



The **lwarp** package

v0.21 — 2017/02/23

L^AT_EX to HTML5

© 2016 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	15
1	Updates	15
2	Introduction	18
2.1	Supported packages and features	19
3	Alternatives	24
3.1	Internet class	24
3.2	TeX4ht	24
3.3	Translators	24
3.4	AsciiDoc	25
3.5	Pandoc	25
3.6	Word Processors	25
3.7	Commercial Systems	26
3.8	Comparisons	26
4	Installation	27
4.1	Installing the lwarp package	27
4.2	Installing the lwarpmk utility	30
4.2.1	Using a local copy of lwarpmk	31
4.3	Installing additional utilities	32
5	Tutorial	34
5.1	Starting a new project	34
5.2	Compiling the print version with lwarpmk	38
5.3	Compiling the HTML version with lwarpmk	39
5.4	Generating the SVG images	40
5.5	Using MathJax for math	41
5.6	Changing the CSS style	42
5.7	Customizing the HTML output	43
5.8	Using latexmk	47
5.9	Using XeLaTeX or LuaLaTeX	48
5.10	Cleaning auxiliary files	49
5.11	Cleaning auxiliary and output files	49
5.12	Processing multiple projects in the same directory	49
5.13	Using the make utility	49
6	Additional details	50
6.1	Font and UTF-8 support	50
6.2	lwarp package loading and options	51
6.3	Selecting the operating system	51
6.4	Selecting actions for print or HTML output	51
6.5	Commands to be placed into the warpprint environment	52

6.6	Commands for a successful HTML conversion	53
6.7	Title page	54
6.8	HTML page meta descriptions	55
6.9	CSS	55
6.10	Special cases and limitations	56
6.10.1	Text formatting	56
6.10.2	Verse and Memoir packages	56
6.10.3	Math	57
6.10.4	Graphics	59
6.10.5	Xcolor	59
6.10.6	Tikz	59
6.10.7	Tabular	60
6.10.8	Minipages	61
6.10.9	Mdframed	61
6.10.10	float, trivfloat, and/or algorithmicx together	62
6.10.11	caption and subcaption packages	62
6.10.12	floatrow package	62
6.10.13	siunitx package	62
6.10.14	newclude package	63
6.10.15	newtxmath package	63
6.10.16	babel package	63
6.10.17	enumitem package	63
7	Expanding the code base	63
7.1	Creating an lwarp version of a package	64
8	Troubleshooting	64
8.1	Using the <code>lwarp.sty</code> package	64
8.1.1	Debug tracing output	66
8.2	Compiling the <code>lwarp.dtx</code> file	66
9	Implementation	67
10	Stack depths	68
11	Source Code	69
12	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	70
13	Unicode Input Characters	70
14	Handling package options	70
15	Misplaced packages	73
16	Required packages	75

17	Loading Packages	80
18	Copying a file	82
19	Debugging messages	83
20	Remembering original formatting macros	83
21	Operating-System portability	84
21.1	Common portability code	85
21.2	Unix, Linux, and Mac OS	85
21.3	MS-Windows	85
22	Stacks	86
22.1	Assigning depths	86
22.2	Closing actions	87
22.3	Closing depths	87
22.4	Pushing and popping the stack	88
23	Data arrays	89
24	HTML entities	90
25	HTML filename generation	90
26	Homepage link	93
27	\PrintStack diagnostic tool	93
28	Closing stack levels	94
29	HTML tags, spans, divs, elements	94
29.1	Mapping L ^A T _E X Sections to HTML Sections	94
29.2	HTML tags	95
29.3	Block tags and comments	96
29.4	Div class and element class	98
29.5	Single-line elements	99
29.6	HTML5 semantic elements	99
29.7	High-level block and inline classes	99
29.8	Closing HTML tags	100
30	Paragraph handling	101
31	Paragraph start/stop handling	105
32	Page headers and footers	107
33	CSS	108

34	HTML meta description	109
35	Footnotes	109
35.1	Regular page footnotes	109
35.2	Minipage footnotes	110
35.3	Titlepage thanks	110
35.4	Regular page footnote implementation	110
36	Marginpars	112
37	Splitting HTML files	112
38	Sectioning	117
38.1	Book class commands	117
38.2	Sectioning support macros	117
38.3	\section and friends	122
39	Starting a new file	124
40	Starting HTML output	126
41	Ending HTML output	129
42	Titles and the titling package	131
42.1	Setting the title, etc.	132
42.2	Changes to HTML titlepage and titlingpage	134
42.3	Printing the title, etc. in HTML	135
42.4	Printing the title, etc. in print form	138
42.5	\maketitle for print output	139
42.6	\maketitle for HTML output	141
43	Abstract	144
44	Quote and verse	145
44.1	Citations and attributions	145
44.2	Quotes, quotations	145
44.3	Verse	146
45	Verbatim	148
46	Fancyvrb	148
47	Theorems	154
48	Lists	155
48.1	Itemize	155
48.2	Enumerate	156
48.3	Description	157

49	Tabular	158
49.1	Token lookahead	159
49.2	Booleans	160
49.3	Handling &	161
49.4	Handling \\	163
49.5	Variables	163
49.6	Parsing @, >, <, ! columns	165
49.7	Parsing 'l', 'c', or 'r' columns	166
49.8	Parsing 'p', 'm', or 'b' columns	167
49.9	Parsing the column specifications	167
49.10	Starting a new row	170
49.11	Data opening tag	171
49.12	Midrules	172
49.13	Multicolumns	173
49.13.1	Parsing multicolumns	173
49.13.2	High-level multicolumn interface	175
49.13.3	Longtable captions	176
49.13.4	\tabledatamulticolumntag	178
49.14	Multirow	179
49.15	Utility macros inside a table	179
49.16	Checking for a new table cell	179
49.17	\mrowcell	181
49.18	New \tabular definition	181
49.19	Array	183
50	Cross-references	184
50.1	Setup	184
50.2	Zref setup	186
50.3	Labels	187
50.4	References	189
50.5	Hyper-references	191
51	Floats	193
51.1	Float captions	194
51.1.1	Caption inside a float environment	195
51.1.2	Caption and LOF linking and tracking	195
52	Table of Contents, LOF, LOT	198
52.1	Reading and printing the TOC	199
52.2	High-level TOC commands	201
52.3	Side TOC	201
52.4	Low-level TOC line formatting	203
53	Index	204
54	Math	206

54.1	Inline and display math	209
54.2	MathJax support	210
54.3	Equation environment	213
54.4	AMS Math environments	215
54.4.1	Support macros	215
54.4.2	Environment patches	216
55	Lateximages	220
56	center, flushleft, flushright	226
57	Siunitx	227
58	Graphics	227
58.1	\graphicspath	228
58.2	Length conversions and graphics options	228
58.3	\includegraphics	231
58.4	\rotatebox, \scalebox, \reflectbox	235
58.5	Null functions	237
59	Cleverref	238
60	Picture	239
61	Tikzpicture	240
62	Boxes and Minipages	241
62.1	Counters and lengths	241
62.2	Footnote handling	242
62.3	Minipage handling	243
62.4	Parbox, makebox, framebox, fbox, raisebox	246
63	Direct formatting	247
64	Skips, spaces, font sizes	250
65	\phantomsection	257
66	\LaTeX and other logos	257
66.1	HTML logos	257
66.2	Print logos	259
67	\AtBeginDocument, \AtEndDocument	260
68	lwarpmk.conf	261
69	Trademarks	262

2	lwarp-newproject.sty	263
70	lwarp-newproject	263
70.1	project_html.tex	265
70.2	project.lwarpmkconf	265
70.3	lwarp.css	266
70.4	lwarp_sagebrush.css	286
70.5	lwarp_formal.css	291
70.6	sample_project.css	295
70.7	lwarp_html.xdy	296
70.8	lwarp_mathjax.txt	296
70.9	Lwarpmk option	298
3	lwarp-afterpage.sty	307
71	Afterpage	307
4	lwarp-algorithmicx.sty	308
72	Algorithmicx	308
5	lwarp-alltt.sty	308
73	Alltt	309
6	lwarp-bookmark.sty	310
74	Bookmark	310
7	lwarp-booktabs.sty	311
75	Booktabs	311
8	lwarp-draftwatermark.sty	311
76	Draftwatermark	311
9	lwarp-epigraph.sty	311
77	Epigraph	312

