{2001}%
```

```
4048 \LWR@minipagestartpars%
4049 }
```

`\qqad` Handles special minipage & horizontal space interactions.

```
4050 \renewcommand*{\qqad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
4051 \renewcommand*{\enskip}{%
4052 \LWR@minipagestoppars%
4053 \HTMLUnicode{2000}%
4054 \LWR@minipagestartpars%
4055 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```
Len \WR@tempheight
Len \WR@tempraise
4056 \newlength{\LWR@tempwidth}
4057 \newlength{\LWR@tempheight}
4058 \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 `\qqad`.

```
4059 \NewDocumentCommand{\LWR@hspace}{s m}{%
4060 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\qqad`:

```
4061 \ifnum\gluestretchorder\LWR@tempwidth>0%
4062 \setlength{\LWR@tempwidth}{2em}%
4063 \fi%
```

Only if the width is not zero:

```
4064 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4065 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
4066 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
4067 {%
4068 \HTMLUnicode{2009}% thin breakable space
4069 }%
```

Print the span with the converted width. Not rounded.

```
4070 {%
4071 \uselengthunit{PT}%
4072 \LWR@htmltagc{%
4073 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
4074 }%
4075 \LWR@htmltagc{/span}%
4076 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4077 \LWR@minipagestartpars%
4078 }%
4079 }
```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
4080 \let\hspace\LWR@hspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML br tag.

```
4081 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
4082 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
4083 \renewcommand*{\pagebreak}[1] [] {
4084
4085 }
```

`\nopagebreak` [$\langle num \rangle$]

```
4086 \renewcommand*{\nopagebreak}[1] [] {}
```

`\enlargethispage` * $\{\langle len \rangle\}$

```
4087 \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.

```
4088 \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:

```
4089 \setlength{\LWR@tempwidth}{#2}%

```

If it's zero-width then skip the entire rule:

```
4090 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
4091 {}% zero- width
4092 {}% non-zero width

```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
4093 \ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%
4094 \lengthtest{\LWR@tempwidth<1pt}}{
4095 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```
4096 \setlength{\LWR@tempheight}{#3}%
4097 \ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%
4098 \lengthtest{\LWR@tempheight<1pt}}{
4099 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
4100 \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.

```
4101 \uselengthunit{PT}%

```

```

4102 \LWR@htmltagc{%
4103 span
4104 style=" %

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

4105 background:\LWR@currenttextcolor; %

```

The width and height are printed, converted to PT:

```

4106 width:\printlength{\LWR@tempwidth}; %
4107 height:\printlength{\LWR@tempheight}; %

```

The raise height is converted to a CSS transform. The `*2` raise multiplier is to approximately match HTML output's X height. Conversion to a \LaTeX length allows a typical \LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of \LaTeX ex height, then print that to HTML with an ex unit.

```

4108 \IfValueTF{#1}%
4109 {%
4110 \setlength{\LWR@tempraise}{Opt-#1}%
4111 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4112 \LWR@orignewline%
4113 -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4114 \LWR@orignewline%
4115 -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
4116 \LWR@orignewline%
4117 transform: translate(0pt,\printlength{\LWR@tempraise}); %
4118 \LWR@orignewline%
4119 }{}%

```

Display inline-block to place the span inline with the text:

```

4120 display:inline-block; "%
4121 }%
4122 \LWR@htmltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

4123 \LWR@minipagestartpars%
4124 }% non-zero width
4125 }

```

`\rule` [$\langle raise \rangle$] [$\langle width \rangle$] [$\langle height \rangle$]

Handles special minipage & horizontal space interactions.

```
4126 \let\rule\LWR@rule
```

```
4127 \end{warpHTML}
```

64 \phantomsection

for HTML output: 4128 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
4129 \newcommand*\phantomsection{\section*{}}
```

```
4130 \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@LwrapStart`.

For CSS conversions, see:

<http://edward.oconnor.cx/2007/08/tex-poshlet>

<http://nitens.org/taraborelli/texlogo>

65.1 HTML logos

for HTML output: 4131 \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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{ConT}_{\mathrm{E}}\mathrm{X}$ t, etc.

```

4132 \newcommand*{\LWR@TeX}
4133 {\InlineClass{latexlogofont}%
4134 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}

```

`\LaTeX` \LaTeX , $\text{\LaTeX} 2_{\epsilon}$

`\LaTeXe`

```

4135 \newcommand*{\LWR@LaTeX}
4136 {\InlineClass{latexlogofont}%
4137 {\InlineClass{latexlogo}%
4138 {L\textsuperscript{a}T\textsubscript{e}X}}}
4139
4140 \renewcommand*{\LaTeXe}
4141 {\LaTeX\InlineClass{latexlogofont}%
4142 {\,2\textsubscript{\textit{HTMLunicode}{3B5}}}}}

```

`\LuaTeX` \LuaTeX , \Lua\LaTeX

`\LuaLaTeX`

```

4143 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
4144 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}

```

`\XeTeX` \XeTeX , \Xe\LaTeX

`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in \XeTeX and spaces \TeX appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in \Xe\LaTeX and spaces \LaTeX appropriately.

```

4145 \newcommand*{\Xe}
4146 {X\textsubscript{\HTMLunicode{18e}}}
4147 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
4148 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt` \ConTeXt

```

4149 \newcommand*{\LWR@ConTeXt}
4150 {\InlineClass{latexlogofont}{Con}\TeX{}}%
4151 \InlineClass{latexlogofont}{t}}

```

`\BibTeX` \BibTeX , *MakeIndex*

`\MakeIndex`

```

4152 \providecommand*{\BibTeX}
4153 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
4154
4155 \newcommand*{\MakeIndex}
4156 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS` \mathcal{AMS}

`amslogo` is a CSS class used for the \mathcal{AMS} logo.

```
4157 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
4158 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX` $\text{MiK}\text{T}_{\text{E}}\text{X}$

```
4159 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` LyX

`lyxlogo` is a CSS class used for the LyX logo.

```
4160 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}

4161 \end{warpHTML}
```

65.2 Print logos

for PRINT output:

```
4162 \begin{warpprint}
4163 \newcommand*{\XeTeXrevE}
4164   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
4165 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
4166 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
4167 \providecommand*{\AmS}{%
4168   \leavevmode\hbox{$\mathcal{A}\kern-.2em\lower.376ex%
4169   \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}$}}
4170 \newcommand*{\LyX}{\textsf{LyX}}
4171 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
4172 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
4173 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
4174 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
4175 \providecommand*{\ConTeXt}{\mbox{Con\TeX{t}}t}
4176 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
4177 \end{warpprint}
```

66 `\AtBeginDocument`, `\AtEndDocument`

for HTML output:

```
4178 \begin{warpHTML}
```

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the
`\LWR@LwarpEnd` document.

```
4179 \AfterEndPreamble{\LWR@LwarpStart}
```

```
4180 \AtEndDocument{\LWR@LwarpEnd}
```

```
4181 \end{warpHTML}
```

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:

```
4182 \providecommand*\HomeHTMLFileName{\BaseJobname}
4183 \providecommand*\HTMLFileName{}
4184 \providecommand*\UseLatexmk{false}
```

for PRINT output:

```
4185 \begin{warpprint}
4186 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
4187 \immediate\openout{LWR@file}=lwarpmk.conf
4188 \ifwindows
4189 \immediate\write{LWR@file}{opsystem = "Windows"}
4190 \else
4191 \immediate\write{LWR@file}{opsystem = "Unix"}
4192 \fi
4193 \ifPDFTeX
4194 \immediate\write{LWR@file}{latexname = "pdflatex"}
4195 \fi
4196 \ifXeTeX
4197 \immediate\write{LWR@file}{latexname = "xelatex"}
4198 \fi
4199 \ifLuaTeX
4200 \immediate\write{LWR@file}{latexname = "lualatex"}
4201 \fi
4202 \immediate\write{LWR@file}{sourcename = "\jobname"}
4203 \immediate\write{LWR@file}{%
4204 homehtmlfilename = "\HomeHTMLFileName"%
4205 }
4206 \immediate\write{LWR@file}{htmlfilename = "\HTMLFileName"}
4207 \immediate\write{LWR@file}{uselatexmk = "\UseLatexmk"}
4208 \immediate\closeout{LWR@file}
4209 \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 width:20%; float:right;
311 text-align:left;
312 margin: 1ex 0.5em 1ex 1em ;
313 padding: 1ex 0.5em 1ex 0.5em ;
314 font-size: 85% ;
315 border-top: 1px solid silver ;
316 border-bottom: 1px solid silver ;
317 overflow-x: auto;
318 }
319
320 .marginpar br { margin-bottom: 2ex ; }
321
322 div.marginblock {
323 width:50%; float:right;
324 text-align:left;
325 margin: 1ex 0.5em 1ex 1em ;
326 padding: 1ex 0.5em 1ex 0.5em ;
327 overflow-x: auto;
328 }
329
330 div.marginblock div.minipage p { font-size: 85%}
331
332 div.marginblock br { margin-bottom: 2ex ; }
333
```

```
334
335 section.textbody div.footnotes{
336     margin: 3ex 0em 0ex 0em ;
337     border-bottom: 2px solid silver ;
338 }
339
340 .footnoteheader {
341     border-top: 2px solid silver ;
342     margin-top: 3ex ;
343     padding-top: 1ex ;
344     font-weight: bold ;
345 }
346
347 .mpfootnotes {
348     text-align: left ;
349     font-size: .85em ;
350     margin: 3ex 0em 3ex 0em ;
351     border-top: 1px solid silver ;
352     border-bottom: 1px solid silver ;
353 /*     padding-top: 1ex ; */
354 }
355
356 /* Remove footnote top border in the title page. */
357 div.titlepage div.mpfootnotes {
358     border-top: none ;
359 }
360
361
362
363 ol {
364     margin: 1ex 1em 1ex 0em;
365     line-height: 1.2;
366 }
367
368 ul, body dir, body menu {
369     margin: 1ex 1em 1ex 0em;
370     line-height: 1.2;
371 }
372
373 li { margin: 0ex 0em 1ex 0em; }
374
375 html {
376     margin: 0;
377     padding: 0;
378 }
379
380 .programlisting {
381     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
382                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
383                 "Courier New", monospace;
```



```
384 margin: 1ex 0ex 1ex 0ex ;
385 padding: .5ex 0pt .5ex 0pt ;
386 overflow-x: auto;
387 }
388
389 section.textbody>pre.programlisting {
390 border-top: 1px solid silver ;
391 border-bottom: 1px solid silver ;
392 }
393
394
395 .inlineprogramlisting {
396 font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
397             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
398             "Courier New", monospace;
399 overflow-x: auto;
400 }
401
402
403 div.abstract {
404 margin: 2em 5% 2em 5% ;
405 padding: 1ex 1em 1ex 1em ;
406 /* font-weight: bold ; */
407 font-size: 90% ;
408 }
409
410 div.abstract dl {line-height:1.5;}
411 div.abstract dt {color:#304070;}
412
413 div.abstracttitle{
414 font-family: "URW Classico", Optima, "Linux Biolinum O",
415             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
416             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
417 font-weight:bold;
418 font-size:1.25em;
419 text-align: center ;
420 }
421
422 span.abstracrunintitle{
423 font-family: "URW Classico", Optima, "Linux Biolinum O",
424             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
425             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
426 font-weight:bold;
427 }
428
429
430 .verbatim {
431 overflow-x: auto ;
432 }
433
```

```
434 .alltt {
435     overflow-x: auto ;
436 }
437
438
439 .bverbatim {
440     margin: 1ex Opt 1ex Opt ;
441     padding: .5ex Opt .5ex Opt ;
442     overflow-x: auto ;
443 }
444
445 .lverbatim {
446     margin: 1ex Opt 1ex Opt ;
447     padding: .5ex Opt .5ex Opt ;
448     overflow-x: auto ;
449 }
450
451 .fancyvrb {
452     font-size:.85em ;
453     margin: 3ex Opt 3ex Opt
454 }
455
456 .fancyvrblabel {
457     font-weight:bold;
458     text-align: center ;
459 }
460
461
462 .verse {
463     font-family: "Linux Libertine Mono O", "Lucida Console",
464                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
465                 "Liberation Mono", "FreeMono", "Andale Mono",
466                 "Nimbus Mono L", "Courier New", monospace;
467     margin-left: 1em ;
468 }
469
470
471 div.singlespace { line-height: 1.2 ; }
472 div.onehalfspace { line-height: 1.5 ; }
473 div.doublespace { line-height: 2 ; }
474
475
476
477
478
479 /* Minipage environments, vertically aligned to top, center, bottom: */
480 .minipage {
481     /* display: inline-block ; */
482     /* Mini pages which follow each other will be tiled. */
483     margin: .25em .25em .25em .25em;
```

```
484     padding: .25em .25em .25em .25em;
485     display: inline-flex;
486     flex-direction: column ;
487     overflow: auto;
488 }
489
490 /* Paragraphs in the flexbox did not collapse their margins. */
491 /* Have not yet researched this. */
492 .minipage p {margin: .75ex 0em .75ex 0em ;}
493
494
495
496 .framebox {
497     margin: 0ex ;
498     padding: 0ex ;
499     border: 1px solid black;
500     border-radius: 0px ;
501     padding: .3ex .2em 0ex .2em ;
502     margin: .1ex ;
503     display: inline-block ;
504 }
505
506
507 .mdframed {
508 /*     padding: 0ex ; */
509 /*     border: 1px solid black; */
510 /*     border-radius: 0px ; */
511     padding: 0ex ;
512     margin: 3ex 5% 3ex 5% ;
513 /*     display: inline-block ; */
514 }
515
516 .mdframed p { padding: 0ex .5em 0ex .5em ; }
517
518 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
519
520 .mdframedtitle {
521     padding: .5em ;
522     display: block ;
523     font-size: 130%
524 }
525
526 .mdframedsubtitle {
527     padding: 0ex .5em 0ex .5em ;
528     display: block ;
529     font-size: 115% ;
530 }
531
532 .mdframedsubsubtitle {
533     padding: 0ex .5em 0ex .5em ;
```