10	lwarp-eso-pic.sty	313
78	Eso-pic	313
11	lwarp-everypage.sty	313
79	Everypage	314
12	lwarp-extramarks.sty	314
80	Extramarks	314
13	lwarp-fancyhdr.sty	314
81	Fancyhdr	315
14	lwarp-float.sty	316
82	Float and \newfloat	316
15	lwarp-floatrow.sty	318
83	Floatrow	318
16	lwarp-fontenc.sty	323
84	Fontenc	323
17	lwarp-fontspec.sty	323
85	Fontspec	323
18	lwarp-geometry.sty	324
86	Geometry	324
19	lwarp-graphics.sty	325
87	Graphics	325

20	lwarp-graphicx.sty	325
88	Graphicx	325
21	lwarp-hyperref.sty	326
89	Hyperref	326
22	lwarp-inputenc.sty	329
90	Inputenc	329
23	lwarp-keyfloat.sty	330
91	Keyfloat	330
24	lwarp-letterspace.sty	331
92	Letterspace	331
25	lwarp-listings.sty	332
93	Listings	332
26	lwarp-longtable.sty	336
94	Longtable	336
27	lwarp-ltcaption.sty	338
95	Ltcaption	338
28	lwarp-mdframed.sty	339
96	Mdframed	339
29	lwarp-microtype.sty	346
97	Microtype	346

30	lwarp-multicol.sty	347
98	Multicol	347
31	lwarp-multirow.sty	349
99	Multirow	349
32	lwarp-needspace.sty	349
100	Needspace	349
33	lwarp-newunicodechar.sty	350
101	Newunicodechar	350
34	lwarp-nowidow.sty	350
102	Nowidow	350
35	lwarp-pagenote.sty	350
103	Pagenote	351
36	lwarp-placeins.sty	352
104	Placeins	352
37	lwarp-ragged2e.sty	353
105	Ragged2e	353
38	lwarp-rotating.sty	354
106	Rotating	354
39	lwarp-setspace.sty	354
107	Setspace	354

40	lwarp-textpos.sty	355
108	Textpos	355
41	lwarp-threeparttable.sty	357
109	Threeparttable	357
42	lwarp-titleps.sty	358
110	Titleps	358
43	lwarp-titlesec.sty	360
111	Titlesec	360
44	lwarp-titletoc.sty	361
112	Titletoc	361
45	lwarp-titling.sty	362
113	Titling	362
46	lwarp-tocloft.sty	363
114	Tocloft	363
47	lwarp-trivfloat.sty	368
115	Trivfloat	368
115.1	Combining \newfloat, \trivfloat, and algorithmicx	368
48	lwarp-wallpaper.sty	369
116	Wallpaper	369

49	lwarp-wrapfig.sty	370
117	Wrapfig	370
50	lwarp-xcolor.sty	372
118	Xcolor	372
51	lwarp-xfrac.sty	375
119	Xfrac	375
	Change History and Index	378

List of Figures

1	tutorial.tex listing	35
---	--------------------------------	----

List of Tables

1	L ^A T _E X–HTML generation — lwarp package — Supported functions	20
2	Required software programs	28
3	Files created in a new project	37
4	Section depths and HTML headings	68
5	Tabular baseline	167
6	Tabular HTML column conversions	168
7	Cross-referencing data structures	185
8	Float data structures	193

Package 1

lwarp.sty

1 Updates

For a detailed list of changes, see the Change History on page 378.

v0.21:

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

v0.20:

- The `makefile` and related infrastructure has been replaced by the `lwarpmk` utility. This provides increased portability, reduced dependencies, and much simpler installation and setup.
- The `lwarp-newproject` package is now used to locally create support files.
- The print and HTML versions of a document may co-exist with their own sets of auxiliary files.
- Package handling is now controlled by a modular system which looks for and loads an `lwarp-<package>` version if available.
- High-level source compatibility is provided for all supported packages, almost totally eliminating the need for `warpprint` and `warpHTML` environments.
- A large number of additional packages are supported.
- A new tutorial is included in the documentation, and many obsolete sections have been removed.

- `\NewHTMLdescription` sets the HTML meta description tag for each file. See section 6.8.
- `\HTMLFileName` may now be empty, allowing filenames without a prefix. Lwarp no longer automatically appends a `-` character. For existing projects, add a `-` to the end of `\HTMLFileName`.
- `\HomeHTMLFileName` and `\HTMLFileName` no longer use escaped underscore characters. Underscores may be used in filenames as-is.
- `lwarp` now tries to auto-detect the operating system, and `\warpOSwindows` is only needed if the auto-detection fails to detect Windows.
- Tabular column types `@`, `>`, and `<` are now supported.
- `BlockClass` and `\InlineClass` add an optional style.
- The `sidebar` and `example` environments have been moved to the test suite, and are no longer included in `lwarp`.

v0.19:

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

v0.18:

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

v0.17:

- `mdframed` package is supported.

v0.16:

- Font and input encoding are now controlled by the user, and `lwarp` is loaded after fonts have been selected.
- Support for \TeX and \LaTeX . 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).² Further, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by ALT tags holding the \LaTeX source for the expression, allowing it to be copy/pasted into other documents.³ Custom \LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside \LaTeX .

The MATHJAX JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files, 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`

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

²Aside — The author, being a programmer, believes in the movement towards “logical punctuation”. (Look it up.)

Aside to the aside — The author also believes that space should be on either side of an em-dash, employing the symbol as a separator rather than something looking like a chemical **bond—as** shown here, but is willing to abide by Chicago style when necessary.

³There 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!

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

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 \LaTeX 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 20.)

2.1 Supported packages and features

Table 1 lists the various \LaTeX packages and macros 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 \LaTeX code. *Ignored* means that the package is not used, but nullified user-level macros and related counters, lengths, etc. are provided for source-level compatibility.

Any package listed in table 1 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.

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

Package or Macro	Status
Engines:	pdfL ^A T _E X, X _Y L ^A T _E X, LuaL ^A T _E X
Classes:	book, report, or article. memoir 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.
<code>titlesec</code>	Ignored.
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>titletoc</code>	Ignored.
<code>tocloft</code>	Ignored, with support for <code>\newlistof</code> and <code>\ctfchapterprecis</code> .
<code>abstract</code> :	Supported
Cross-references:	Emulated, with hyperlinks.
<code>cleverref</code> :	Supported, including ranges and lists, with hyperlinks.
<code>nameref</code> :	Emulated
<code>hyperref</code> :	Emulated. HTML hyperlinks are generated for TOC, LOF, LOT, <code>\nameref</code> , <code>\ref</code> , the <code>cleverref</code> commands, and index entries.
<code>bookmark</code> :	Emulated.
Footnotes:	Supported, emulated as pagenotes per HTML page.
<code>pagenote</code> :	Supported.
Indexing:	<code>texindy</code> is used, with hyperlinks.
Bibliography:	Supported, without hyperlinks so far.

lwarp Supported Functions — continued

Package or Macro	Status
babel:	Supported.
Math:	Supported. Converted to SVG images with HTML ALT tags containing the \LaTeX source for the math expression. MathJax supported as an alternative. \LaTeX environments are supported. User-defined macros are available during conversion, due to native \LaTeX processing.
Floats:	Supported, appear where declared. float is emulated, newfloat is supported, caption and subcaption are supported, capt-of is supported, placeins is ignored, trivfloat is supported, floatrow is emulated, keyfloat is supported, wrapfig is emulated.
tabular:	Emulated. <code>\multirow</code> and <code>\multicolumn</code> are available, but cannot be used at the same time. Nested tables are not supported.
multirow:	Emulated.
longtable:	Emulated. Converted to a tabular. Captions supported. Extra headings and <code>\kill</code> lines must be enclosed in <code>\warpprintonly{}</code>
booktabs:	Emulated. <code>\toprule</code> and <code>\bottomrule</code> form black rules, <code>\midrule</code> forms silver rules, as demonstrated on this table. <code>\cmidrule</code> , demonstrated at this line, does not use width or trim options.
threeparttable:	Emulated
geometry:	Ignored.
titlesp:	Used by lwarp, but emulated for the user.
fancyhdr:	Ignored.
extramarks:	Ignored.
afterpage, draftwatermark, eso-pic, everypage, wallpaper:	Ignored.

lwarp Supported Functions — continued

Package or Macro	Status
graphics, graphicx:	Emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle.
rotating:	Emulated. All objects are displayed unrotated.
Lists:	Supported
enumitem:	Supported, although spacing is still controlled by CSS.
Environments:	Standard L ^A T _E X environments are supported.
theorem:	Supported, with added CSS tags.
epigraph:	Emulated
csquotes:	Supported
verse and memoir's verse-related macros	Supported
picture and tikz:	Converted to an SVG image.
minipage:	Supported with some HTML5-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
fancyvrb:	Supported except for verbatim footnotes.
alltt, listings:	Supported
algorithmicx:	Supported
mdframed:	Supported
multicol:	Emulated, with CSS3. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
siunitx:	Supported except for <code>per-mode=fraction</code> .
xfrac:	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.
<code>placeins</code> , <code>setspace</code> , <code>needspace</code> :	Ignored.
<code>textpos</code> :	Emulated.
<code>nowidow</code> :	Ignored.
<code>microtype</code> , <code>letterspace</code> :	Ignored. <code>microtype</code> is used to disable ligatures.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
<code>xcolor</code> :	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.
CJKutf8:	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.
Ignored:	The L ^A T _E X package is not used, but null functions are provided for source compatibility.