```
534     display: block ;
535 }
536
537 .mdtheorem {
538     padding: 0ex .5em 0ex .5em ;
539     margin: 3ex 5% 3ex 5% ;
540 /*     display: inline-block ; */
541 }
542
543
544 /* framed package */
545 .framed {
546     margin: 3ex 0em 3ex 0em ;
547     border: 1px solid black;
548     border-radius: 0px ;
549     padding: .3ex 1em 0ex 1em ;
550     display: block ;
551 }
552
553 .snugframed {
554     margin: 3ex 0em 3ex 0em ;
555     border: 1px solid black;
556     border-radius: 0px ;
557     display: block ;
558 }
559
560 .framedleftbar {
561     margin: 3ex 0em 3ex 0em ;
562     border-left: 3pt solid black;
563     border-radius: 0px ;
564     padding: .3ex .2em .3ex 1em ;
565     display: block ;
566 }
567
568 .framedtitle {
569 margin: 0em ;
570 padding: 0em ;
571     font-size: 130%
572 }
573
574 .framedtitle p { padding: .3em }
575
576
577
578 dl {
579     margin: 1ex 2em 1ex 0em;
580     line-height: 1.3;
581 }
582
583 dl dt {
```

```
584 margin-top: 1ex;
585 font-weight: bold;
586 }
587
588 dl dd p { margin-top: 0em; }
589
590
591 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
592     font-family: "URW Classico", Optima, "Linux Biolinum O",
593         "DejaVu Sans", "Bitstream Vera Sans",
594         Geneva, Verdana, sans-serif ;
595     margin-bottom: 4ex ;
596 }
597
598 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
599     line-height: 1.2 ;
600     margin-top: .5ex ;
601     margin-bottom: .5ex;
602     font-size: .9em ;
603 }
604
605
606
607 img, img.hyperimage, img.borderimage {
608     max-width: 600px;
609     border: 1px solid silver;
610     box-shadow: 3px 3px 3px #808080 ;
611     padding: .5% ;
612     margin: .5% ;
613     background: none ;
614 }
615
616 img.inlineimage{
617     padding: 0px ;
618     box-shadow: none ;
619     border: none ;
620     background: none ;
621     margin: 0px ;
622     display: inline-block ;
623     border-radius: 0px ;
624 }
625
626 img.logoimage{
627     max-width: 300px ;
628     box-shadow: 3px 3px 3px #808080 ;
629     border: 1px solid black ;
630     background:none ;
631     padding:0 ;
632     margin:.5ex ;
633     border-radius: 10px ;
```

```
634 }
635
636
637 .section {
638 /*
639     To have each section float relative to each other:
640 */
641 /*
642     display: block ;
643     float: left ;
644     position: relative ;
645     background: white ;
646     border: 1px solid silver ;
647     padding: .5em ;
648 */
649     margin: 0ex .5em 0ex .5em ;
650     padding: 0 ;
651 }
652
653
654 figure {
655     margin: 3ex auto 3ex auto ;
656     padding: 1ex 1em 1ex 1em ;
657     overflow-x: auto ;
658 }
659
660
661 /* To automatically center images in figures: */
662 /*
663 figure img.inlineimage {
664     margin: 0ex auto 0ex auto ;
665     display: block ;
666 }
667 */
668
669 /* To automatically center minipages in figures: */
670 /*
671 figure div.minipage, figure div.minipage div.minipage {
672     margin: 1ex auto 1ex auto ;
673     display: block ;
674 }
675 */
676
677 figure div.minipage p { font-size: 85% ; }
678
679 figure.subfigure, figure.subtable {
680     display: inline-block ; margin: 3ex 1em 3ex 1em ;
681 }
682
683 figcaption .minipage { margin:0 ; padding: 0 }
```

```

684
685 div.floatrow { text-align: center; }
686
687 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
688
689 div.floatfoot { font-size: .85em ;
690     border-top: 1px solid silver ; line-height: 1.2 ; }
691
692 figcaption , .lstlistingtitle {
693     font-size: .85em ;
694     text-align: center ;
695     font-weight: bold ;
696 margin-top: 1ex ;
697 margin-bottom: 1ex ;
698 }
699
700 figure.subfigure figcaption, figure.subtable figcaption {
701     border-bottom: none ; background: none ;
702 }
703
704 div.nonfloatcaption {
705     margin: 1ex auto 1ex auto ;
706     font-size: .85em ;
707     text-align: center ;
708     font-weight: bold ;
709 }
710
711 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
712 figure div.floatrow div.minipage figcaption {
713 border: none ;
714 background: none ;
715 }
716
717
718 table {
719     margin: 1ex auto 1ex auto ;
720     border-collapse: collapse ;
721     border-spacing: 0px ;
722     line-height: 1.3 ;
723 }
724
725 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
726     margin-bottom: 0ex ; } /* for \hline */
727
728 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
729     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
730
731 td {padding: 1ex .5em 1ex .5em ;}
732
733 table td.tdl { text-align: left ; vertical-align: middle ; }

```

```

734 table td.tdc { text-align: center ; vertical-align: middle ; }
735 table td.tdr { text-align: right ; vertical-align: middle ; }
736 table td.tdp { text-align: left ; vertical-align: bottom ; }
737 table td.tdm { text-align: left ; vertical-align: middle ; }
738 table td.tdb { text-align: left ; vertical-align: top ; }
739 table td.tdP { text-align: center ; vertical-align: bottom ; }
740 table td.tdM { text-align: center ; vertical-align: middle ; }
741 table td.tdB { text-align: center ; vertical-align: top ; }
742 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
743     vertical-align: middle ; } /* for cmidrule */
744 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
745     vertical-align: middle ; }
746 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
747     vertical-align: middle ; }
748 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
749     vertical-align: bottom ; }
750 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
751     vertical-align: middle ; }
752 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
753     vertical-align: top ; }
754 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
755     vertical-align: bottom ; }
756 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
757     vertical-align: middle ; }
758 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
759     vertical-align: top ; }
760
761 /* Margins of paragraphs inside table cells: */
762 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
763     padding-bottom: 1ex ; margin: 0ex ; }
764 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
765     padding-bottom: 1ex ; margin: 0ex ; }
766 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
767     padding-bottom: 1ex ; margin: 0ex ; }
768
769 td.tdp , td.tdprule , td.tdP , td.tdPrule
770     { padding: 0ex .5em 0ex .5em ; }
771 td.tdm , td.tdmrule , td.tdM , td.tdMrule
772     { padding: 0ex .5em 0ex .5em ; }
773 td.tdb , td.tdbrule , td.tdB , td.tdBrule
774     { padding: 0ex .5em 0ex .5em ; }
775
776
777 /* table notes: */
778 .tnotes {
779     margin: 0ex 5% 1ex 5% ;
780     padding: 0.5ex 1em 0.5ex 1em;
781     font-size:.85em;
782     text-align: left ;
783 }

```



```
784
785 .tnotes dl dt p {margin-bottom:0px;}
786
787 .tnoteitemheader {margin-right: 1em;}
788
789
790
791 /* center, flushleft, flushright environments */
792 div.center{text-align:center;}
793 div.center table {margin-left:auto;margin-right:auto;}
794 div.flushleft{text-align:left;}
795 div.flushleft table {margin-left:0em ; margin-right:auto;}
796 div.flushright{text-align:right;}
797 div.flushright table {margin-left:auto ; margin-right: 0em ;}
798
799
800
801
802 /* program listing callouts: */
803 span.callout {
804     font-family: "DejaVu Sans", "Bitstream Vera Sans",
805     Geneva, Verdana, sans-serif ;
806     border-radius: .5em;
807     background-color:black;
808     color:white;
809     padding:0px .25em 0px .25em;
810 margin: 0 ;
811     font-weight: bold;
812     font-size:.72em ;
813 }
814
815 div.programlisting pre.verbatim span.callout{
816 font-size: .85em ;
817 }
818
819
820
821
822
823 div.published
824 {
825     text-align: center ;
826     font-variant: normal ;
827     font-style: italic ;
828     font-size: 1em ;
829     margin: 3ex 0em 3ex 0em ;
830 }
831
832 div.subtitle
833 {
```

```
834     text-align: center ;
835     font-variant: normal ;
836     font-style: italic ;
837     font-size: 1.25em ;
838     margin: 3ex 0em 3ex 0em ;
839 }
840
841 div.subtitle p { margin: 1ex ; }
842
843 div.author
844 {
845     font-variant: normal ;
846     font-style: normal ;
847     font-size: 1em ;
848     margin: 3ex 0em 3ex 0em ;
849 }
850
851 div.author table {
852     margin: 3ex auto 0ex auto ;
853     background: none ;
854 }
855
856 div.author table tbody tr td { padding: .25ex ; }
857
858 span.affiliation {font-size: .85em ; font-variant: small-caps; }
859
860 div.titledate {
861     text-align: center ;
862     font-size: .85em ;
863     font-style: italic;
864     margin: 6ex 0em 6ex 0em ;
865 }
866
867
868 nav.topnavigation{
869     text-align: left ;
870     padding: 0.5ex 1em 0.5ex 1em ;
871 /*     margin: 2ex 0em 3ex 0em ; */
872     margin: 0 ;
873     border-bottom: 1px solid silver ;
874     border-top: 1px solid silver ;
875     clear:right ;
876 }
877
878 nav.botnavigation{
879     text-align: left ;
880     padding: 0.5ex 1em 0.5ex 1em ;
881 /*     margin: 3ex 0em 2ex 0em ; */
882     margin: 0 ;
883     border-top: 1px solid silver ;
```

```
884     border-bottom: 1px solid silver ;
885     clear:right ;
886 }
887
888
889 header{
890     line-height: 1.2 ;
891     font-size: 1em ;
892 /*     border-bottom: 2px solid silver ; */
893     margin: 0px ;
894     padding: 0ex 1em 0ex 1em ;
895     text-align:center ;
896 }
897
898 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
899
900
901 footer{
902     font-size: .85em ;
903     line-height: 1.2 ;
904     margin-top: 1ex ;
905     border-top: 2px solid silver ;
906     padding: 2ex 1em 2ex 1em ;
907     clear:right ;
908     text-align:left ;
909 }
910
911
912 a.linkhome { font-weight:bold ; font-size: 1em ;}
913
914
915 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
916
917 img.lateximage{
918     padding: 0px 0px 0px 0px ;
919     box-shadow: none ;
920     border: none ;
921     background: none ;
922     margin: 0px 0px -.15ex 0px ;
923     /* pdfcrop leaves a slight margin, adjust to baseline */
924     max-width: 100% ;
925     border-radius: 0ex ;
926     border: none ;
927 }
928
929
930
931 nav.sidetoc {
932     font-family: "DejaVu Serif", "Bitstream Vera Serif",
933         "Lucida Bright", Georgia, serif;
```

```
934     float:right ;
935     width: 20%;
936     border-left: 1px solid silver;
937     border-top: 1px solid silver;
938     border-bottom: 1px solid silver;
939 /*     border-top: 2px solid #808080 ; */
940     background: #FAF7F4 ;
941     padding: 2ex 0em 2ex 1em ;
942     margin: 0ex 0em 2ex 1em ;
943     font-size:.9em ;
944     border-radius: 20px 0px 0px 20px ;
945 }
946
947 div.sidetoccontents {
948 /*     border-top: 1px solid silver ; */
949     overflow-y: auto ;
950     width: 100% ;
951     text-align: left ;
952 }
953
954 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
955     text-indent: 0 ; }
956 nav.sidetoc p a {color:black ; font-size: .7em ;}
957 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
958     border-bottom: 1px solid silver ; }
959 nav.sidetoc a:hover {text-decoration: underline ; }
960
961
962
963 section.textbody { margin: 0ex 1em 0ex 1em ;}
964
965
966 div.multicolsheading { -webkit-column-span: all;
967     -moz-column-span: all; column-span: all; }
968 div.multicols { -webkit-columns: 3 380px ;
969     -moz-columns: 3 380px ; columns: 3 380px ; }
970 div.multicols p {margin-top: 0ex}
971
972
973
974 /* Used to support algorithmicx: */
975 span.floatright { float: right ; }
976
977
978
979
980 /* Native LaTeX theorems: */
981
982 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
983 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
```

```
984
985
986 /* theorem, amsthm, and ntheorem packages */
987
988 span.theoremheader,
989 span.theoremheaderplain,
990 span.theoremheaderdefinition,
991 span.theoremheaderbreak,
992 span.theoremheadermarginbreak,
993 span.theoremheaderchangebreak,
994 span.theoremheaderchange,
995 span.theoremheadermargin
996 {
997 font-style:normal ; font-weight: bold ; margin-right: 1em ;
998 }
999
1000 span.amsthmnameplain,
1001 span.amsthmnamedefinition,
1002 span.amsthmnumberplain,
1003 span.amsthmnumberdefinition
1004 {
1005 font-style:normal ; font-weight: bold ;
1006 }
1007
1008
1009 span.amsthmnameremark,
1010 span.amsthmnumberremark
1011 {font-style:italic ; font-weight: normal ; }
1012
1013
1014 span.amsthmnoteplain,
1015 span.amsthmnotedefinition
1016 {font-style:normal ;}
1017
1018
1019 span.theoremheaderremark,
1020 span.theoremheaderproof,
1021 span.amsthmproofname
1022 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1023
1024 span.theoremheadersc
1025 {
1026 font-style:normal ;
1027 font-variant: small-caps ;
1028 font-weight: normal ;
1029 margin-right: 1em ;
1030 }
1031
1032 .theoremendmark {float:right}
1033
```

```
1034 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1035 div.theorembodybreak, div.theorembodynonumberbreak,
1036 div.theorembodymarginbreak,
1037 div.theorembodychangebreak,
1038 div.theorembodychange,
1039 div.theorembodymargin
1040 {
1041 font-style:italic;
1042 margin-top: 3ex ; margin-bottom: 3ex ;
1043 }
1044
1045 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1046 div.theorembodyplainupright, nonumberplainuprightsc,
1047 div.amsthmbodydefinition, div.amsthmbodyremark,
1048 div.amsthmproof
1049 {
1050 font-style: normal ;
1051 margin-top: 3ex ; margin-bottom: 3ex ;
1052 }
1053
1054 span.amsthmnoteremark {}
1055
1056
1057
1058 /*
1059 For CSS LaTeX and related logos:
1060 Based on:
1061 http://edward.oconnor.cx/2007/08/tex-poshlet
1062 http://nitens.org/taraborelli/texlogo
1063 */
1064
1065 .latexlogofont {
1066     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1067         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1068     font-variant: normal ;
1069 }
1070
1071 .latexlogo {
1072     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1073         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1074     letter-spacing: .03em ;
1075     font-size: 1.1em;
1076 }
1077
1078 .latexlogo sup {
1079     text-transform: uppercase;
1080     letter-spacing: .03em ;
1081     font-size: 0.85em;
1082     vertical-align: 0.15em;
1083     margin-left: -0.36em;
```

```
1084 margin-right: -0.15em;
1085 }
1086
1087 .latexlogo sub {
1088   text-transform: uppercase;
1089   vertical-align: -0.5ex;
1090   margin-left: -0.1667em;
1091   margin-right: -0.125em;
1092   font-size: 1em;
1093 }
1094
1095 .xetexlogo {
1096   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1097               "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1098   letter-spacing: .03em ;
1099   font-size: 1.1em;
1100 }
1101
1102 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1103 .xetexlogo sub {
1104   text-transform: uppercase;
1105   vertical-align: -0.5ex;
1106   margin-left: -0.0667em;
1107   margin-right: -0.2em;
1108   font-size: 1em;
1109   letter-spacing: .03em ;
1110 }
1111
1112 /* A large gap between Xe and LaTeX v.s. TeX: */
1113 .xelatexlogo sub {
1114   text-transform: uppercase;
1115   vertical-align: -0.5ex;
1116   margin-left: -0.0667em;
1117   margin-right: -.05em;
1118   font-size: 1em;
1119   letter-spacing: .03em ;
1120 }
1121
1122 .amslogo {
1123   font-family: "TeXGyreChorus", "URW Chancery L",
1124               "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
1125               "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1126               "Hoefler Text", Times, "Times New Roman", serif;
1127   font-style: italic;
1128 }
1129
1130 .lyxlogo {
1131   font-family: "URW Classico", Optima, "Linux Biolinum O",
1132               "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1133   Verdana, sans-serif ;
```

```
1134 }
1135
1136
1137
1138
1139 /* Only display top and bottom navigation if a small screen: */
1140 /* Hide the sidetoc if a small screen: */
1141 nav.topnavigation { display:none; }
1142 nav.botnavigation { display:none; }
1143
1144 @media screen and (max-width: 45em) {
1145 /*     nav.sidetoc {display:none;} */
1146     nav.sidetoc {
1147         float: none ;
1148         width: 100% ;
1149         margin: 5ex 0px 5ex 0px ;
1150         padding: 0 ;
1151         border-radius: 0 ;
1152         border-bottom: 1px solid black ;
1153         border-top: 1px solid black ;
1154         box-shadow: none ;
1155     }
1156 /*     nav.topnavigation { display:block } */
1157     nav.botnavigation { display:block }
1158 }
1159
1160 @media print {
1161     body {
1162         font-family: "Linux Libertine O",
1163         "DejaVu Serif", "Bitstream Vera Serif",
1164         "Liberation Serif", "Nimbus Roman No 9 L",
1165         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1166     }
1167     nav.sidetoc { display:none; }
1168     nav.topnavigation { display: none; }
1169     nav.botnavigation { display: none; }
1170 }
1171
1172 @media handheld {
1173     nav.sidetoc { display:none; }
1174     nav.topnavigation { display:block }
1175     nav.botnavigation { display:block }
1176 }
1177
1178 @media projection {
1179     nav.sidetoc { display:none; }
1180     nav.topnavigation { display:block }
1181     nav.botnavigation { display:block }
1182 }
1183 \end{VerbatimOut}
```



```
1184 % \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.

```
1185 \begin{VerbatimOut}{lwarp_sagebrush.css}
1186 @import url("lwarp.css") ;
1187
1188
1189 A:link {color:#105030 ; text-decoration: none ; }
1190 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1191 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1192 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1193
1194
1195
1196 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1197 {
1198     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1199         "Linux Libertine 0", "Liberation Serif",
1200         "Nimbus Roman No 9 L", "FreeSerif",
1201         "Hoefler Text", Times, "Times New Roman", serif;
1202     font-variant: small-caps ;
1203     font-weight: normal ;
1204     color: #304070 ;
1205     text-shadow: 2px 2px 3px #808080;
1206 }
1207
1208 h1 { /* title of the entire website, used on each page */
1209     font-variant: small-caps ;
1210     color: #304070 ;
1211     text-shadow: 2px 2px 3px #808080;
1212     background-color: #F7F7F0 ;
1213     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1214 }
1215
1216 h1 {
1217     border-bottom: 1px solid #304070;
1218     border-top: 2px solid #304070;
1219 }
1220
1221 h2 {
1222     border-bottom: 1px solid #304070;
```

```
1223 border-top: 2px solid #304070;
1224 background-color: #F7F7F0 ;
1225 background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1226 }
1227
1228
1229
1230 div.abstract {
1231     background: #f5f5eb ;
1232     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1233
1234     border: 1px solid silver;
1235     border-radius: 1em ;
1236 }
1237
1238 div.abstract dl {line-height:1.5;}
1239 div.abstract dt {color:#304070;}
1240
1241 div.abstracttitle{
1242     font-family: "URW Classico", Optima, "Linux Biolinum O",
1243         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1244         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1245     font-weight:bold;
1246     font-variant: small-caps ;
1247     font-size:1.5em;
1248     border-bottom: 1px solid silver ;
1249     color: #304070 ;
1250     text-align: center ;
1251     text-shadow: 1px 1px 2px #808080;
1252 }
1253
1254 span.abstracrunintitle{
1255     font-family: "URW Classico", Optima, "Linux Biolinum O",
1256         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1257         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1258     font-weight:bold;
1259 }
1260
1261
1262 div.epigraph {
1263     background: #f5f5eb ;
1264     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1265
1266     border: 1px solid silver ;
1267     border-radius: 1ex ;
1268     box-shadow: 3px 3px 3px #808080 ;
1269 }
1270
1271
1272 .example {
```

```
1273     background-color: #f5f5eb ;
1274     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1275
1276 }
1277
1278 div.exampletitle{
1279     font-family: "URW Classico", Optima, "Linux Biolinum O",
1280         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1281         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1282     font-weight:bold;
1283     font-variant: small-caps ;
1284     border-bottom: 1px solid silver ;
1285     color: #304070 ;
1286     text-align: center ;
1287     text-shadow: 1px 1px 2px #808080;
1288 }
1289
1290
1291 .sidebar {
1292     background-color: #f5f5eb ;
1293     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1294
1295 }
1296
1297 div.sidebartitle{
1298     font-family: "URW Classico", Optima, "Linux Biolinum O",
1299         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1300         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1301     font-weight:bold;
1302     font-variant: small-caps ;
1303     border-bottom: 1px solid silver ;
1304     color: #304070 ;
1305     text-align: center ;
1306     text-shadow: 1px 1px 2px #808080;
1307 }
1308
1309
1310 .fancyvrblabel {
1311     font-family: "URW Classico", Optima, "Linux Biolinum O",
1312         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1313         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1314     font-weight:bold;
1315     font-variant: small-caps ;
1316     font-size: 1.5em ;
1317     color: #304070 ;
1318     text-align: center ;
1319     text-shadow: 1px 1px 2px #808080;
1320 }
1321
1322
```

```
1323
1324 .minipage {
1325     background-color: #eeeeee7 ;
1326     border: 1px solid silver ;
1327     border-radius: 1ex ;
1328 }
1329
1330 .framed .minipage , .framedleftbar .minipage {
1331     border: none ;
1332     background: none ;
1333     padding: 0ex ;
1334     margin: 0ex ;
1335 }
1336
1337 figure.figure .minipage, figcaption .minipage { border: none; }
1338
1339 div.marginblock div.minipage { border: none; }
1340
1341 figure , div.marginblock {
1342     background-color: #eeeeee7 ;
1343     border: 1px solid silver ;
1344     border-radius: 1ex ;
1345     box-shadow: 3px 3px 3px #808080 ;
1346 }
1347
1348 figure figure {
1349     border: 1px solid silver ;
1350     margin: 0em ;
1351     box-shadow: none ;
1352 }
1353
1354 /*
1355 figcaption {
1356     border-top: 1px solid silver ;
1357     border-bottom: 1px solid silver ;
1358     background-color: #e8e8e8 ;
1359 }
1360 */
1361
1362
1363 div.table {
1364     box-shadow: 3px 3px 3px #808080 ;
1365 }
1366
1367 /*
1368 .tnotes {
1369     background: #e8e8e8;
1370     border: 1px solid silver;
1371 }
1372 */
```

```

1373
1374
1375 nav.topnavigation{
1376     background-color: #b0b8b0 ;
1377     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1378 }
1379
1380 nav.botnavigation{
1381     background-color: #b0b8b0 ;
1382     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1383 }
1384
1385
1386
1387 header{
1388     background-color: #F7F7F0 ;
1389     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1390 }
1391
1392 footer{
1393     background-color: #F7F7F0 ;
1394     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1395 }
1396
1397
1398
1399 nav.sidetoc {
1400     background-color: #F7F7F0 ;
1401     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1402     box-shadow: 3px 3px 3px #808080 ;
1403     border-radius: 0px 0px 0px 20px ;
1404 }
1405
1406 div.sidetoctitle {color: #304070 ; }
1407 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1408
1409
1410 @media screen and (max-width: 45em) {
1411     nav.sidetoc { border-radius: 0 ; }
1412 }
1413
1414
1415 \end{VerbatimOut}
1416 % \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.

```
1417 \begin{VerbatimOut}{lwarp_formal.css}
1418 @import url("lwarp.css") ;
1419
1420
1421
1422 A:link {color:#802020 ; text-decoration:none; }
1423 A:visited {color:#802020 ; text-shadow:none ;}
1424 A:hover {color:#400000 ; text-shadow:none ;}
1425 A:active {color:#C00000 ; text-shadow:none ;}
1426
1427
1428 body {
1429     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1430         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1431         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1432         "Times New Roman", serif;
1433     background: #fffcf5;
1434 }
1435
1436 span.textrm {
1437     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1438         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1439         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1440         "Times New Roman", serif;
1441 }
1442
1443 span.textsf {
1444     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1445         Geneva, Verdana, sans-serif ;
1446 }
1447
1448
1449
1450 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1451 {
1452     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1453         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1454         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1455         "Times New Roman", serif;
1456     color: #800000 ;
1457     text-shadow: none ;
1458 }
1459
1460 h1, h2 {
1461     background-color: #fffcf5 ;
1462     background-image: none ;
```

```
1463     border-bottom: 1px solid #808080;
1464     border-top: 2px solid #808080;
1465 }
1466
1467 div.abstracttitle {
1468     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1469         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1470         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1471         "Times New Roman", serif;
1472     color: black ;
1473     text-shadow: none ;
1474 }
1475
1476 span.abstracrunintitle {
1477     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1478         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1479         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1480         "Times New Roman", serif;
1481     color: black ;
1482     text-shadow: none ;
1483 }
1484
1485 div.abstract { font-size: 100% }
1486
1487 .sidebar {
1488     background: #fffcf5;
1489     background-image: none ;
1490     margin: 2em 5% 2em 5%;
1491     padding: 0.5em 1em;
1492     border: none ;
1493     border-top : 1px solid silver;
1494     border-bottom : 1px solid silver;
1495     font-size: 90% ;
1496 }
1497
1498 div.sidebartitle{
1499     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1500         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1501         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1502         "Times New Roman", serif;
1503     color: #800000 ;
1504     text-shadow: none ;
1505     border: none ;
1506 }
1507
1508 .example {
1509     background: #fffcf5;
1510     background-image: none ;
1511     margin: 2em 5% 2em 5%;
1512     padding: 0.5em 1em;
```

```
1513 border: none ;
1514 border-top : 1px solid silver;
1515 border-bottom : 1px solid silver;
1516 }
1517
1518 div.exampletitle{
1519     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1520         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1521         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1522         "Times New Roman", serif;
1523     color: #800000 ;
1524     text-shadow: none ;
1525     border: none ;
1526 }
1527
1528 div.fancyvrblabel{
1529     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1530         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1531         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1532         "Times New Roman", serif;
1533     color: #800000 ;
1534     text-shadow: none ;
1535     border: none ;
1536 }
1537
1538
1539
1540 .verse {
1541     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1542         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1543         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1544         "Times New Roman", serif;
1545 }
1546
1547
1548 figure {
1549     margin: 3ex 5% 3ex 5% ;
1550     padding: 1ex 1em 1ex 1em ;
1551     background-color: #ffffcf5 ;
1552     overflow-x: auto ;
1553     border: none ;
1554 /*     border-top: 1px solid silver; */
1555 /*     border-bottom: 1px solid silver; */
1556 }
1557
1558
1559 figcaption , .lstlisting {
1560     border: none ;
1561 /*     border-top: 1px solid silver ; */
1562 /*     border-bottom: 1px solid silver ; */
```



```
1563     background-color: #fffcf5 ;
1564 }
1565
1566 .tnotes {
1567     background: #fffcf5 ;
1568 }
1569
1570 .theorem {
1571     background: none ;
1572 }
1573
1574 .minipage {
1575     background-color: #fffcf5 ;
1576     border: none ;
1577 }
1578
1579 div.floatrow figure { border: none ; }
1580
1581 figure figure { border: none ; }
1582
1583
1584 nav.toc, nav.lof, nav.lot, nav.lol {
1585     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1586         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1587         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1588         "Times New Roman", serif;
1589 }
1590
1591 nav.sidetoc {
1592     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1593         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1594         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1595         "Times New Roman", serif;
1596     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
1597     border-radius: 0px 0px 0px 20px ;
1598 }
1599
1600 div.sidetoc{
1601     color: #800000 ;
1602 }
1603
1604 header{
1605     background-color: #e0e0e0 ;
1606     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
1607     text-align:center ;
1608 }
1609
1610 footer{
1611     background-color: #e0e0e0 ;
1612     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
```

```

1613     padding: 2ex 1em 2ex 1em ;
1614     clear:right ;
1615     text-align:left ;
1616 }
1617
1618 nav.botnavigation {
1619     background: #dedcd5 ;
1620     border-top: 1px solid black ;
1621 }
1622 \end{VerbatimOut}
1623 % \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.

```

1624 \begin{VerbatimOut}{sample_project.css}
1625 /* ( --- Start of project.css --- ) */
1626 /* A sample project-specific CSS file for lwarp --- ) */
1627
1628 /* Load default lwarp settings: */
1629 @import url("lwarp.css") ;
1630 /* or lwarp_formal.css, lwarp_sagebrush.css */
1631
1632 /* Project-specific CSS setting follow here. */
1633 /* . . . */
1634
1635 /* ( --- End of project.css --- ) */
1636 \end{VerbatimOut}
1637 % \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.