3 Alternatives

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

3.1 Internet class

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

3.2 TeX4ht

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

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

3.3 Translators

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

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

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

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

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

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

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

Prog **TeX2page** **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® FrameMaker®, Adobe® InDesign®, and Madcap Flare™.
Prog	FrameMaker	
Prog	InDesign	
	Flare	
Prog	Madcap	3.8 Comparisons

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

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

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

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

4 Installation

Table 2 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 2: 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 3.

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

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

latexmk: Optionally used by lwarpmk to compile L^AT_EX code. On a MiK_TE_X system, Perl may need to be installed first.

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

Poppler PDF utilities:

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

From Poppler: poppler.freedesktop.org.

For Windows, also see:

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

pdftotext: Used to convert PDF to text.

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

pdftocairo: Used to convert images to SVG.

Perl:

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

perl.org, strawberryperl.com

Automatically downloaded from the internet as required:

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

From: mathjax.org

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 \Rightarrow `pdflatex lwarp.dtx` (several times)

6. Create the .sty files:

Enter \Rightarrow `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 \Rightarrow `mktexlsr` — or —

Enter \Rightarrow `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 2 on page 28.

Perl may require installation as well. See table 2.

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

```
Enter ⇒ luatex -version

Enter ⇒ xindy -version

Enter ⇒ latexmk -version

Enter ⇒ perl -version

Enter ⇒ pdfcrop -version

Enter ⇒ pdftotext -v

Enter ⇒ pdfseparate -version

Enter ⇒ pdftocairo -v
```

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

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

To add the Poppler utilities to a Windows machine:

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

8. Click "Ok" when done.

Prog `perl`

To add Perl to a Windows machine:

1. Download and install a version of Perl, such as Strawberry Perl, to a directory without a space in its name, such as **C:\Strawberry**.
2. Edit the Path as seen above for the Poppler utilities.
3. Enter the **bin** directory of the Perl utility, such as:
`C:\Strawberry\perl\bin`
Be sure to include `\bin`.
4. Click "Ok" when done.

5 Tutorial

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

5.1 Starting a new project

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

File `tutorial.tex`

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

Copy from the documentation PDF:

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

Copy from the `lwarp` documentation directory:

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

Enter \Rightarrow `texdoc -l lwarp_tutorial.txt`

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

File `lwarp_tutorial.txt`

 Note: `.txt` suffix!

 Bad Formatting!

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

3. Compile the project:

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

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

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

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

Figure 1: tutorial.tex listing

Note: There are two pages!

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

\documentclass{book}

\usepackage{iftex}

% --- LOAD FONT SELECTION AND ENCODING BEFORE LOADING LWARP ---
\ifPDFTeX
\usepackage{lmodern}           % pdflatex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\else
\usepackage{fontspec}         % XeLaTeX or LuaLaTeX
\fi

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

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

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

\usepackage{hyperref,cleveref} % LOAD THESE LAST!

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarp Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2}      % Include subsections in the TOC.
\setcounter{secnumdepth}{2}   % Number down to subsections.
\setcounter{FileDepth}{1}     % Split HTML files at sections
\booltrue{CombineHigherDepths} % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1}  % Include subsections in the sideTOC
\SetFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\SetPageTop{\fbox{LOGO}}
\SetPageBottom{Contact Information and Copyright}
\NewCSS{lwarp_sagebrush.css}
```

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

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

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

\printindex

\end{document}
```

Table 3: Files created in a new project

tutorial.pdf:	The PDF output from L ^A T _E X.
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 ^A T _E X files <code>tutorial.aux</code> , <code>tutorial.log</code> . When an HTML version of the document is created, <code>_html</code> versions of the auxiliary files will also be generated.
lwarpmk.conf:	A configuration file for <code>lwarpmk</code> , which is used to automate the compilation of PDF or HTML versions of the document.
tutorial.lwarpmkconf:	Another configuration file used by <code>lwarpmk</code> , which is only useful if you wish to have several projects residing in the same directory.
.css files:	<code>lwarp.css</code> , <code>lwarp_formal.css</code> , <code>lwarp_sagebrush.css</code> These files are standard for <code>lwarp</code> , 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 <code>lwarp</code> defaults.
lwarp_html.xdy:	Used by <code>lwarp</code> while creating an index. This file should not need to be modified by the user.
lwarp_mathjax.txt:	Inserted into the HTML files when <code>MathJax</code> is used to display math. This file should not need to be modified by the user.
comment.cut:	A temporary file used by <code>lwarp</code> 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.⁴

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.

⁴ 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`⁵, `Some-math.html`, and the document's index in `_Index.html`.⁶

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.

⁵ `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`.

⁶ `index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

5.4 Generating the SVG images

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


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


1. Create the SVG images:

```
Enter ⇒ lwarpmk limages
```

```
Enter ⇒ lwarpmk html
```

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

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


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

5.5 Using MathJax for math

math with MathJax Math may also be represented using the MathJax Javascript project.

1. In the tutorial's source code, change
`\usepackage{lwarp}`
to
`\usepackage[mathjax]{lwarp}`
2. Recompile

Enter \Rightarrow `lwarpmk html`
3. Reload the math page.

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

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

5.6 Changing the CSS style

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

The styles provided by `lwarp` include:


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

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

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

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

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

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

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

5.7 Customizing the HTML output

Placement!

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

Changes!

Note that if changes are made, it is best to first use `lwarpmk clearall` to clear all the HTML, PDF, and auxiliary files. Also, if `\HomeHTMLFilename`, `\HTMLFileName`, or `\UseLatexmk` are changed, it is necessary to reenale the `lwarp-newproject` package and then recompile the print version in order to recreate the configuration files for `lwarpmk`.

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

<code>\HomeHTMLFilename</code>	<p>\HomeHTMLFilename: Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code>. A common setting is:</p> <pre>\newcommand{HomeHTMLFilename}{index}</pre> <p>causing the homepage to be the file <code>index.html</code>. Underscores are allowed in filenames.</p>
<code>\HTMLFileName</code>	<p>\HTMLFileName: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.</p>
<code>\UseLatexmk</code>	<p>\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:</p> <pre>\newcommand*{\UseLatexmk}{true}</pre>

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

<code>Ctrl tocdepth</code>	<p>tocdepth: Sectioning depth of the table of contents. See section 10 for a list of L^AT_EX stack depths.</p>
<code>Ctrl SideTOCDepth</code>	<p>SideTOCDepth: Sectioning depth of the sideTOC. Defaults to 1, causing the sidetoc to show sections but not subsections.</p>
<code>sideTOC</code>	<p>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.</p> <p>It is recommended to set:</p> <pre>SideTOCDepth = FileDepth, or SideTOCDepth = FileDepth+1</pre> <p>If <code>SideTOCDepth < FileDepth</code>, web pages will be inaccessible via the sideTOC.</p>
<code>Ctrl FileDepth</code>	<p>FileDepth: Sectioning depth of file splits. Defaults to -5, causing the entire HTML website to be one single file.</p>

- 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 L^AT_EX.

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 T_EX engine was used. X_YLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

4. To recompile the document:

```
Enter ⇒  lwarpmk print  -and-
```

```
Enter ⇒  lwarpmk html
```

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

5.10 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

5.11 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

5.12 Processing multiple projects in the same directory

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

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

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

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

5.13 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

6 Additional details

6.1 Font and UTF-8 support

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

vector fonts
Computer Modern



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

```
usepackage{lmodern}
```

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

```
\usepackage{dejavu}
```

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



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

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

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

- For X_YLaTeX or LuaLaTeX:

Pkg `fontspec`

- `fontspec` and font choices

`ligatures`

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

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

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File `glyphtounicode`

- `\input glyphtounicode.tex`
- `\input glyphtounicode-cmr.tex`% from the `pdfx` package
- `\pdfgentounicode=1`

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

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

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

6.2 lwarp package loading and options

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

Pkg	<code>lwarp</code>	Load the <code>lwarp</code> package immediately after the font and UTF-8 setup commands.
Opt lwarp	<code>warpprint</code>	Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When <code>lwarp-newproject</code> is loaded in print mode, it creates <code><project>_html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code><project>_html.tex</code> will create an HTML version.
Opt lwarp	<code>warpHTML</code>	
Opt lwarp	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 6.10.3.
Opt lwarp	<code>mathjax</code>	

6.3 Selecting the operating system

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

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

6.4 Selecting actions for print or HTML output

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.5 Commands to be placed into the **warpprint** environment

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

- Paragraph formatting: `\parindent` `\parskip`
- Variable spaces such as `\vspace`. `\hfill` is turned into a `\quad`. Fixed spaces such as `\quad` are emulated correctly.
- Manual page positions such as the `textpos` package, which is emulated but only in a limited way.

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

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

6.6 Commands for a successful HTML conversion

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

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

\centering, \raggedright, \raggedleft:

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

Superscripts and other non-math uses of math mode:

Use `x` instead of x

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

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

\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 div class <code>titlepage</code> , and <code>\printtitle</code> , etc. may be used inside this environment. |
| Env <code>titlingpage</code> | Another form of custom title page, where <code>\maketitle</code> is allowed, and additional information may be included as well. |

`\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 Verse and Memoir packages

`\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</p>
--	---

until they are manually changed again. One reason to change `\HTMLleftmargin` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargin` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

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

6.10.3 Math

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

SVG math option For SVG math, math is rendered as usual by \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.

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.

⁷See section 119 regarding fonts and fractions.

MathJax math option	The popular MathJax alternative (mathjax.org) may be used to display math.
<div> <div>Prog</div> <div>MathJax</div> </div>	When MathJax is enabled, math is rendered twice: <ol style="list-style-type: none"> 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:
<div> <div>Prog</div> <div>MathJax</div> </div>	
chapter numbers	<ul style="list-style-type: none"> • In document classes which have chapters, <code>\tagged</code> equations have the chapter number prepended in HTML output, unlike \LaTeX. <code>\tag*</code> equations (correctly) do not. This may be improved with future versions of the MathJax support script. <p>https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY</p>
subequations	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Footnotes inside equations are not yet supported while using MathJax.
lateximage	<ul style="list-style-type: none"> • Math appearing inside a <code>lateximage</code>, and therefore also inside a <code>Tikz</code> or <code>picture</code> environment, is rendered as SVG math even if MathJax is used in the rest of the document.
siunitx	<ul style="list-style-type: none"> • Usage of <code>siunitx</code> inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use <code>\SI</code> or <code>\si</code> inside <code>\text</code>, where it will be rendered as normal text. <p>https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx</p>
<div> <div>△</div> <div>siunitx inside an equation</div> </div>	
\LaTeX macros	<ul style="list-style-type: none"> • 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 208.


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 \LaTeX , 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. 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 Tikz






 **preamble** For recent versions of `Tikz` (v3.0.0 — 2013/12/20 and later), include in the document preamble:

`\usetikzlibrary{babel}` % v3.0.0 and up

For older versions of Tikz without the Tikz babel library, lwarp automatically handles the catcode change for the \$ math shift.

See section 61.

6.10.7 Tabular


column types	<ul style="list-style-type: none"> Vertical rules are not yet supported. * in a column specification is not used (so far). Repeat the column type the correct number of times.
 <code>\multirow & \multicolumn</code>	<ul style="list-style-type: none"> Multirow and multicolumn cannot be used at the same time. (No rectangular holes wider than one column or taller than one row.)
 <code>\multirow</code>	<ul style="list-style-type: none"> For multirow, insert <code>\mrowcell</code> 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.
<code>\multirow with rules</code>	<ul style="list-style-type: none"> If a multirow reaches to the bottom of a table, and <code>\bottomrule</code> does not go over to that edge, try adding a line of empty cells below the <code>\bottomrule</code>. This may be a browser bug.
rule at last row	<ul style="list-style-type: none"> If a <code>\midrule</code> is desired after the last row, an additional row of blank cells must be used.
 paragraphs	<ul style="list-style-type: none"> Multiple paragraphs in one cell of a p, b, m column must have <code>\newline</code> between paragraphs.
<code>\cmidrule width, trim</code>	<ul style="list-style-type: none"> <code>\cmidrule</code> does not support width or trim options due to CSS limitations.
<code>longtable headings</code>	<ul style="list-style-type: none"> For longtable, place headings and footings which do not apply to HTML inside <code>\warpprintonly{}</code>.
 <code>\warpprintonly</code>	<ul style="list-style-type: none"> For <code>\toprule</code> and <code>\bottomrule</code>, when combined with a <code>warpprint</code> or <code>warpHTML</code> environment, if a “misplaced <code>\noalign</code>” error occurs, change <code>This & That \endhead</code> to <code>\warpprintonly{This & That \endhead}</code> and likewise with the other <code>\end</code> headings. Keep the <code>\endfirsthead</code> row unchanged, as it is still relevant to HTML output.
 S columns	<ul style="list-style-type: none"> For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside <code>{ }</code> braces, including commands such as <code>\multirow</code>. 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 \\}
```

6.10.8 Minipages


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

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

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.

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

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

6.10.9 Mdframed

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

 **loading** When used, `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.

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

 `package conflicts` If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 115.1.


6.10.11 `caption` and `subcaption` packages

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

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


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

6.10.12 `floatrow` package

 `\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 `siunitx` package

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

6.10.14 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}
...
```

6.10.15 newtxmath package

Pkg **newtxmath** newtxmath must be loaded after lwarp.



loading

6.10.16 babel package

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

```
\frenchbsetup{StandardLists=true}
```



French

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}{:~}
```

6.10.17 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.

`enumitem`

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 the 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`.

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.

Note that 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.

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 supplied options, then the `lwarp-` version continues loading as well. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable as-is, there is no need to create a `lwarp-` version.

8 Troubleshooting

8.1 Using the lwarp.sty package

Also see:

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

Section 6.5: Commands to be placed into the `warpprint` environment

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 4: 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 4. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the L^AT_EX document structure.

When starting a new section, the program first must close out any existing sections and lists of a deeper level to keep the HTML tags nested correctly.

Support for the `memoir` package will require the addition of a `book` level, which will push the HTML headings down a step, and also cause `subsubsection` to become a `div` due to a limit of six HTML headings.

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

11 Source Code


This is where the documented source code for **lwarp** begins, continuing through the following sections all the way to the change log and index at the end of this document.

The following sections document the actual implementation of the **lwarp** package.

line numbers The small numbers at the left end of a line refer to line numbers in the **lwarp.sty** file.

subjects Blue-colored tags in the left margin aid in quickly identifying the subject of each paragraph.

objects Black-colored tags in the left margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are command macros. Each of these also appears in the index as individual entries, and are also listed together under “files”, “packages”, “environments”, “booleans”, and “counters”.

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

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

lwarp source code begins on the following page.

12 Detecting the T_EX Engine — pdf_latex, lua_latex, xelatex

```

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

```

13 Unicode Input Characters

for HTML & PRINT: If using pdf_latex, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

A commonly-used multiply symbol is declared to be `\texttimes`.

The first arguments of `\newunicodechar` below are text ligatures in the source code, even though they are not printed in the following listing.