```

1638 \begin{VerbatimOut}{lwarp_html.xdy}
1639 (require "tex/inputenc/latin.xdy")
1640 (merge-rule "\\PS *" "Postscript")
1641 (require "texindy.xdy")

```

```

1642 (require "page-ranges.xdy")
1643 (require "book-order.xdy")
1644 (markup-locref :open "\hyperindexref{" :close "}")
1645 \end{VerbatimOut}
1646 % \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.

```

1647 \begin{VerbatimOut}{lwarp_mathjax.txt}
1648 <!-- https://groups.google.com/forum/#!topic/
1649                               mathjax-users/jUtewUcE2bY -->
1650 <script type="text/x-mathjax-config">
1651 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1652     var seteqsectionDefault = {name: "", num: 0};
1653     var seteqsections = {}, seteqsection = seteqsectionDefault;
1654     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1655     var AMS = MathJax.Extension["TeX/AMSmath"];
1656     TEX.Definitions.Add({
1657     macros: {
1658         seteqsection: "mySection",
1659         seteqnumber: "mySetEqNumber"
1660     }
1661     });
1662
1663     PARSE.Augment({
1664     mySection: function (name) {
1665         seteqsection.num = AMS.number;
1666         var n = this.GetArgument(name);
1667         if (n === "") {
1668             seteqsection = seteqsectionDefault;
1669         } else {
1670             if (!seteqsections["_"+n])
1671                 seteqsections["_"+n] = {name:n, num:0};
1672             seteqsection = seteqsections["_"+n];
1673         }
1674         AMS.number = seteqsection.num;
1675     },
1676     mySetEqNumber: function (name) {
1677         var n = this.GetArgument(name);
1678         if (!n || !n.match(/^\s*[0-9]+\s*$/))
1679             n = ""; else n = parseInt(n)-1;
1680         <!-- $ syntax highlighting -->

```

```

1681         if (n === "" || n < 1)
1682             TEX.Error
1683             ("Argument to "+name+" should be a positive integer");
1684         AMS.number = n;
1685     }
1686 });
1687 MathJax.Hub.Config({
1688   TeX: {
1689     equationNumbers: {
1690       formatTag: function (n)
1691         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
1692       formatID: function (n) {
1693         n = (seteqsection.name+'.'+n).replace
1694           (/[:"'<>&]/g,"").replace(/\./,"");
1695         return 'mjax-eqn-' + n;
1696       }
1697     }
1698   }
1699 });
1700 });
1701 </script>
1702
1703 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1704 <script type="text/x-mathjax-config">
1705   MathJax.Ajax.config.path["Contrib"] =
1706     "https://cdn.mathjax.org/mathjax/contrib";
1707 </script>
1708
1709 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1710                                     tree/master/siunitx -->
1711 <script type="text/x-mathjax-config">
1712   MathJax.Hub.Config({
1713     extensions: ["tex2jax.js","[Contrib]/siunitx/siunitx.js"],
1714     jax: ["input/TeX","output/HTML-CSS"],
1715     tex2jax: {inlineMath: [[["$","$"],["\\(","\\)"]]}},
1716     TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
1717   });
1718 </script>
1719
1720 <script type="text/x-mathjax-config">
1721   MathJax.Hub.Config({
1722     TeX: {
1723       equationNumbers: {
1724         autoNumber: "AMS"
1725       }
1726     }
1727   });
1728 </script>
1729
1730 <script

```

```

1731   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1732 </script>
1733 \end{VerbatimOut}
1734 % \end{Verbatim}% for syntax highlighting

```

69.9 Lwarpmk option

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

```

1735 \begin{LWR@createlwarpmk}

```

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

```

1736 \begin{VerbatimOut}{lwarpmk.lua}
1737 #!/usr/bin/env texlua
1738
1739 -- Copyright 2016-2017 Brian Dunn
1740
1741 -- Print the usage of the lwarpmk command:
1742
1743 printversion = "v0.25"
1744
1745 function printhelp ()
1746 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1747 end
1748
1749 function printusage ()
1750 print ( [[
1751
1752 lwarpmk print [project]: Compile a print version.
1753 lwarpmk printindex [project]: Process the index for the print version.
1754 lwarpmk html [project]: Compile an HTML version.
1755 lwarpmk htmlindex [project]: Process the index for the html version.
1756 lwarpmk again [project]: Touch the source code to trigger recompiles.
1757 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
1758 lwarpmk pdftohtml [project]:
1759     For use with latexmk or a Makefile:
1760     Convert project_html.pdf to project_html.html and
1761     individual HTML files.
1762 lwarpmk clean [project]: Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
1763 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1764 lwarpmk -h: Print this help message.
1765 lwarpmk --help: Print this help message.
1766
1767 ]] )
1768 printconf ()
1769 end

```

```

1770
1771 -- Print the format of the configuration file lwarpmk.conf:
1772
1773 function printconf ()
1774 print ( [[
1775 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1776 --
1777 opsystem = "Unix"    (or "Windows")
1778 latexname = "pdflatex" (or "lualatex", or "xelatex")
1779 sourcename = "projectname" (the source-code filename w/o .tex)
1780 homehtmlfilename = "index" (or perhaps the project name)
1781 htmlfilename = "" (or "projectname" - filename prefix)
1782 uselatexmk = "false" (or "true" to use latexmk to build PDFs)
1783 --
1784 Filenames must contain only letters, numbers, underscore, or dash.
1785 Values must be in "quotes".
1786
1787 ]] ) ;
1788 end
1789
1790
1791 -- Split one large sourcefile into a number of files,
1792 -- starting with destfile.
1793 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1794
1795 function splitfile (destfile,sourcefile)
1796 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1797 io.input(sourcefile)
1798 io.output(destfile)
1799 for line in io.lines() do
1800 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|") ;
1801 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1802 io.output(newfilename) ;
1803 else -- not a splitpoint
1804 io.write (line .. "\n") ;
1805 end
1806 end -- do
1807 end -- function
1808
1809 -- Incorrect value, so print an error and exit.
1810
1811 function cvalueerror ( line, linenum , cvalue )
1812     print ( linenum .. " : " .. line ) ;
1813     print ("lwarpmk: incorrect variable value \" " .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
1814     printconf () ;
1815     os.exit(1) ;
1816 end
1817
1818 -- Load settings from the project's "lwarpmk.conf" file:
1819

```

```
1820 function loadconf ( )
1821 -- Default configuration filename:
1822 local conffile = "lwarpmk.conf"
1823 -- Optional configuration filename:
1824 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1825 -- Verify the file exists:
1826 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1827 print("lwarpmk: " .. conffile .. " does not exist.")
1828 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1829 printhelp ( ) ;
1830 os.exit(1) -- exit the entire lwarpmk script
1831 else -- file exists
1832 -- Read the file:
1833 print ("lwarpmk: Reading " .. conffile .. ".")
1834 io.input(conffile) ;
1835 -- Scan each line:
1836 local linenum = 0
1837 for line in io.lines() do -- scan lines
1838 linenum = linenum + 1
1839 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w-_]*)\"") ;
1840 -- Error if incorrect enclosing characters:
1841 if ( i == nil ) then
1842 print ( linenum .. " : " .. line ) ;
1843 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
1844 printconf ( ) ;
1845 os.exit(1) ;
1846 end
1847 if ( cvarname == "opsystem" ) then
1848 -- Verify choice of opsystem:
1849 if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1850 opsystem = cvalue
1851 else
1852 cvalueerror ( line, linenum , cvalue )
1853 end
1854 elseif ( cvarname == "latexname" ) then
1855 -- Verify choice of LaTeX compiler:
1856 if (
1857 (cvalue == "pdflatex") or
1858 (cvalue == "xelatex") or
1859 (cvalue == "lualatex")
1860 ) then
1861 latexname = cvalue
1862 else
1863 cvalueerror ( line, linenum , cvalue )
1864 end
1865 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1866 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1867 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1868 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1869 else
```

```

1870 print ( linenum .. " : " .. line ) ;
1871 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
1872 printconf ( ) ;
1873 os.exit(1) ;
1874 end
1875 end -- do scan lines
1876 end -- file exists
1877 -- Select some operating-system commands:
1878 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1879   rmname = "rm"
1880   mvname = "mv"
1881   touchnamepre = "touch"
1882   touchnamepost = ""
1883   dirslash = "/"
1884   opquote= "\"'"
1885 elseif opsystem=="Windows" then -- For Windows
1886   rmname = "DEL"
1887   mvname = "MOVE"
1888   touchnamepre = "COPY /b"
1889   touchnamepost = "+,,"
1890   dirslash = "\\"
1891   opquote= "\""
1892 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1893 end --- for Windows
1894 end -- loadconf
1895
1896
1897 function refreshdate ( )
1898   os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
1899 end
1900
1901
1902 -- Scan the LaTeX log file for the phrase "Rerun to get",
1903 -- indicating that the file should be compiled again.
1904 -- Return true if found.
1905
1906 function reruntoget (filesourse)
1907   io.input(filesourse)
1908   for line in io.lines() do
1909     if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1910   end
1911   return false
1912 end
1913
1914 -- Compile one time, return true if should compile again.
1915 -- fsuffix is "" for print, "_html" for HTML output.
1916
1917 function onetime (fsuffix)
1918   print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
1919   err = os.execute(

```



```

1920 --      "echo " ..
1921      latexname .. " " .. sourcename..fsuffix )
1922 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1923 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
1924 end
1925
1926 -- Compile up to five times.
1927 -- fsuffix is "" for print, "_html" for HTML output
1928
1929 function manytimes (fsuffix)
1930 if onetime(fsuffix) == true then
1931 if onetime(fsuffix) == true then
1932 if onetime(fsuffix) == true then
1933 if onetime(fsuffix) == true then
1934 if onetime(fsuffix) == true then
1935 end end end end end
1936 end
1937
1938 -- Exit if the given file does not exist.
1939
1940 function verifyfileexists (filename)
1941 if (lfs.attributes ( filename , "modification" ) == nil ) then
1942 print ( "lwarpmk: " .. filename .. " not found." ) ;
1943 os.exit (1) ;
1944 end
1945 end
1946
1947 -- Convert <project>_html.pdf into HTML files:
1948
1949 function pdftohtml ()
1950     -- Convert to text:
1951     print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html"
1952     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf " .. sou
1953     -- Split the result into individual HTML files:
1954     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
1955 end
1956
1957 -- Remove auxiliary files:
1958
1959 function removeaux ()
1960     os.execute ( rmname .. " " ..
1961         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
1962         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
1963         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
1964         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
1965         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
1966         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
1967         sourcename .. ".log " .. sourcename .. "_html.log "
1968     )
1969 end

```

```

1970
1971
1972
1973 -- Create lateximages based on lateximages.txt:
1974 function createlateximages ()
1975 print ("lwarpmk: Creating lateximages.")
1976 io.input("lateximages.txt")
1977 -- Create the lateximages directory, ignore error if already exists
1978 err = os.execute("mkdir lateximages")
1979 -- Scan lateximages.txt
1980 for line in io.lines() do
1981 -- lwimpage is the page number in the PDF which has the image
1982 -- lwimnum is the sequential lateximage number to assign for the image
1983 i,j,lwimpage,lwimnum = string.find (line,"|(.*)|(.*)|")
1984 -- For each entry:
1985 if ( i~=nil ) then
1986 -- Separate out the image into its own single-page pdf:
1987 err = os.execute(
1988 "pdfseparate -f " .. lwimpage .. " -l " ..
1989 lwimpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
1990 -- Crop the image:
1991 err = os.execute(
1992 "pdfcrop -- hires lateximagetemp-" .. lwimpage .. ".pdf lateximage-" .. lwimnum .. ".pdf")
1993 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1994 -- Convert the image to svg:
1995 err = os.execute(
1996 "pdftocairo -svg lateximage-" .. lwimnum .. ".pdf lateximage-" .. lwimnum .. ".svg")
1997 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1998 -- Move the result into lateximages/:
1999 err = os.execute(
2000 mvname .. " lateximage-" .. lwimnum .. ".svg lateximages" .. dirslash )
2001 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2002 -- Remove the temporary files:
2003 err = os.execute(
2004 rmname .. " lateximage-" .. lwimnum .. ".pdf lateximagetemp-" .. lwimpage .. ".pdf")
2005 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2006 end
2007 end -- do
2008 end -- function
2009
2010
2011
2012
2013 -- lwarpmk --version :
2014
2015 if (arg[1] == "--version") then
2016 print ( "lwarpmk: " .. printversion )
2017
2018 else -- not -- version
2019

```

```

2020 -- print intro:
2021
2022 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2023
2024 -- lwarpmk print:
2025
2026 if arg[1] == "print" then
2027 loadconf ()
2028 if ( uselatexmk == "true" ) then
2029     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\" " .. latexname .. " %0 %S\" " .. sourcenam
2030     print ("lwarpmk: Done.")
2031 else -- not latexmk
2032     verifyfileexists (sourcename .. ".tex") ;
2033     -- See if up to date:
2034     if (
2035         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2036         (
2037             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2038             lfs.attributes ( sourcename .. ".pdf" , "modification" )
2039         )
2040     ) then
2041         -- Recompile if not yet up to date:
2042         manytimes("")
2043         print ("lwarpmk: Done.") ;
2044     else
2045         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
2046     end
2047 end -- not latexmk
2048
2049 -- lwarp printindex:
2050 -- Compile the index then touch the source
2051 -- to trigger a recompile of the document:
2052
2053 elseif arg[1] == "printindex" then
2054 loadconf ()
2055 print ("lwarpmk: Processing the index.")
2056 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. ".idx")
2057 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2058 refreshdate ()
2059 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2060 print ("lwarpmk: Done.")
2061
2062 -- lwarpmk html:
2063
2064 elseif arg[1] == "html" then
2065 loadconf ()
2066 if ( uselatexmk == "true" ) then
2067     -- The recorder option is required to detect changes in <project>.tex
2068     -- while we are loading <project>_html.tex.
2069     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "

```

```

2070     .. "-e "
2071     .. opquote .. "$makeindex = q/texindy -M lwarp_html.xdy/" .. opquote
2072     .. " -pdflatex=\"\" .. latexname .." %O %S\" "
2073     .. sourcename .. "_html.tex" );
2074     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2075     pdftohtml ()
2076     print ("lwarpmk: Done.")
2077 else -- not latexmk
2078     verifyfileexists ( sourcename .. ".tex" ) ;
2079     -- See if exists and is up to date:
2080     if (
2081         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2082         (
2083             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2084             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2085         )
2086     ) then
2087         -- Recompile if not yet up to date:
2088         manytimes("_html")
2089         pdftohtml ()
2090         print ("lwarpmk: Done.")
2091     else
2092         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2093     end
2094 end -- not latexmk
2095
2096 elseif arg[1] == "pdftohtml" then
2097     loadconf ()
2098     pdftohtml ()
2099
2100 -- lwarpmk htmlindex:
2101 -- Compile the index then touch the source
2102 -- to trigger a recompile of the document:
2103
2104 elseif arg[1] == "htmlindex" then
2105     loadconf ()
2106     print ("lwarpmk: Processing the index.")
2107     os.execute("texindy -M lwarp_html.xdy " .. sourcename .. "_html.idx")
2108     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2109     refreshdate ()
2110     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2111     print ("lwarpmk: Done.")
2112
2113 -- lwarpmk limages:
2114 -- Scan the lateximages.txt file to create lateximages,
2115 -- then touch the source to trigger a recompile.
2116
2117 elseif arg[1] == "limages" then
2118     loadconf ()
2119     print ("lwarpmk: Processing images.")

```

```
2120 createlateximages ()
2121 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2122 refreshdate ()
2123 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2124 print ("lwarpmk: Done.")
2125
2126 -- lwarpmk again:
2127 -- Touch the source to trigger a recompile.
2128
2129 elseif arg[1] == "again" then
2130 loadconf ()
2131 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2132 refreshdate ()
2133 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2134 print ("lwarpmk: Done.")
2135
2136 -- lwarpmk clean:
2137 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2138
2139 elseif arg[1] == "clean" then
2140 loadconf ()
2141 removeaux ()
2142 print ("lwarpmk: Done.")
2143
2144 -- lwarpmk cleanall
2145 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2146 -- and also project.pdf, *.html
2147
2148 elseif arg[1] == "cleanall" then
2149 loadconf ()
2150 removeaux ()
2151 os.execute ( rmname .. " " ..
2152     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2153     "*.html"
2154 )
2155 print ("lwarpmk: Done.")
2156
2157 -- lwarpmk with no argument :
2158
2159 elseif (arg[1] == nil) then
2160 printhelp ()
2161
2162 -- lwarpmk -h or lwarpmk --help :
2163
2164 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2165 printusage ()
2166
2167 else
2168 print ("lwarpmk: Unknown command \"" .. arg[1] .. "\".\n")
2169 printhelp ()
```

```
2170 end
2171
2172 end -- not --version
2173 \end{VerbatimOut}
2174 % \end{Verbatim}% for syntax highlighting

2175 \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 141.1.

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   \@xp\@ifdefinable\csname #2\endcsname{%
17     \global\@xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1\}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25       \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27 }%
28 \@tempa
29 }

```

Patched to enclose with CSS:

```

30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for CSS:

```

45 \def\@begintheorem#1#2[#3]{%
46   \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{theoremdendmark}{\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\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

Package 12

lwarp-dcolumn.sty

79 Dcolumn

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

Package 13

lwarp-draftwatermark.sty

80 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 14

lwarp-ellipsis.sty

81 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 15

lwarp-emptypage.sty

82 Emptypage

Pkg emptypage emptypage is ignored.

for **HTML output**: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

Package 16

lwarp-enumerate.sty

83 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 17

lwarp-epigraph.sty

84 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 18

lwarp-eso-pic.sty

85 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 19

lwarp-everypage.sty

86 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 20

lwarp-extramarks.sty

87 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 21

lwarp-fancyhdr.sty

88 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 22

lwarp-float.sty

89 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 23


lwarp-floatrow.sty

90 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 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 24

lwarp-fontenc.sty

91 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 25

lwarp-fontspec.sty

92 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 26

lwarp-framed.sty

93 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 27

lwarp-ftnright.sty

94 Ftnright

Pkg `ftnright` `ftnright` is ignored.

for **HTML output:** Discard all options for `lwarp-ftnright`:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

Package 28

lwarp-geometry.sty

95 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 29

lwarp-graphics.sty

96 Graphics

Pkg graphics graphics is not used. The user is required to use graphicx instead.

for HTML output: 1 \LWR@loadnever{graphics}{graphicx}

Package 30

lwarp-graphicx.sty

97 Graphicx

Pkg graphicx graphicx is emulated during HTML output, and the graphicx package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

Package 31

lwarp-hyperref.sty

98 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 32

lwarp-indentfirst.sty

99 Indentfirst

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

Package 33

lwarp-inputenc.sty

100 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 34

lwarp-keyfloat.sty

101 Keyfloat

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```

2 \AtBeginDocument{

3 % \newsavebox{\KFLT@envbox}
4
5 \let\KFLT@boxinner\relax
6 \let\endKFLT@boxinner\relax
7
8 \NewEnviron{KFLT@boxinner}
9 {%
10 \LWR@traceinfo{kflt@boxinner}%
11 \LWR@stoppars%
12 \KFLT@frame{\BODY}%
13 \LWR@startpars%
14 \LWR@traceinfo{ended kflt@boxinner}%
15 }

16 \RenewDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
17 {% start
18 \LWR@maybeinthisfloat%
19 % \BlockClass{marginblock}
20 \LWR@stoppars%
21 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
22 \LWR@startpars%
23 \captionsetup{type=#2}%
24 }
25 {
26 % \endBlockClass
27 \LWR@htmldivclassend{div}
28 }

29 }% AtBeginDocument

```

Package 35

lwarp-layout.sty

102 Layout

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

Package 36

lwarp-letterspace.sty

103 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 37

lwarp-lips.sty

104 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 38

lwarp-listings.sty

105 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\lst@name\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 \@tempskipa}%
73 %             \def\lst@vskip{\nobreak\@vskip\@tempskipa\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 39

lwarp-longtable.sty

106 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 40

lwarp-lscape.sty

107 Lscape

Pkg lscape lscape is nullified.
 for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

Package 41

lwarp-ltcaption.sty

108 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 42

lwarp-mcaption.sty

109 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 43


lwarp-mdframed.sty

110 Mdframed

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 }

```

[*numberedlike*] {*caption*} [*within*]

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

```

[*numberedlike*] {*caption*} [*within*]

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 44

lwarp-microtype.sty

111 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 45

lwarp-multicol.sty

112 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 46

lwarp-multirow.sty

113 Multirow

Pkg multirow multirow is emulated during HTML output, and the multirow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multirow}

Package 47

lwarp-nameref.sty

114 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 48

lwarp-needspace.sty

115 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 49

lwarp-newclude.sty

116 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 50

lwarp-newunicodechar.sty

117 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 51

lwarp-nextpage.sty

118 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 52

lwarp-nowidow.sty

119 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 53

lwarp-ntheorem.sty

120 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}{\booltrue{LWR@theoremamsthm}}
20 \DeclareOption{hyperref}{\booltrue{LWR@theoremamsthm}}
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     \theoremstyle{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30 }%
31 {
32 \theoremstyle{#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   \ifthen@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 54

lwarp-pagenote.sty

121 Pagenote

Pkg pagenote pagenote is used by lwarp, and its user-interface macros have been nullified.

Both footnotes and pagenotes appear as footnotes in the HTML output.

Any manipulation of pagenote in the user's document after the preamble should be encased within a `warpprint` environment.

Discard all options for lwarp-pagenote:

for HTML output: 1 \LWR@ProvidesPackageDrop{pagenote}

\pagenote is already modified to work with lwarp.

```
2 \renewcommand*{\makepagenote}{}
3 \RenewDocumentCommand{\printnotes}{s}{}
4 \renewcommand{\printnotes}[2] [] {}
5 \renewcommand{\addtonotes}[1] {}
```

Package 55

lwarp-parskip.sty

122 Parskip

Pkg `parskip` `parskip` is ignored.

for HTML output: Discard all options for `lwarp-parskip`.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

Package 56

lwarp-placeins.sty

123 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 57

lwarp-ragged2e.sty

124 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 58

lwarp-rotating.sty

125 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 59

lwarp-setspace.sty

126 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 60

lwarp-showidx.sty

127 Showidx

Pkg `showidx` `showidx` is ignored.

for **HTML output**: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

Package 61

lwarp-showkeys.sty

128 Showkeys

Pkg `showkeys` `showkeys` is ignored.

for **HTML output**: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}
```

Package 62

lwarp-sidecap.sty

129 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 63

lwarp-subfig.sty

130 Subfig

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

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

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

for HTML output: Accept all options for lwarp-subfig:

```
1 \LWR@ProvidesPackagePass{subfig}
```

```
\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with \hfill between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21  \setbox\@tempboxa \hbox{#4}%
22    \@tempdima=\wd\@tempboxa
```

```

23 \ifundefined{FBsc@max}{}%
24 {\global\advance\Xhsize-\wd\@tempboxa
25 \dimen@=\ht\@tempboxa
26 \advance\dimen@\dp\@tempboxa
27 \ifdim\dimen@>\FBso@max
28 \global\FBso@max\dimen@
29 \fi}%
30 \vtop\bgroup
31 \vbox\bgroup
32 \ifcase\@tempcnta
33 \@minipagefalse
34 \or
35 \vskip\sf@top
36 \or
37 \ifdim \lastskip=\z@ \else
38 \@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 {\langle caption label \rangle} {\langle caption text \rangle}

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\@captype\space
209     \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210     \@nameuse{p@sub\@captype}%
211     \@nameuse{thesub\@captype}.)}}

```

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 64

lwarp-tabularx.sty

131 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 65

lwarp-textpos.sty

132 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 66

lwarp-theorem.sty

133 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 67

lwarp-threeparttable.sty

134 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 68

lwarp-tikz.sty

135 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 69

lwarp-titles.sty

136 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 o m m m m m}{}{}
```

```
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}{}
```

```
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}{}
```

```
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}{}
```

```
28 \newcommand{\newmarkset}[1]{}{}
```

```
29 \NewDocumentCommand{\newextramarkset}{s m m}{}{}
```

```
30 \newcommand{\botextramarks}[1]{}{}
```

```
31 \newcommand{\topextramarks}[1]{}{}
```

```
32 \newcommand{\firstextramarks}[1]{}{}
```

```
33 \newcommand{\nexttoextramarks}[1]{}{}
```

```
34 \newcommand{\outerextramarks}[1]{}{}
```

```
35 \newcommand{\innerextramarks}[1]{}{}
```

Package 70

lwarp-titlesec.sty

137 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 71

lwarp-titletoc.sty

138 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 72

lwarp-titling.sty

139 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 `\@bsmttitleempty`:

```
2 \let\LWR@orig@bsmttitleempty\@bsmttitleempty
3 \renewcommand*{\@bsmttitleempty}{%
4 \LWR@orig@bsmttitleempty%
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 73

lwarp-tocloft.sty

140 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{\pagenumberson}[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 74

lwarp-trivfloat.sty

141 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}`

141.1 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 75

lwarp-verse.sty

142 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 76

lwarp-wallpaper.sty

143 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 77

lwarp-wrapfig.sty

144 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 78

lwarp-xcolor.sty

145 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 79

lwarp-xfrac.sty

146 Xfrac

Pkg **xfrac** Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 font size In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by pdftotext to a plain text file. While the text may be in any font which pdftotext can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine \xfracHTMLfontsize with a different em size.