```

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

```

In PDF_T_EX, preserve upright quotes in verbatim text:

```

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

```

14 Handling package options

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

```

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

```

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

```

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

```

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

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

```

42 \DeclareVoidOption{warpHTML}{%
43 \PackageInfo{lwarp}{Using option 'warpHTML'}
44 \excludecomment{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 \excludecomment{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@noloadbefore {\langle packagename \rangle} Error if loaded before lwarp.

```

```

70 \newcommand*\LWR@noloadbefore[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@noloadafter {\langle packagename \rangle} Error if loaded after lwarp.

```

```

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

```

```

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

```

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`:

```

86 \LWR@noloadbefore{afterpage}
87 \LWR@noloadbefore{algorithmicx}
88 \LWR@noloadbefore{alltt}
89 \LWR@noloadbefore{bookmark}
90 \LWR@noloadbefore{booktabs}
91 \LWR@noloadbefore{draftwatermark}
92 \LWR@noloadbefore{epigraph}
93 \LWR@noloadbefore{eso-pic}
94 \LWR@noloadbefore{everypage}
95 \LWR@noloadbefore{extramarks}
96 \LWR@noloadbefore{fancyhdr}
97 \LWR@noloadbefore{floatrow}
98 \LWR@noloadbefore{float}
99 \LWR@noloadbefore{geometry}
100 \LWR@noloadbefore{graphics}
101 \LWR@noloadbefore{graphicx}
102 \LWR@noloadbefore{hyperref}
103 \LWR@noloadbefore{keyfloat}
104 \LWR@noloadbefore{letterspace}
105 \LWR@noloadbefore{listings}
106 \LWR@noloadbefore{longtable}
107 \LWR@noloadbefore{ltcaption}
108 \LWR@noloadbefore{mdframed}
109 \LWR@noloadbefore{microtype}
110 %\LWR@noloadbefore{multicol}% loaded by ltxdoc
111 \LWR@noloadbefore{multirow}
112 \LWR@noloadbefore{needspace}
113 \LWR@noloadbefore{nowidow}
114 \LWR@noloadbefore{pagenote}
115 \LWR@noloadbefore{placeins}
116 \LWR@noloadbefore{ragged2e}
117 \LWR@noloadbefore{rotating}
118 \LWR@noloadbefore{setspace}
119 \LWR@noloadbefore{textpos}
120 \LWR@noloadbefore{threeparttable}
121 \LWR@noloadbefore{titleps}
122 \LWR@noloadbefore{titlesec}
123 \LWR@noloadbefore{titletoc}
124 \LWR@noloadbefore{tocloft}
125 \LWR@noloadbefore{trivfloat}

```

```

126 \LWR@noloadbefore{wallpaper}
127 \LWR@noloadbefore{wrapfig}
128 \LWR@noloadbefore{xcolor}
129 \LWR@noloadbefore{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: 130 `\begin{warppall}`

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

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

```

131 \RequirePackage{iftex}
132 \newif\ifxetexorluatex
133 \ifXeTeX
134   \xetexorluatextrue
135 \else
136   \ifLuaTeX
137     \xetexorluatextrue
138   \else
139     \xetexorluatexfalse
140   \fi
141 \fi

```

```

142 \end{warppall}

```

for HTML output: 143 `\begin{warpHTML}`

```

144 \ifxetexorluatex
145 % ^^A \usepackage[no-math]{fontspec}

```

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

```

146 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
147 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}

```

```

148 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
149 \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.

```

150 \RequirePackage {microtype}
151
152 \microtypesetup{
153 protrusion=false,
154 expansion=false,
155 tracking=false,
156 kerning=false,
157 spacing=false}
158
159 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

160 \fi

161 \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:

```

162 \begin{warpHTML}
163 \RequirePackage[paperheight=190in,paperwidth=20in,%
164 left=7in,right=7in,%
165 top=1in,bottom=1in,%
166 ]{geometry}
167 \end{warpHTML}

```

for HTML & PRINT:

```

168 \begin{warpall}

```

Pkg **xparse**

L^AT_EX3 command argument parsing

```

169 \RequirePackage{xparse}

```

```
170 \end{warpall}
```

for HTML output: 171 \begin{warpHTML}

Pkg **expl3**

L^AT_EX3 programming

```
172 \RequirePackage{expl3}
```

Pkg **getttitlestring**

Used to emulate \nameref.

```
173 \RequirePackage{getttitlestring}
```

Pkg **everyhook**

everyhook is used to patch paragraph handling.

```
174 \RequirePackage{everyhook}
```

```
175 \end{warpHTML}
```

for HTML & PRINT: 176 \begin{warpall}

Pkg **fancyvrb**

Used for Verbatim, verse.

```
177 \RequirePackage{fancyvrb}
```

```
178 \end{warpall}
```

for HTML output: 179 \begin{warpHTML}

Pkg **xifthen**

```
180 \RequirePackage{xifthen}
```

Pkg **xstring**

```
181 \RequirePackage{xstring}
```

Pkg **makeidx**

```
182 \RequirePackage{makeidx}
```

```
183 \makeindex
```

Pkg `calc`

```
184 \RequirePackage{calc}
```

Pkg `newfloat`

```
185 \RequirePackage{newfloat}
```

Pkg `caption`

```
186 \RequirePackage{caption}
```

Pkg `subcaption`

```
187 \RequirePackage{subcaption}
```

Pkg `enumitem`

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

```
188 \RequirePackage{enumitem}
189 \setlist[itemize]{leftmargin=0em}
190 \setlist[enumerate]{leftmargin=0em}
191 \setlist[description]{leftmargin=0em}
```

Pkg `pagenote`

Used to convert footnotes into pagenotes.

```
192 \RequirePackage[continuous]{pagenote}
193 \makepagenote
194 \let\LWR@origmakepagenote\makepagenote
195 \let\LWR@origprintnotes\printnotes
196 \let\LWR@origpagenote\pagenote
197 \end{warpHTML}
```

for HTML & PRINT: 198 `\begin{warpall}`

Pkg `titling`

Used for `\maketitle` and the title page. See section [42](#).

```
199 \RequirePackage{titling}
```

```
200 \end{warpall}
```

for HTML output: 201 `\begin{warpHTML}`

Pkg **zref**

Used for cross-references.

```
202 \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.

```
203 \PassOptionsToPackage{leqno,fleqn}{amsmath}
204 \RequirePackage{amsmath}
```

Pkg **environ**

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

```
205 \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.

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

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

```
207 \let\LWR@origpagestyle\pagestyle
208 \renewcommand*\pagestyle[1]{}
```

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

```
209 \let\LWR@origthispagestyle\thispagestyle
210 \renewcommand*\thispagestyle[1]{}
```

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

```
211 \let\LWR@origpagenumbering\pagenumbering
212 \renewcommand*\pagenumbering[1]{}
```

Pkg **xfrac**

Patched for HTML use. See section 119.

```
213 \RequirePackage{xfrac}
```

Used to convert lengths for image width/height options.

```
214 \RequirePackage{printlen}
```

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

17 Loading Packages

for HTML output: 216 \begin{warpHTML}

Remember the original \RequirePackage:

```
217 \let\LWR@origRequirePackage\RequirePackage
```

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

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

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

Note that argument 4 is passed directly to \StrBetween.

```
219 \newcommand*\LWR@findword[3][,]{%
220   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
221 }
```

\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.

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

Find the n'th package name from the list:

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

See if the package name was found:


```

224 \IfStrEq{\LWR@strresult}{  

225 {}  

226 {}  


```

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

```

227 \IfFileExists{lwarp-\LWR@strresult.sty}  

228 {}  

229 \StrSubstitute  

230 {\LWR@requirepackagenames}  

231 {\LWR@strresult}  

232 {lwarp-\LWR@strresult}[\LWR@requirepackagenames]  

233 }  

234 {}  

235 {}  

236 }

```

`\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.

```

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

```

Redirect up to nine names:

```

238 \renewcommand*\LWR@requirepackagenames{#2}  

239 \LWR@lookforpackagename{1}  

240 \LWR@lookforpackagename{2}  

241 \LWR@lookforpackagename{3}  

242 \LWR@lookforpackagename{4}  

243 \LWR@lookforpackagename{5}  

244 \LWR@lookforpackagename{6}  

245 \LWR@lookforpackagename{7}  

246 \LWR@lookforpackagename{8}  

247 \LWR@lookforpackagename{9}

```

`\RequirePackage` depending on the options and version:

```

248 \IfValueTF{#1}  

249 {}  

250 \IfValueTF{#3}% version given?  

251 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}  

252 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}  

253 }  

254 {}  

255 \IfValueTF{#3}% version given?  

256 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}

```

```

257 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
258 }
259 }
260 \let\usepackage\RequirePackage

```

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

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

```

261 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
262 \PackageInfo{lwarp}{Using the lwarp version of package ‘#1’, including options,}%
263 \IfValueTF{#2}
264 {\ProvidesPackage{lwarp-#1}[#2]}
265 {\ProvidesPackage{lwarp-#1}}
266 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
267 \ProcessOptions\relax
268
269 \IfValueTF{#2}
270 {\LWR@origRequirePackage{#1}[#2]}
271 {\LWR@origRequirePackage{#1}}
272 }