\sfrac [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a lateximage, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

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			
General: 2016/03/08 Initial version	1	floatrow: Added.	339
v0.11		Docs: Commands for a successful HTML conversion.	56
General: 2016/03/11	1	Docs: Commands into a warpprint environment.	55
Added section:		Docs: Newclude limitations.	67
Operating-System portability.	90	Docs: Table: Cross-referencing data structures.	191
Added section: Selecting the operating system.	54	Docs: Table: Float data structures.	201
Test Suite: limages and index in README.txt	1	Docs: Trademarks section.	270
Test Suite: MS-Windows in README.txt	1	Docs: Troubleshooting cross-references.	70
v0.12		Pagenote: Assigned cleveref name.	247
\LWR@newhtmlfile: Bugfix: TOC with numbered files.	121	Test Suite: Assigned cleveref name for Test Float.	1
General: 2016/03/14	1	Test Suite: Floatrow	1
Global: Uses \p@(type) in float captions.	1	v0.15	
Test Suite: Sub-figures	1	\printpublished: No div if empty.	143
v0.13		\printsubtitle: No div if empty.	143
\CaptionSeparator: Fix for newer babel package.	203	General: 2016/04/06	1
\LWR@LwarpStart: \up and \fup	133	Added	344
\published: Default to empty published.	139	Ampersand (&): Fixed handling when passed as an argument.	167
\subtitle: Default to empty subtitle.	140	Docs: Added warning icons for items needing special attention.	74
General: 2016/03/24	1	Docs: Clarify print/HTML output.	54
Removed package: subfig	1	Docs: Moved the supported functions table to the introduction.	22
Test Suite: Ordinals, Subcaption	1	Files: lwarp_formal.css added.	1
tikzpicture: Fix dollar-redefined bug for newer package.	411	Fix: steps counter	343
v0.14		Fixed & handling.	341
\LWR@htmlsectionfilename: Fix: Links to home page.	97	Test Suite: test_suite_formal.css file added.	1
\noteentry: Fix for label inside a footnote.	116		
General: 2016/03/31	1		

v0.16	
General: 2016/04/11	1
\lwr@titlingpage: Improved	
print-output spacing.	144
Added XeLaTeX, LuaLaTeX	
support.	81
Docs: Font and UTF-8 support.	53
Docs: Moved location of	
\usepackage{lwrap}.	54
Docs: Text not converting.	70
Fix: amsmath options clash	84
Fix: newtxmath compatibility.	85
Lwrap no longer selects fonts.	53, 81
Removed package: suffix	1
Test Suite: Improved	
titlingpage.	144
Test Suite: Lwrap no longer	
selects fonts.	1
Test Suite: Supports XeLaTeX,	
LuaLaTeX.	1
xfrac: Adjusted for the use of	
any font:	433
v0.17	
\lwr@htmlsectionfilename: Fix:	
Links when entire doc is one	
HTML page.	97
General: 2016/04/14	1
mdframed: Added.	366
Test Suite: Fix: Print-version	
front-matter page numbers.	1
Test Suite: Mdfamed	1
v0.18	
\lwr@hspace: \hspace supported.	261
\lwr@includegraphicsb: Add:	
svgz file extension.	240
em, ex, %, px dimensions	
preserved.	240
Fix: \linewidth, \textwidth,	
\textheight inside a	
minipage.	240
Improved HTML output	
linebreaks.	240
\lwr@myshorttoc: Reorganize	
\HomeHTMLFilename logic.	207
\lwr@newhtmlfile: Sidetoc after	
title, improving responsive	
design.	121
\lwr@requesttoc: Reorganize	
\HomeHTMLFilename logic.	135
\lwr@subhyperref: Improved	
HTML output linebreaks.	198
\lwr@subhyperrefclass: Improved	
HTML output linebreaks.	199
\lwr@subinlineimage: Surpress	
extra space.	200
General: 2016/05/19	1
File: lwrap.css: Improved TOC	
outline display.	1
Files: lwrap.css and	
lwrap_formal.css: Improved	
responsive design.	1
Microtype disabled during	
HTML generation	81
PDF Unicode input characters.	75
Test Suite: Verse package	1
\lateximage: pdfcrop: --hires	
added.	232
Reorganize \HomeHTMLFilename	
logic.	232
Surpress extra space.	232
verse: Supports verse, memoir	
packages.	424
minipage: Fix: \linewidth,	
\textwidth, \textheight	
inside a minipage.	252
v0.19	
\HTMLFileName: Docs: Escape	
filename underscores.	96
\HomeHTMLFileName: Docs: Escape	
filename underscores.	96
\lwr@LwrapStart: Enabled \\	
equal to \newline.	132
\lwr@doubledollar: MathJax	
support.	218
\lwr@filestart:	
lwrap_mathjax.txt loaded.	130
\lwr@hspace: Fix: \hspace length	
computations.	261
\lwr@minipagestartpars:	
Surpresses paragraph tags	
between minipages.	260
\lwr@singledollar: MathJax	
support.	218
\lwr@tabledatamultirowtag:	
Added optional args.	186
\LateximageFontSizeName: Add:	
User-adjustable	
math/lateximage font size.	230

<code>\minipagefullwidth</code> : Added: No width tag for the next minipage in HTML.	252	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space.	261
<code>\rule</code> : Added	264	<code>\LWR@htmldivclass</code> : Added optional style.	104
<code>\warpHTMLonly</code> : Added.	76	<code>\LWR@htmlelementclass</code> : Added optional style.	104
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code>	76	<code>\LWR@htmlsectionfilename</code> : HTMLFileName: removed additional trailing '-', and may be empty.	97
General: 2016/06/08	1	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	97
Added.	432	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	240
Avoids MathJax.	217	Fix: Expands filename.	240
cleveref: Loaded <code>\AtEndPreamble</code>	247	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	184
CSS for table note item.	410	<code>\LWR@section</code> : Combined higher-level sections together into files.	124
Docs: multirow browser bug workaround.	63, 164	<code>\LWR@subhtml elementclass</code> : Factored code.	103
Docs: Math options.	54	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	97
Docs: Table: Cross-referencing data structures, updated.	191	<code>\cpagerefFor</code> : User-redefinable word for page references.	247
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\dotfill</code> : Inserts an ellipsis.	258
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added	1	<code>\hfill</code> : Inserts a <code>\qqquad</code>	258
Introduction: MathJax support mentioned.	21	<code>\hrulefill</code> : Inserts a short rule.	258
MathJax support added.	219, 223, 225	<code>\hyperindexref</code> : Print mode provided in case hyperref not used.	214
Options: <code>mathsvg</code> and <code>mathjax</code>	77	<code>\pageref</code> : Added.	197
Supports colored <code>\rule</code>	429	<code>\tracinglwarp</code> : Added.	89
titles: null <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	85	<code>\warpOSwindows</code> : Auto-detects operating system.	91
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	411	General: 2017/02/09	1
<code>equation</code> : MathJax support.	222	<code>afterpage</code> : Added.	321
v0.20 <code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	106	<code>alltt</code> : Added.	323
<code>\HTMLFileName</code> : No longer escape underscores.	96	<code>bookmark</code> : Added.	326
<code>\HomeHTMLFileName</code> : No longer escape underscores.	96	caption and subcaption supported.	1
<code>\InlineClass</code> : Renamed from "inlineclass".	106	cleveref and referencing patches: Applied <code>\AfterEndPreamble</code>	247
<code>\LWR@LwarpStart</code> : Fix: math cross references.	134	<code>draftwatermark</code> : Added.	330
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	109	<code>eso-pic</code> : Added.	334
No break tags in the start/end of a tabular.	109	<code>everypage</code> : Added.	335
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	259	<code>extramarks</code> : Added.	335
<code>\LWR@filestart</code> : Adds meta description.	130	<code>fancyhdr</code> : Added.	336

hyperref: Additional user macros.	351	Tabular: <code>\unskip</code> extra spaces.	164
keyfloat: Added.	355	Test Suite: HTML meta descriptions.	1
letterspace: User-interface emulated.	356	Verbatim: Added.	155
listings: Added.	358	verbatim: Added.	153
lrcaption: Added.	364	BlockClass: Added optional style.	105
lwarp-newproject: Added.	271	Renamed from "blockclass".	105
microtype: User-interface emulated.	373	LWR@nestspan: Fix: Minipages inside a span.	101
needspace: Added.	377	v0.21	
nowidow: Added.	379	<code>\LWR@LwarpStart</code> : Changed <code>lateximages</code> to a .txt file.	132
pagenote: Added.	391	<code>\LWR@filestart</code> : Skip title if not given.	130
placeins: Added.	392	<code>\LWR@newhtmlfile</code> : Skip title if not given.	121
ragged2e: Added.	393	<code>\marginpar</code> : Fixed source listing.	118
setspace: Improved support.	395	General: 2017/02/23	1
textpos: Added.	405	fontenc: Added.	345
titles: Added.	412	fontspec: Added.	345
titlesec: Added.	414	inputenc: Added.	354
titletoc: Added.	415	newclue: Added.	377
titling: Improved compatibility.	416	newunicodechar: Added.	378
tocloft: Added.	417	Add: Errors for misplaced packages.	78
wallpaper: Added.	426	Docs: Added <code>internet</code> class.	27
wrapfig: Added.	427	Docs: Added <code>TeX2page</code> , <code>GladTeX</code>	27
Added @, <, > columns.	164	Docs: Installing on Windows.	33
Added <code>\NewHTMLdescription</code>	114	File <code>lwarp_tutorial.txt</code> added.	37
Added single-expansion data arrays.	95	lwarpmk: Fix: <code>lwarpmk</code> again for Windows.	309
Code factored into independent <code>lwarp_html</code> files.	318	lwarpmk: Fix: <code>lwarpmk</code> <code>limages</code> for Windows.	309
Docs: Examples for generating HTML file names.	47	lwarpmk: Fix: <code>lwarpmk</code> uses <code>lateximages</code> text file instead of shell script.	309
Docs: Improved index.	1	v0.22	
Enhanced titling support.	141	<code>\LWR@parseDcolumn</code> : Added tabular D column.	173
File: <code>lwarp.css</code> : Minor fixes for validation.	1	<code>\LWR@parsebangcolumn</code> : Added tabular ! column.	170
File: <code>lwarpmk</code> used to compile print, HTML, indexes, and <code>lateximages</code>	1	<code>\LWR@parsetablecols</code> : Unknown table column types become l. Added tabular D, !, X columns.	175
Fix: <code>\linewidth</code> in a floatrow.	342	<code>\LWR@printmccoldata</code> : Added tabular D, !, and X columns.	181
Improved float caption type handling.	337	General: 2017/03/02	1
Moved sidebar and example code to test suite.	1	abstract: Added.	319
Page geometry set to 6in wide with large margins.	82	changepage: Added.	329
Parallel versions of aux files for print/HTML.	1		
Removed reliance on make, grep, gawk.	1		

dcolumn: Added.	329	tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	411
enumerate: Added.	332	Docs: Filename underscore.	46, 56
ftnright: Added.	349	No longer preloads subcaption; conflicted with subfig.	84
geometry: Nullified commands.	349	picture: Fix for inline images.	248
indentfirst: Added.	354	tikzpicture: Fix for inline images.	411
layout: Added.	356	v0.25	
lscap: Added.	364	\LWR@loadnever: Added the ability to prevent conflicting packages.	79
mcaption: Added.	365	\addcontentsline: Handles theorems.	206
nameref: Added.	376	General: 2016/03/22	1
nextpage: Added.	378	amsthm: Added.	324
parskip: Added.	392	ccaption: Prevented.	328
showkeys: Added.	396	ellipsis: Added.	330
sidecap: Added.	397	emptypage: Added.	331
tabularx: Added.	404	framed: Added.	346
varioref: Supported.	66	lips: Added.	357
verse: Added.	424	mdframed: Help avoid hyphenation.	367
v0.23		ntheorem: Added.	380
\LWR@parsetablecols: Fix for vert bar column type.	175	showidx: Added.	396
\LWR@printmccoldata: Fix for vert bar column type.	181	theorem: Added.	406
General: 2017/03/02	1	Basic L ^A T _E X theorems: improved CSS.	159
v0.24		Docs: Adds credits for patched code.	1
\LWR@hspace: Add: \hspace \fill converts to 2em	261	Docs: Testing lwrap.	69
\LWR@htmlfileref: Fix: Index links while \tracinglwrap.	194	Fix: Allows XE ^L A _T E _X and Lua ^L A _T E _X to preload graphics and graphicx.	79
\hypertocfloat: List of floats responds to lofdepth, lotdepth.	212		
General: 2017/03/15	1		
floatrow: Support for subfig.	339		
subfig: Added.	398		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		
\\$	<i>217</i>	\@@setcref <i>3798</i>
\&	<i>96, 2106</i>	\@@setcrefrange <i>3799</i>
\(.	<u>3095</u>	@author <i>138</i>
\)	<u>3095</u>	\@begintheorem <u>1965</u>
\@@setcpageref	<u>3802</u>	\@capttype <u>2823</u>
		\@currentlabel <i>117</i>

<code>\@currentlabelname</code>	2617	<code>\BaseJobname</code>	498
<code>@date</code>	138	<code>BaseJobname</code> (option) [lwarp]	77
<code>\@dlbfloat</code>	2806	baseline	
<code>\@endtheorem</code>	1973	tabular	174
<code>\@float</code>	2806	<code>\bfseries</code>	3970
<code>\@fnsymbol</code>	1698	<code>\BibTeX</code>	4152
<code>\@makecaption</code>	2825	<code>BlockClass</code> (environment)	712
<code>\@maketitle</code>	1709	<code>\BlockClassSingle</code>	724
<code>\@nameedef</code>	69	<code>bookmark</code> (package)	326
<code>\@opargbegintheorem</code>	1969	<code>booktabs</code> (package)	328
<code>@published</code>	138	booleans:	
<code>@subtitle</code>	138	<code>CombineHigherDepths</code>	47, 118
<code>@title</code>	138	<code>FileSectionNames</code>	47, 97
<code>\\</code>	259	<code>LWR@amsmultline</code>	223
<code>\$</code>	217	<code>LWR@doingapar</code>	107
<code>\$\$</code>	217	<code>LWR@doinghline</code>	165
		<code>LWR@doingstartpars</code>	107
		<code>LWR@doingtbrule</code>	165
		<code>LWR@freezethisfloat</code>	203
		<code>LWR@infloatrow</code>	240
		<code>LWR@intabularmetadata</code>	166
		<code>LWR@minipagefullwidth</code>	251
		<code>LWR@minipagethispar</code>	252
		<code>LWR@skippingmrowcell</code>	166
		<code>LWR@starredlongtable</code>	183
		<code>LWR@startedrow</code>	165
		<code>LWR@tableparcell</code>	165
		<code>LWR@tracinglwarp</code>	89
		<code>LWR@validtablecol</code>	173
		<code>LWR@verbtags</code>	153
		<code>mathjax</code>	76
		<code>usingOSwindows</code>	91
		<code>warpingHTML</code>	76
		<code>warpingprint</code>	76
		bugs	70
		<code>BVerbatim</code> (environment)	1951
		C	
		<code>calc</code> (package)	83
		<code>caption</code> (package)	84
		<code>\caption@begin</code>	2856
		<code>\caption@end</code>	2856
		<code>\captionlistentry</code>	2860
		<code>\captionof</code>	2890
		<code>\CaptionSeparator</code>	2824
		<code>ccaption</code> (package)	328
		<code>center</code> (environment)	3529
		<code>\centering</code>	133
		<code>changepage</code> (package)	329
		<code>\chapter</code>	1216
		A	
<code>abstract</code> (environment)	1739		
<code>abstract</code> (package)	319		
<code>\abstractname</code>	48, 1738		
accents	123		
<code>\addcontentsline</code>	2877		
Adobe (program)	29		
<code>\affiliation</code>	1407		
<code>afterpage</code> (package)	321		
algorithmic			
with newfloat, trivfloat	422		
<code>algorithmicx</code> (package)	322		
<code>align</code> (environment)	3311		
<code>align*</code> (environment)	3333		
<code>alltt</code> (package)	323		
<code>\AmS</code>	4157		
<code>amsmath</code> (package)	84		
<code>amsthm</code> (package)	324		
<code>\and</code>	138		
<code>array</code> (package)	191		
<code>AsciiDoc</code> (program)	28		
<code>AsciiDoctor</code> (program)	28		
<code>Asciidoctor-LaTeX</code> (program)	28		
<code>\attrib</code>	66, 152, 424		
<code>\attribution</code>	1748		
<code>\author</code>	58, 138, 1414		
<code>autosec</code>	123		
		B	
<code>babel</code> (package)	67		
<code>\backmatter</code>	1098		
<code>BaseJobName</code> (option) [lwarp-			
newproject]	271		