```

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

Provides a new package which drops/discards its options.

```

273 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
274 \PackageInfo{lwarp}{Using the lwarp version of package ‘#1’, discarding options,}%
275 \IfValueTF{#2}
276 {\ProvidesPackage{lwarp-#1}[#2]}
277 {\ProvidesPackage{lwarp-#1}}
278 \DeclareOption*{}
279 \ProcessOptions\relax
280 }

281 \end{warpHTML}

```

18 Copying a file

for HTML output: 282 \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.

```

283 \newcommand*{\LWR@copyfile}[2]{%
284 \newwrite\copyfile % open the file to write to
285 \immediate\openout\copyfile=#2
286 \newread\file % open the file to read from
287 \openin\file=#1
288 \begingroup\endlinechar=-1
289 \makeatletter
290 \loop\unless\ifeof\file
291 \read\file to\fileline % Read one line and store it into \fileline
292 % \fileline\par % print the content into the pdf
293 % print the content:
294 \immediate\write\copyfile{\unexpanded\expandafter{\fileline}}%
295 \repeat
296 \closeout\copyfile
297 \endgroup
298 }

299 \end{warpHTML}

```

19 Debugging messages

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

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

`\tracinglwarp` Turns on the debug tracing messages.

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

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

```

302 \newcommand{\LWR@traceinfo}[1]{%
303 \ifbool{LWR@tracinglwarp}%
304 {\PackageInfo{lwarp}{#1 : }}%
305 {}%
306 }

```

20 Remembering original formatting macros

for HTML output: `307 \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.

```

308 \let\LWR@origtextit\textit
309 \let\LWR@origtextbf\textbf
310 \let\LWR@origtexttt\texttt
311 \let\LWR@origtextsc\textsc
312 \let\LWR@origtextsf\textsf
313 \let\LWR@origtextrm\textrm
314 \let\LWR@origbfseries\bfseries
315 \let\LWR@origrmfamily\rmfamily
316 \let\LWR@origttfamily\ttfamily
317 \let\LWR@orignormalfont\normalfont
318
319 \let\LWR@origraggedright\raggedright
320 \let\LWR@origonecolumn\onecolumn
321
322 \let\LWR@origtextsuperscript\textsuperscript
323 \let\LWR@origtextsubscript\textsubscript
324
325 \let\LWR@origscriptsize\scriptsize
326
327 \let\LWR@orignewpage\newpage
328
329 \let\LWR@origminipage\minipage
330 \let\LWR@origendminipage\endminipage
331
332 \let\LWR@orignewline\newline
333
334 \let\LWR@origitem\item
335
336 \let\LWR@origpar\par
337
338
339 \let\LWR@origfootnote\footnote
340 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
341
342 \let\LWR@origclearpage\clearpage
343 \let\clearpage\relax
344 \let\cleardoublepage\relax
345 \end{warpHTML}

```

21 Operating-System portability

lwarp tries to detect which operating system is being used.

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

for HTML & PRINT: 346 `\begin{warpall}`

21.1 Common portability code

Bool **usingOSwindows** Set if `\warpOSwindows`.

```

347 \newbool{usingOSwindows}
348 \boolfalse{usingOSwindows}

```

21.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

```

349 \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.

```

350 \newcommand*{\warpOSwindows}
351 {
352 \booltrue{usingOSwindows}
353 \renewcommand*{\OSPathSymbol}{\@backslashchar}
354 }

```

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

```

355 \ifwindows
356 \warpOSwindows
357 \fi

358 \end{warpall}

```

22 Stacks

for HTML output: 359 `\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:

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

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

```
361 \newcommand*{\LWR@depthfinished}{-4}
362 \newcommand*{\LWR@depthpart}{-1}
363 \newcommand*{\LWR@depthchapter}{0}
364 \newcommand*{\LWR@depthsection}{1}
365 \newcommand*{\LWR@depthsubsection}{2}
366 \newcommand*{\LWR@depthsubsubsection}{3}
367 \newcommand*{\LWR@depthparagraph}{4}
368 \newcommand*{\LWR@depthsubparagraph}{5}
```

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

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

used by `\item`:

```
370 \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.

```

371 \newcommand*{\LWR@closeone}{}% top of the stack
372 \newcommand*{\LWR@closetwo}{}
373 \newcommand*{\LWR@closeethree}{}
374 \newcommand*{\LWR@closefour}{}
375 \newcommand*{\LWR@closefive}{}
376 \newcommand*{\LWR@closesix}{}
377 \newcommand*{\LWR@closeseven}{}
378 \newcommand*{\LWR@closeeight}{}
379 \newcommand*{\LWR@closenine}{}
380 \newcommand*{\LWR@closeten}{}
381 \newcommand*{\LWR@closeeleven}{}
382 \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}

```

```

383 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
384 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
385 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
386 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
387 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
388 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
389 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
390 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
391 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
392 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
393 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}

```

```
394 \newcommand*{\LWR@closedepthtwelve}{\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.

```
395 \NewDocumentCommand{\pushclose}{m m}
396 {
397 \let\LWR@closetwelve\LWR@closeeleven
398 \let\LWR@closeeleven\LWR@closeten
399 \let\LWR@closeten\LWR@closenine
400 \let\LWR@closenine\LWR@closeeight
401 \let\LWR@closeeight\LWR@closeseven
402 \let\LWR@closeseven\LWR@closesix
403 \let\LWR@closesix\LWR@closefive
404 \let\LWR@closefive\LWR@closefour
405 \let\LWR@closefour\LWR@closethree
406 \let\LWR@closethree\LWR@closetwo
407 \let\LWR@closetwo\LWR@closeone
408 \let\LWR@closeone#1
409 \let\LWR@closedepthtwelve\LWR@closedeptheleven
410 \let\LWR@closedeptheleven\LWR@closedephten
411 \let\LWR@closedephten\LWR@closedepthnine
412 \let\LWR@closedepthnine\LWR@closedeptheight
413 \let\LWR@closedeptheight\LWR@closedepthseven
414 \let\LWR@closedepthseven\LWR@closedepthsix
415 \let\LWR@closedepthsix\LWR@closedepthfive
416 \let\LWR@closedepthfive\LWR@closedepthfour
417 \let\LWR@closedepthfour\LWR@closedepththree
418 \let\LWR@closedepththree\LWR@closedepthtwo
419 \let\LWR@closedepthtwo\LWR@closedepthone
420 \let\LWR@closedepthone#2
421 }
```

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

```
422 \newcommand*{\popclose}
423 {
424 \let\LWR@closeone\LWR@closetwo
425 \let\LWR@closetwo\LWR@closethree
426 \let\LWR@closethree\LWR@closefour
427 \let\LWR@closefour\LWR@closefive
428 \let\LWR@closefive\LWR@closesix
429 \let\LWR@closesix\LWR@closeseven
430 \let\LWR@closeseven\LWR@closeeight
```



```

431 \let\LWR@closeeight\LWR@closenine
432 \let\LWR@closenine\LWR@closeten
433 \let\LWR@closeten\LWR@closeeleven
434 \let\LWR@closeeleven\LWR@closetwelve
435 \let\LWR@closedepthone\LWR@closedepthtwo
436 \let\LWR@closedepthtwo\LWR@closedepththree
437 \let\LWR@closedepththree\LWR@closedepthfour
438 \let\LWR@closedepthfour\LWR@closedepthfive
439 \let\LWR@closedepthfive\LWR@closedepthsix
440 \let\LWR@closedepthsix\LWR@closedepthseven
441 \let\LWR@closedepthseven\LWR@closedeptheight
442 \let\LWR@closedeptheight\LWR@closedepthnine
443 \let\LWR@closedepthnine\LWR@closedepthten
444 \let\LWR@closedepthten\LWR@closedeptheleven
445 \let\LWR@closedeptheleven\LWR@closedepthtwelve
446 }

447 \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: 448 \begin{warpHTML}

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

```

449 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
450 \expandafter\edef\csname #1#2\endcsname{\expandonce#3}%
451 }

```

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

```

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

453 \end{warpHTML}

```

24 HTML entities

```

for HTML output: 454 \begin{warpHTML}

                  HTML entites and HTML Unicode entities:

455 \let\LWR@origampersand\&

\HTMLentity      {\entitytag}

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

\HTMLunicode     {\hex_unicode}

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

\&

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

\textless
\textgreater
459 \let\LWR@origtextless\textless
460 \renewcommand*{\textless}{\HTMLentity{lt}}
461
462 \let\LWR@origtextgreater\textgreater
463 \renewcommand*{\textgreater}{\HTMLentity{gt}}

464 \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: 465 \begin{warpall}

\BaseJobname      The \jobname of the printed version, even if currently compiling the HTML ver-
                  sion. I.e. this is the \jobname without _html appended. This is used to set
                  \HomeHTMLFileName if the user did not provide one.

466 \providecommand*{\BaseJobname}{\jobname}

```

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

```
467 \providecommand*\HTMLFileName{}
```

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

```
468 \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.