<code>\citetitle</code>	1755	Deja Vu	53
classes:		description	
internet	27	HTML meta tag	58, 115
cleveref (package)	247	description (environment)	2035
cmap (package)	54	displaymath (environment)	3104
CombineHigherDepths (boolean)	47, 118	\dotfill	3986
comment (package)	76	draftwatermark (package)	330
Computer Modern	53		
\ConTeXt	4149		
counters:		E	
FileDepth	46, 118	ellipsis (package)	330
lofdepth	212	\emph	3959
lotdepth	212	emptypage (package)	331
LWR@externalfilecnt	217	\end@dlbfloat	2815
LWR@htmlfilenumber	97	\end@float	2815
LWR@latestautopage	204	\enlargethispage	4087
LWR@lateximagedepth	230	\enskip	260, 4051
LWR@lateximagenumber	230	enumerate (environment)	2004
LWR@midrulecounter	179	enumerate (package)	68, 332
LWR@minipagedepth	249	enumitem (package)	68, 84
LWR@nextautofloat	204	environ (package)	85
LWR@nextautopage	204	environments:	
LWR@nextequation	219	abstract	1739
LWR@nextpage	230	align	3311
LWR@prevFileDepth	124	align*	3333
LWR@spandepth	107	BlockClass	712
LWR@tablecolspos	169	BVerbatim	1951
LWR@tablecolwidth	169	center	3529
LWR@tabletotalcols	169	description	2035
LWR@tabletotalcolsnext	169	displaymath	3104
LWR@thisfloat	203	enumerate	2004
SideTOCDepth	46, 210	equation	3171
tocdepth	46	flalign	3355
\cpagerefFor	3801	flalign*	3377
\cref@currentlabel	117	flushleft	3535
cross-references		flushright	3532
missing or incorrect	70	gather	3267
CSS		gather*	3289
file selection	59	itemize	1986
lwrap.css	59	lateximage	3451
per HTML page	59	longtable	2
project-specific changes	59	LVerbatim	1962
		LWR@nestspan	607
		LWR@tabular	2575
D		math	3103
danger icon	74	minipage	3877
\date	58	multline	3220
dcolumn (package)	329	multline*	3243
debugging	70	picture	248, 3819
\DeclareGraphicsExtensions	3550	quote	1763
\DeclareGraphicsRule	3550	theindex	3030

tikzpicture	8	Flare (program)	29
titlepage	57, 1448	float (package)	337
titlingpage	57, 1451, 1464	\floatname	11
Verbatim	1819	\floatplacement	14
verbatim	1778	floatrow (package)	339
VerbatimClass	1811	\floatstyle	17
verse	2	flushleft (environment)	3535
warpall	34, 55	flushright (environment)	3532
warpHTML	42, 49, 55	font	
warpprint	35, 49, 55	bitmapped	53
epigraph (package)	333	Computer Modern	53
equation (environment)	3171	Deja Vu	53
error messages	70	ligatures	53, 54
eso-pic (package)	334	packages	53
etoolbox (package)	75	selection	53
everyhook (package)	83	size — lateximage	60, 215, 230
everypage (package)	335	size — math, SVG	60, 215, 230
expl3 (package)	83	size — xfrac	432
extramarks (package)	335	fontenc (package)	53, 345
		fontspec (package)	53, 345
F		\footnote	929
fancyhdr (package)	336	footnotes	115
fancyvrb (package)	83	for	
FAQ	70	HTML & PRINT	74
fbox		HTML output	74
around a minipage	57	PRINT output	74
\fbox	3949	\framebox	3946
file		framed (package)	346
inaccessible	46, 47	FrameMaker (program)	29
underscore	46, 56	Frequently Asked Questions	70
FileDepth (counter)	46, 118	\frontmatter	1095
files:		ftnright (package)	349
glyphtounicode	53	\fup	3983
lwarp.css	59, 274		
lwarp_formal.css	301	G	
lwarp_html.xdy	306	gather (environment)	3267
lwarp_mathjax.txt	307	gather* (environment)	3289
lwarp_sagebrush.css	297	GELLMU (program)	27
lwarp_tutorial.txt	37	generator	
lwarpmk	309	HTML meta tag	131
lwarpmk.conf	269	geometry (package)	82, 349
project.css	59	getttitlestring (package)	83
project.lwarpmkconf	273	GladTeX (program)	28
project_html.tex	273	glyphtounicode (file)	53
sample_project.css	59, 306	graphics (package)	236, 350
tutorial.tex	37	\graphicspath	3548
FileSectionNames (boolean)	47, 97	graphicx (package)	236, 350
\flagverse	425		
flalign (environment)	3355	H	
flalign* (environment)	3377	Hevea (program)	27

<code>\hfill</code>	3984		
<code>\HomeHTMLFileName</code>	500		
<code>\HomeHTMLFilename</code>	46		
horizontal space			
between minipages	260		
<code>\href</code>	2758		
<code>\hrulefill</code>	3985		
<code>\hspace</code>	260, 4080		
HTML			
commands for a successful conver-			
sion	56		
conversion settings	46		
filename generation	47		
headings	73		
meta tag			
description	58, 115		
generator	131		
viewport	131		
tabular column conversion	174		
<code>\HTMLentity</code>	488		
<code>\HTMLFileName</code>	46, 499		
<code>HTMLleftmargini</code> (length)			
.....	66, 152, 153, 424		
<code>\HTMLunicode</code>	489		
<code>HTMLvleftskip</code> (length) ..	66, 152, 424		
<code>\hyperindexref</code>	3047		
<code>\hyperref</code> (package)	198, 351		
<code>\hypertoc</code>	2977		
<code>\hypertocfloat</code>	2990		
		I	
icon			
warning	74		
<code>ifplatform</code> (package)	76		
images			
graphicx package	236		
in strange places	71		
<code>\includegraphics</code>	3730		
<code>indentfirst</code> (package)	354		
<code>InDesign</code> (program)	29		
<code>\InlineClass</code>	732		
<code>inputenc</code> (package)	53, 354		
<code>internet</code> (class)	27		
item, list, empty	56		
<code>itemize</code> (environment)	1986		
<code>\itshape</code>	3975		
		J	
JavaScript			
MathJax	59, 215		
		K	
<code>keyfloat</code> (package)	355		
<code>kvoptions</code> (package)	76		
		L	
label			
in HTML	134		
math environment	224		
language HTML metadata	130		
<code>\LaTeX</code>	4135		
<code>LaTeX2HTML</code> (program)	27		
<code>\LaTeXe</code>	4135		
<code>lateximage</code> (environment)	3451		
<code>\LateximageFontSizeName</code>	3415		
lateximages			
font size	60, 215, 230		
<code>LaTeXML</code> (program)	27		
<code>layout</code> (package)	356		
lengths:			
<code>HTMLleftmargini</code> ..	66, 152, 153, 424		
<code>HTMLvleftskip</code>	66, 152, 424		
<code>LWR@minipageheight</code>	249		
<code>LWR@minipagewidth</code>	249		
<code>LWR@tempheight</code>	261		
<code>LWR@tempraise</code>	261		
<code>LWR@tempwidth</code>	261		
<code>\VerbatimHTMLWidth</code>	153		
<code>vleftmargini</code>	66, 152, 424		
<code>vleftskip</code>	66, 152, 424, 425		
<code>letterspace</code> (package)	356		
<code>LibreOffice</code> (program)	28		
ligatures	53, 54, 82		
line numbers	74		
<code>\linebreak</code>	4081		
<code>\LinkHome</code>	550		
<code>lips</code> (package)	357		
list item, empty	56		
<code>listings</code> (package)	358		
<code>\listof</code>	2944		
<code>lmodern</code> (package)	53		
<code>lofdepth</code> (counter)	212		
<code>longtable</code> (environment)	2		
<code>longtable</code> (package)	362		
<code>lotdepth</code> (counter)	212		
<code>lscap</code> (package)	364		
<code>ltxcaption</code> (package)	364		
LuaLaTeX			
detection	81		
file & section names	123		
<code>\LuaLaTeX</code>	4143		

LuaLaTeX (program) [requirement] ..	31	\LWR@docmidrule	2351
\LuaTeX	4143	\LWR@doingapar (boolean)	107
LVerbatim (environment)	1962	\LWR@doinghline (boolean)	165
lwarp		\LWR@doingstartpars (boolean) ...	107
loading	54	\LWR@doingtbrule (boolean)	165
options	54	\LWR@domulticolumn	2424
[lwarp]:		\LWR@doubledollar	3073
BaseJobname (option)	77	\LWR@endhideamsmath	3213
mathjax (option)	54, 77	\LWR@endminipage	3839
mathsvg (option)	54, 77	\LWR@endoffline	4029
warpHTML (option)	54, 77	\LWR@endsubminipage	3862
warpprint (option)	54, 76	\LWR@ensuredoingapar	773
lwarp (package)	54	\LWR@externalfilecnt (counter) ..	217
[lwarp-newproject]:		\LWR@filenamenoblanks	966
BaseJobName (option)	271	\LWR@filestart	1268
lwarpmk (option)	271	\LWR@findword	251
warpHTML (option)	271	\LWR@floatbegin	2795
warpprint (option)	271	\LWR@floatend	2808
lwarp-newproject (package)	271	\LWR@freezethisfloat (boolean) ..	203
lwarp.css (file)	59, 274	\LWR@futurenonspacelct	2063
lwarp_formal.css (file)	301	\LWR@getexparray	484
lwarp_html.xdy (file)	306	\LWR@getmynexttoken	2070
lwarp_mathjax.txt (file)	307	\LWR@hidelatexequation	3146
lwarp_sagebrush.css (file)	297	\LWR@hspace	4059
lwarp_tutorial.txt (file)	37	\LWR@htmlblockcomment	651
lwarpmk (file)	309	\LWR@htmlblocktag	659
lwarpmk (option) [lwarp-newproject]	271	\LWR@htmlblocktagcomment	664
lwarpmk.conf (file)	269	\LWR@htmlclosecomment	638
\LWR@addmathjax	3160	\LWR@htmlcomment	647
\LWR@afterendverbatim	1803	\LWR@htmldivclass	689
\LWR@amsmultline (boolean)	223	\LWR@htmldivclassend	692
\LWR@atbeginverbatim	1795	\LWR@htmllelement	702
\LWR@beginhideamsmath	3205	\LWR@htmllelementclass	678
\LWR@botnavigation	558	\LWR@htmllelementclassend	683
\LWR@caption@begin	2837	\LWR@htmllelementclassline	695
\LWR@caption@end	2847	\LWR@htmllelementend	705
\LWR@clearmidrules	2332	\LWR@htmlfilenumber (counter)	97
\LWR@closeparagraph	799	\LWR@htmlfileref	2643
\LWR@closeprevious	578	\LWR@htmlmathlabel	3191
\LWR@closetabledatacell	2084	\LWR@htmlopencomment	638
\LWR@colafterspec	169	\LWR@htmlrefsectionfilename	543
\LWR@colatspec	169	\LWR@HTMLsanitize	3418
\LWR@colbangspec	169	\LWR@HTMLsanitizeexpand	3436
\LWR@colbeforespec	169	\LWR@htmlsectionfilename	510
\LWR@commentfooterstyle	230	\LWR@htmlspan	613
\LWR@copyfile	315	\LWR@htmlspanclass	621
\LWR@createautosec	1104	\LWR@htmltag	629
\LWR@currentcss	900	\LWR@htmltagc	600
\LWR@currentHTMLdescription	915	\LWR@includegraphicsb	3641
\LWR@descitem	2023	\LWR@indexitem	3035, 3039, 3043