```
469 \newcommand*\SetHTMLFileNumber[1]{%
470 \setcounter{LWR@htmlfilenumber}{#1}%
471 }
```

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

Defaults to use section names in the filenames.

```
472 \newbool{FileSectionNames}
473 \booltrue{FileSectionNames}

474 \end{warpall}
```

for HTML output: 475 `\begin{warpHTML}`

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

```
476 \newcounter{LWR@htmlfilenumber}
477 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
478 \newcommand*\LWR@htmlsectionfilename[1]{%
479 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```

480 \LWR@traceinfo{about to assign temp}%
481 \edef\LWR@tempone{#1}%
482 \LWR@traceinfo{about to compare with ??}%
483 \ifthenelse{\equal{\LWR@tempone}{??}}{%
484 {%
485 \LWR@traceinfo{found ??}%
486 }{%
487 \LWR@traceinfo{not found ??}%
488 }%
489 \LWR@traceinfo{about to compare with zero or empty}%
490 \ifthenelse{%
491 \equal{\LWR@tempone}{0}%
492 \OR \equal{\LWR@tempone}{}%
493 \OR \equal{\LWR@tempone}{??}}%
494 }%
495 {%
496 \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFileName.html}%
497 \HomeHTMLFileName.html%
498 }%

```

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`:

```

499 {%
500 \LWR@traceinfo{\LWR@htmlsectionfilename C}%
501 \ifthenelse{%
502 \equal{\HTMLFileName}{ } \AND \equal{\LWR@tempone}{Index} \OR \equal{\LWR@tempone}{index}%
503 }%
504 {\_#1.html}%

```

Otherwise, create a filename with the chosen prefix:

```

505 {\HTMLFileName#1.html}%
506 }%
507 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
508 }

```

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

Prints the filename for the given label

```

509 \newcommand*\LWR@htmlrefsectionfilename[1]{%
510 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
511 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
512 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
513 }

514 \end{warpHTML}

```

26 Homepage link

for HTML output: 515 \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.

```
516 \newcommand*{\LinkHome}{%
517 \LWR@subhyperrefclass{%
518 \HomeHTMLFileName.html}%
519 {Home}{linkhome}}%
520 }
```

`\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.

```
521 \newcommand*{\LWR@topnavigation}{
522 \LWR@htmllementclassline{nav}{topnavigation}{\LinkHome}
523 }
```

`\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.

```
524 \newcommand*{\LWR@botnavigation}{
525 \LWR@htmllementclassline{nav}{botnavigation}{\LinkHome}
526 }
```

```
527 \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: 528 \begin{warpHTML}

`\PrintStack` Prints the closedepth stack.

```
529 \newcommand*{\PrintStack}{
530 \LWR@startpars
531 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
532 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}}
```

```

533 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
534 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
535 }

536 \end{warpHTML}

```

28 Closing stack levels

for HTML output: 537 \begin{warpHTML}

Close one nested level:

```

538 \newcommand*{\LWR@closeoneprevious}{%
539
540 \LWR@closeone{}
541
542 \popclose{}
543 }

```

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

```

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

Close any pending paragraph:

545 \LWR@stoppars

Close anything nested deeper than the desired depth:

546 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
547 }

548 \end{warpHTML}

```

29 HTML tags, spans, divs, elements

for HTML output: 549 \begin{warpHTML}

29.1 Mapping L^AT_EX Sections to HTML Sections

```

550 \newcommand*{\LWR@tagpart}{h2}

```

```

551 \newcommand*{\LWR@tagpartend}{/h2}
552 \newcommand*{\LWR@tagchapter}{h3}
553 \newcommand*{\LWR@tagchapterend}{/h3}
554 \newcommand*{\LWR@tagsection}{h4}
555 \newcommand*{\LWR@tagsectionend}{/h4}
556 \newcommand*{\LWR@tagsubsection}{h5}
557 \newcommand*{\LWR@tagsubsectionend}{/h5}
558 \newcommand*{\LWR@tagsubsubsection}{h6}
559 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
560 \newcommand*{\LWR@tagparagraph}{span class="paragraph">{}}
561 \newcommand*{\LWR@tagparagraphend}{/span}
562 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph">{}}
563 \newcommand*{\LWR@tagsubparagraphend}{/span}
564
565 \newcommand*{\LWR@tagregularparagraph}{p}

```

29.2 HTML tags

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

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

```

566 \newcommand*{\LWR@htmltagc}[1]{%
567 {%
568 \LWR@origttfamily%
569 % \protect\LWR@origtexttt{%
570 % \protect\LWR@origtextbf{%
571 \LWR@origtextless#1\LWR@origtextgreater%
572 % }%
573 % }%
574 }%
575 }

```

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.

```

576 \newenvironment*{LWR@nestspan}
577 {%
578 \addtocounter{LWR@spandepth}{1}%
579 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
580 }%
581 {\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.

```

582 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
583 \LWR@ensuredoingapar%
584 \LWR@htmltagc{#1}%
585 \begin{LWR@nestspan}%
586 #2%
587 \LWR@htmltagc{/#1}%
588 \end{LWR@nestspan}%
589 }

```

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

```

590 \NewDocumentCommand{\LWR@htmlspanclass}{m o +m}{%
591 \LWR@ensuredoingapar%
592 \LWR@subhtml elementclass{span}{#1}[#2]%
593 \begin{LWR@nestspan}%
594 #3%
595 \LWR@htmltagc{/span}%
596 \end{LWR@nestspan}%
597 }

```

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

print an HTML tag: `<tag>`

```

598 \newcommand*{\LWR@htmltagb}[1]{%
599 \LWR@htmltagc{#1}%
600 \endgroup%
601 }
602
603 \newcommand*{\LWR@htmltag}{%
604 \begingroup\catcode'\_ =12
605 \LWR@htmltagb%
606 }

```

29.3 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment

```



```

607 \newcommand*{\LWR@htmlopencomment}{%
608 % \LWR@origtextttt{\LWR@origtextbf{\LWR@origtextless{!}{-}{-}}}%
609 \begingroup\LWR@origttfamily\LWR@origtextless{!}{-}{-}\endgroup%
610 }
611
612 \newcommand*{\LWR@htmlclosecomment}{%
613 % \LWR@origtextttt{\LWR@origtextbf{-}{-}\LWR@origtextgreater{}}}%
614 \begingroup\LWR@origttfamily{-}{-}\LWR@origtextgreater{}\endgroup%
615 }

```

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

```

616 \newcommand{\LWR@htmlcomment}[1]{%
617 \LWR@htmlopencomment}%
618 \LWR@origtextrm{#1}%
619 \LWR@htmlclosecomment}%

```

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

```

620 \newcommand{\LWR@htmlblockcommentb}[1]
621 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
622
623 \newcommand{\LWR@htmlblockcomment}
624 {%
625 \begingroup\catcode'\_ =12%
626 \LWR@htmlblockcommentb%
627 }

```

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

```

628 \newcommand*{\LWR@htmlblocktag}[1]{%
629 \LWR@stoppars%
630 \LWR@htmltag{#1}%
631 \LWR@startpars%
632 }

```

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

```

633 \NewDocumentCommand{\LWR@htmlblocktagcomment}{m +m}{%
634 \LWR@stoppars%
635 \LWR@htmltag{#1}\LWR@htmlcomment{#2}\LWR@orignewline%
636 \LWR@startpars%
637 }

```

29.4 Div class and element class

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

Factored and reused in several places.

```
638 \NewDocumentCommand{\LWR@subhtmllementclass}{m m o}{%
639 \IfValueTF{#3}%
640 {% option
641 \ifthenelse{\equal{#3}{}}{%
642 {\LWR@htmltag{#1 class="#2"}}% empty option
643 {\LWR@htmltag{#1 class="#2" style="#3"}}% non-empty option
644 }% option
645 {\LWR@htmltag{#1 class="#2"}}% no option
646 }
```