\LWR@indexsection	3019	\LWR@parseatcolumn	2142
\LWR@infloatrow (boolean)	240	\LWR@parsebangcolumn	2164
\LWR@intabularmetadata (boolean)	166	\LWR@parsebeforecolumn	2185
\LWR@itemizeitem	1979	\LWR@parsedcolumn	2221
\LWR@latestautopage (counter)	204	\LWR@parsenormalcolumn	2207
\LWR@lateximagedepth (counter)	230	\LWR@parsepcolumn	2217
\LWR@lateximagedepthref	2647	\LWR@parseskipcolumn	2204
\LWR@lateximagenumber (counter)	230	\LWR@parsetablecols	2226
\LWR@lateximagenumberref	2650	\LWR@prevFileDepth (counter)	124
\LWR@loadafter	70	\LWR@printmccoldata	2385
\LWR@loadbefore	81	\LWR@printmccoltype	2355
\LWR@loadnever	86	\LWR@printpendingnotes	940
\LWR@longtabledatacaptiontag	2445	\LWR@printthetitle	1540
\LWR@lookforpackagename	254	\LWR@ProvidesPackageDrop	305
\LWR@LwarpEnd	1388, 4179	\LWR@ProvidesPackagePass	293
\LWR@LwarpStart	1322, 4179	\LWR@pushoneclose	1107
\LWR@makefnmark	3838	\LWR@requesttoc	1381
\LWR@makefnmtxt	3831	\LWR@requirepackagenames	250
\LWR@maketitlesetup	1689	\LWR@restoremathlatexformatting	3128
\LWR@maybeinthisfloat	2820	\LWR@rotboxorigin	3741
\LWR@maybenewtablerow	2281	\LWR@rotstyle	3631
\LWR@midrulecounter (counter)	179	\LWR@rule	4088
\LWR@midrules	179	\LWR@scalestyle	3634
\LWR@minipagedepth (counter)	249	\LWR@section	1114
\LWR@minipagefullwidth (boolean)	251	\LWR@sectionnumber	1101
\LWR@minipageheight (length)	249	\LWR@setexparray	481
\LWR@minipagestartpars	4033	\LWR@setlatestname	2619
\LWR@minipagestoppars	4039	\LWR@sidetoc	2957
\LWR@minipagethispar (boolean)	252	\LWR@singledollar	3085
\LWR@minipagewidth (length)	249	\LWR@skippingmrowcell (boolean)	166
\LWR@multicolother	2376	\LWR@spandepth (counter)	107
\LWR@multicolpartext	2371	\LWR@splabel	2653
\LWR@multicolskip	2382	\LWR@starredlongtable (boolean)	183
\LWR@mynexttoken	2062	\LWR@startedrow (boolean)	165
\LWR@myshorttoc	2902	\LWR@startnewdepth	1108
\LWR@nameref	2640	\LWR@startpars	837
\LWR@nestspan (environment)	607	\LWR@startref	2689
\LWR@newhtmlfile	1044	\LWR@stoppars	849
\LWR@newlabel	2677	\LWR@stripperiod	2618
\LWR@newref	2725	\LWR@strresult	2136
\LWR@nextautofloat (counter)	204	\LWR@subcmidrule	2343
\LWR@nextautopage (counter)	204	\LWR@subhtml elementclass	669
\LWR@nextequation (counter)	219	\LWR@subhyperref	2749
\LWR@nextpage (counter)	230	\LWR@subhyperrefclass	2753
\LWR@nullrule	3830	\LWR@subinlineimage	2788
\LWR@openparagraph	778	\LWR@subminipage	3853
\LWR@origcolspec	2137	\LWR@subnewref	2719
\LWR@originname	3742	\LWR@subsublabel	2656
\LWR@originnames	3749	\LWR@subtableofcontents	2917
\LWR@parseaftercolumn	2194	\LWR@syncmathjax	3107

\LWR@tablecolspec	2135	MathJax	
LWR@tablecolspos (counter)	169	mathjax option	77
LWR@tablecolswidth (counter)	169	subequations	61, 216
\LWR@tabledatacolumnntag	2529	summary	60, 216
\LWR@tabledatamulticolumnntag	2509	tagged equations	61, 216
\LWR@tabledatamultirowtag	2515	MathJax (program)	60, 216
\LWR@tabledatasinglecolumnntag	2297	MathJax (program) [requirement]	31
LWR@tableparcell (boolean)	165	mathjax (boolean)	76
LWR@tabletotalcols (counter)	169	mathjax (option) [lwrap]	54, 77
LWR@tabletotalcolsnext (counter)	169	mathsvg (option) [lwrap]	54, 77
LWR@tabular (environment)	2575	mcaption (package)	365
\LWR@tabularendoffline	2119	mdframed (package)	366
LWR@temphheight (length)	261	\mdseries	3969
LWR@tempraise (length)	261	memoir	
LWR@tempwidth (length)	261	verse	425
\LWR@thisfilename	964	memoir (package)	425
LWR@thisfloat (counter)	203	meta tag, HTML	
\LWR@thisnewfilename	965	description	58, 115
\LWR@titlingmaketitle	1726	generator	131
\LWR@topnavigation	555	viewport	131
\LWR@traceinfo	334	\MetaLanguage	48, 1267
LWR@tracinglwrap (boolean)	89	microtype (package)	54, 81, 373
LWR@validtablecol (boolean)	173	\MiKTeX	4159
\LWR@Verbatimclass	1810	minipage	
LWR@verbtags (boolean)	153	horizontal space between	260
\LWRsetnextfloat	2828	in an \fbox	57
\LyX	4160	minipage (environment)	3877
		\minipagefullwidth	3869
M		missing sections	46, 47
Madcap (program)	29	\mrowcell	2571
\mainmatter	1091	MS-Windows (program)	54, 90
\makebox	3943	multicol (package)	374
makeidx (package)	83	multirow (package)	186, 376
\MakeIndex	4152	multiline (environment)	3220
\maketitle	57, 145, 1701	multiline* (environment)	3243
\maketitlehookaa	137	N	
\maketitlehookaaa	137	\Nameref	2746
margin		\nameref	2737
numbers	74	nameref (package)	376
tags	74	needspace (package)	377
\marginpar	953	newclude (package)	67, 377
markup languages	29	\NewCSS	45, 48, 59, 901
math		newfloat	
font size — SVG	60, 215, 230	with trivfloat, algorithmic	422
mathjax option	77	\newfloat	2
MathJax summary	60, 216	newfloat (package)	84
mathsvg option	77	\NewHTMLdescription	49, 58, 916
SVG summary	60, 215	\newline	4027
math (environment)	3103	\newlist	2053

<code>\newtheorem</code>	159	<code>cleveref</code>	247
<code>newtxmath</code> (package)	67	<code>cmap</code>	54
<code>newunicodechar</code> (package)	53, 378	<code>comment</code>	76
<code>nextpage</code> (package)	378	<code>dcolumn</code>	329
<code>\nolinebreak</code>	4082	<code>draftwatermark</code>	330
<code>\nolinkurl</code>	2770	<code>ellipsis</code>	330
<code>\nopagebreak</code>	4086	<code>emptypage</code>	331
<code>\notedivision</code>	926	<code>enumerate</code>	68, 332
<code>\noteentry</code>	932	<code>enumitem</code>	68, 84
<code>\notenuminnotes</code>	930	<code>environ</code>	85
<code>nowidow</code> (package)	379	<code>epigraph</code>	333
<code>ntheorem</code> (package)	380	<code>eso-pic</code>	334
<code>\numberline</code>	2974	<code>etoolbox</code>	75
numbers		<code>everyhook</code>	83
left margin	74	<code>everypage</code>	335
O			
OpenOffice (program)	28	<code>expl3</code>	83
options:		<code>extramarks</code>	335
[lwarp-newproject]:		<code>fancyhdr</code>	336
BaseJobName	271	<code>fancyvrb</code>	83
lwarpmk	271	<code>float</code>	337
warpHTML	271	<code>floatrow</code>	339
warpprint	271	<code>fontenc</code>	53, 345
[lwarp]:		<code>fontspec</code>	53, 345
BaseJobname	77	<code>framed</code>	346
mathjax	54, 77	<code>ftnright</code>	349
mathsvg	54, 77	<code>geometry</code>	82, 349
warpHTML	54, 77	<code>getttitlestring</code>	83
warpprint	54, 76	<code>graphics</code>	236, 350
<code>\OSPathSymbol</code>	381	<code>graphicx</code>	236, 350
P			
packages		<code>hyperref</code>	198, 351
required	81	<code>ifplatform</code>	76
packages:		<code>indentfirst</code>	354
abstract	319	<code>inputenc</code>	53, 354
afterpage	321	<code>keyfloat</code>	355
algorithmicx	322	<code>kvoptions</code>	76
alltt	323	<code>layout</code>	356
amsmath	84	<code>letterspace</code>	356
amsthm	324	<code>lips</code>	357
array	191	<code>listings</code>	358
babel	67	<code>lmodern</code>	53
bookmark	326	<code>longtable</code>	362
booktabs	328	<code>lscape</code>	364
calc	83	<code>ltcaption</code>	364
caption	84	<code>lwarp</code>	54
ccaption	328	<code>lwarp-newproject</code>	271
changepage	329	<code>makeidx</code>	83
		<code>mcaption</code>	365
		<code>mdframed</code>	366
		<code>memoir</code>	425
		<code>microtype</code>	54, 81, 373

multicol	374	\pagerefPageFor	2731
multirow	186, 376	\pagestyle	239
nameref	376	Pandoc (program)	28
needspace	377	\paragraph	1251
newclude	67, 377	\parbox	3936
newfloat	84	\parsemulticolumnalignment	2412
newtxmath	67	parskip (package)	392
newunicodechar	53, 378	\part	1205
nextpage	378	pdfcrop (program) [requirement]	31
nowidow	379	pdfLaTeX (program) [requirement]	31
ntheorem	380	pdfseparate (program) [requirement]	31, 35
pagenote	84, 391	pdftocairo (program) [requirement]	31, 35
parskip	392	pdftotext (program) [requirement]	31, 35
placeins	392	Perl	36
ragged2e	393	perl (program) [requirement]	36
rotating	394	\phantomsection	4129
setspace	395	picture (environment)	248, 3819
showidx	396	placeins (package)	392
showkeys	396	Plastex (program)	27
sidecap	397	\popclose	454
siunitx	67, 235	Poppler	31, 35
subfig	398	\postpublished	137
tabularx	404	\postsubtitle	137
textcomp	54	\prepublished	137
textpos	405	\presubtitle	137, 147
theorem	406	\printauthor	138, 1557, 1572
threeparttable	410	\printdate	138, 1562, 1574
tikz	411	\printindex	3022
titlesps	85, 412	\printpublished	138, 1523, 1569
titlesec	414	\PrintStack	563
titletoc	415	\prints subtitle	138, 1548, 1571
titling	84, 416	\printthanks	137, 141
tocloft	417	\printtitle	138, 1532, 1570
trivfloat	422	problems	70
verse	424, 425	programs	
wallpaper	426	utility	30
wrapfig	427	programs:	
xcolor	429	[requirement]:	
xfrac	85, 432	LuaLaTeX	31
xifthen	83	MathJax	31
xparse	82	pdfcrop	31
xstring	83	pdfLaTeX	31
zref	84	pdfseparate	31, 35
page		pdftocairo	31, 35
inaccessible	46, 47	pdftotext	31, 35
\pagebreak	4083	perl	36
pagenote (package)	84, 391	XeLaTeX	31
\pagenotesubhead	928	Adobe	29
\pagenumbering	243		
\pageref	2732		

SVG		
math summary	60, 215	
mathsvg option	77	
T		
<code>\tableofcontents</code>	48	
tabular		
baseline	174	
HTML column conversion	174	
tabularx (package)	404	
tagged equations		
Mathjax	61, 216	
<code>\TeX</code>	4132	
TeX2page (program)	27	
TeX4ht (program)	27	
<code>\textbf</code>	3961	
textcomp (package)	54	
<code>\textgreater</code>	491	
<code>\textit</code>	3966	
<code>\textless</code>	491	
<code>\textmd</code>	3960	
<code>\textnormal</code>	3968	
textpos (package)	405	
<code>\textrm</code>	3962	
<code>\textsc</code>	3967	
<code>\textsf</code>	3963	
<code>\textsubscript</code>	3981	
<code>\textsuperscript</code>	3980	
<code>\texttt</code>	3964	
<code>\textup</code>	3965	
<code>\tfl@chapter@fix</code>	422	
<code>\thanks</code>	58	
<code>\thanksmarkseries</code>	1730	
<code>\theauthor</code>	138	
<code>\thedata</code>	138	
theindex (environment)	3030	
theorem (package)	406	
<code>\thepublished</code>	138	
<code>\thesubtitle</code>	138	
<code>\thetitle</code>	138	
<code>\thispagestyle</code>	241	
threeparttable (package)	410	
tikz		
catcodes	411	
dollar redefined	411	
tikz (package)	411	
tikzpicture (environment)	8	
<code>\title</code>	58	
titlepage (environment)	57, 1448	
titlesp (package)	85, 412	
titlesec (package)	414	
titletoc (package)	415	
titling (package)	84, 416	
titlingpage (environment)	57, 1451, 1464	
tocdepth (counter)	46	
tocloft (package)	417	
<code>\tracinglwrap</code>	71, 333	
trivfloat		
with newfloat, algorithmic	422	
trivfloat (package)	422	
troubleshooting	70	
<code>\ttfamily</code>	3973	
TtH (program)	27	
tutorial.tex (file)	37	
U		
underscore		
filename	46, 56	
Unicode		
enhanced coverage	53	
file & section names	123	
input characters	75	
selection	53	
<code>\up</code>	3982	
<code>\upshape</code>	3974	
<code>\url</code>	2777	
<code>\UseLatexmk</code>	46	
<code>\UseVerbatim</code>	1963	
usingOSwindows (boolean)	91	
UTF-8		
enhanced coverage	53	
file & section names	123	
selection	53	
utility		
programs	30	
V		
Verbatim (environment)	1819	
verbatim (environment)	1778	
VerbatimClass (environment)	1811	
<code>\VerbatimHTMLWidth</code> (length)	153	
verse (environment)	2	
verse (package)	424, 425	
viewport		
HTML meta tag	131	
vleftmargini (length)	66, 152, 424	
vleftskip (length)	66, 152, 424, 425	
W		
wallpaper (package)	426	

warning icon	74		
warpall (environment)	34, 55		
warpHTML (environment)	42, 49, 55		
warpHTML (option) [lwrap-newproject]	271		
warpHTML (option) [lwrap]	54, 77		
\warpHTMLonly	33, 49, 55		
warpingHTML (boolean)	76		
warpingprint (boolean)	76		
\warpOSwindows	54, 90, 382		
warpprint (environment)	35, 49, 55		
warpprint (option) [lwrap-newproject]	271		
warpprint (option) [lwrap]	54, 76		
\warpprintonly	32, 49, 55		
Windows (program)	54, 90		
Word (program)	28		
wrapfig (package)	427		
		X	
xcolor (package)	429		
XeLaTeX			
detection	81		
file & section names	123		
\XeLaTeX	4145		
XeLaTeX (program) [requirement]	31		
\XeTeX	4145		
xfrac (package)	85, 432		
xifthen (package)	83		
xparse (package)	82		
xstring (package)	83		
		Z	
zref (package)	84		