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

```
647 \NewDocumentCommand{\LWR@htmllementclass}{m m o}{%
648 \LWR@stoppars%
649 \LWR@subhtmllementclass{#1}{#2}[#3]%
650 \LWR@startpars%
651 }
```

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

```
652 \newcommand*{\LWR@htmllementclassend}[2]{%
653 \LWR@stoppars%
654 \LWR@htmltag{/#1}%
655 \LWR@htmlcomment{End of #1 ‘‘#2’’}%
656 \LWR@startpars%
657 }
```

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

```
658 \NewDocumentCommand{\LWR@htmldivclass}{m o}{%
659 \LWR@htmllementclass{div}{#1}[#2]%
660 }
```

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

```
661 \newcommand*{\LWR@htmldivclassend}[1]{%
662 \LWR@htmllementclassend{div}{#1}%
663 }
```

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}

664 \NewDocumentCommand{\LWR@htmlelementclassline}{m m o +m}{%
665 \LWR@stoppars
666 \LWR@subhtmlelementclass{#1}{#2}[#3]%
667 #4%
668 \LWR@htmltag{/#1}
669 \LWR@startpars
670 }
```

29.6 HTML5 semantic elements

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

671 \newcommand*{\LWR@htmlelement}[1]{%
672 \LWR@htmlblocktag{#1}
673 }
```

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

674 \newcommand*{\LWR@htmlelementend}[1]{%
675 \LWR@stoppars
676 \LWR@htmltag{/#1}
677 \LWR@startpars
678 }
679
680 \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 [63](#).

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

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

for HTML output: 681 `\begin{warpHTML}`
 682 `\NewDocumentEnvironment{BlockClass}{m o}`
 683 `{`
 684 `\LWR@htmldivclass{#1}[#2]`
 685 `}`
 686 `{`
 687 `\LWR@htmldivclassend{#1}`
 688 `}`
 689 `\end{warpHTML}`

for PRINT output: 690 `\begin{warpprint}`
 691 `\NewDocumentEnvironment{BlockClass}{m o}{\}\}`
 692 `\end{warpprint}`

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

for HTML output: 693 `\begin{warpHTML}`
 694 `\newcommand{\BlockClassSingle}[2]{%`
 695 `\LWR@html@element@classline{div}{#1}{#2}%`
 696 `}`
 697 `\end{warpHTML}`

for PRINT output: 698 `\begin{warpprint}`
 699 `\newcommand{\BlockClassSingle}[2]{#2}`
 700 `\end{warpprint}`

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

for HTML output: 701 `\begin{warpHTML}`
 702 `\NewDocumentCommand{\InlineClass}{m o m}{%`
 703 `\LWR@htmlspanclass{#1}[#2]{#3}%`
 704 `}`
 705 `\end{warpHTML}`

for PRINT output: 706 `\begin{warpprint}`
 707 `\NewDocumentCommand{\InlineClass}{m o m}{#3}`
 708 `\end{warpprint}`

29.8 Closing HTML tags

for HTML output: 709 `\begin{warpHTML}`

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

```

710 \newcommand*{\LWR@printclosepart}
711     {\LWR@htmlcomment{Closing part}}
712 \newcommand*{\LWR@printclosechapter}
713     {\LWR@htmlcomment{Closing chapter}}
714 \newcommand*{\LWR@printclosesection}
715     {\LWR@htmlcomment{Closing section}}
716 \newcommand*{\LWR@printclosesubsection}
717     {\LWR@htmlcomment{Closing subsection}}
718 \newcommand*{\LWR@printclosesubsubsection}
719     {\LWR@htmlcomment{Closing subsubsection}}
720 \newcommand*{\LWR@printcloseparagraph}
721     {\LWR@htmlcomment{Closing paragraph}}
722 \newcommand*{\LWR@printclosesubparagraph}
723     {\LWR@htmlcomment{Closing subparagraph}}

```

Lists require closing HTML tags:

```

724 \newcommand*{\LWR@printcloselistitem}
725     {\LWR@htmltag{/li}}
726 \newcommand*{\LWR@printclosedescitem}
727     {\LWR@htmltag{/dd}}
728 \newcommand*{\LWR@printcloseitemize}
729     {\LWR@htmltag{/ul}}
730 \newcommand*{\LWR@printcloseenumerate}
731     {\LWR@htmltag{/ol}}
732 \newcommand*{\LWR@printclosedescription}
733     {\LWR@htmltag{/dl}}

734 \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: 735 `\begin{warpHTML}`

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

```

736 \newcounter{LWR@spandepth}
737 \setcounter{LWR@spandepth}{0}

```

Bool LWR@doingstartpars Tells whether paragraphs may be generated.

```

738 \newbool{LWR@doingstartpars}
739 \boolfalse{LWR@doingstartpars}

```

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

```

740 \newbool{LWR@doingapar}
741 \global\boolfalse{LWR@doingapar}

```

\LWR@ensuredoingapar If are about to print something visible, and if allowed to start a new paragraph, ensure that are LWR@doingapar, so that paragraph tags are placed:

```

742 \newcommand*{\LWR@ensuredoingapar}{%
743 \ifbool{LWR@doingstartpars}%
744 {\global\booltrue{LWR@doingapar}}%
745 {}%
746 }

```

\LWR@openparagraph

```

747 \newcommand*{\LWR@openparagraph}
748 {%

```

See if paragraph handling is enabled:

```

749 \ifbool{LWR@doingstartpars}%
750 {% handling pars

```

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

```

751 \ifthenelse{%
752 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
753 \cnttest{\value{LWR@spandepth}}{>}{0}%
754 }{% 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`.

```

755 {}% no nested par tags

```

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

```

756 {% yes nest par tags
757 \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

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

See if are nested inside an item list:

```

758 \ifnumcomp{\@listdepth}{>}{0}%
759 {%

```

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

```

760 \LWR@orighspace{1in}%
761 }{}%

```

Now have started a paragraph.

```

762 \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.)

```

763 \let\par\LWR@closeparagraph%
764 }% end of yes nest par tags
765 }% end of handling pars
766 {}% not handling pars
767 }

```

`\LWR@closeparagraph`

```

768 \newcommand*{\LWR@closeparagraph}
769 {%

```

See if paragraph handling is enabled:

```

770 \ifbool{\LWR@doingapar}%

```

If currently in paragraph mode:

```

771 {% handling pars

```

See if already started a `lateximage` or a `span`:

```

772 \ifthenelse{%

```

```

773 \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
774 \cnttest{\value{LWR@spandepth}}{>}{0}%
775 }%

```

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

```

776 {% no nested par tags
777 \ifthenelse{%
778 \cnttest{\value{LWR@spandepth}}{>}{0}%
779 \AND%
780 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
781 }%
782 {%
783 \ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}%
784 }%
785 {}%
786 }% no nested par tags

```

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

```

787 {% yes nest par tags

```

Print a closing tag:

```

788 \unskip%
789 \LWR@htmltagc{/LWR@tagregularparagraph}%

```

No longer doing a paragraph:

```

790 \global\boolfalse{LWR@doingapar}%
791 % Disable the special \env{minipage} & \cs{hspace} interaction
792 % until a new minipage is found:
793 % \begin{macrocode}
794 \global\boolfalse{LWR@minipagethispar}%
795 }% end of yes nest par tags
796 }% end of handling pars

```

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

```

797 {% not handling pars
798 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
799 {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
800 {}%
801 }% not handling pars

```

Finish with regular paragraph processing

```

802 \LWR@origpar%

```



```

803 }
804 \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: 805 `\begin{warpHTML}`

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

```

806 \newcommand*{\LWR@startpars}%
807 {%

```

See if currently handling HTML paragraphs:

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

```

If already in paragraph mode, do nothing.

```
809 {}%

```

If not currently in paragraph mode:

```
810 {%

```

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

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

```

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

```

812 \let\par\LWR@closeparagraph
813
814 }% an intentionally blank line

```

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

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

```

No <par> tag yet to undo:

```
816 \global\boolfalse{LWR@doingapar}%
817 }
```

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

```
818 \newcommand*{\LWR@stoppars}%
819 {%
```

See if currently handling HTML paragraphs:

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

if currently in an HTML paragraph:

```
821 {%
```

Print a closing tag:

```
822 \unskip%
823 \LWR@htmltagc{/LWR@tagregularparagraph}%
```

No longer have an open HTML paragraph:

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

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

```
825 \global\boolfalse{LWR@minipagethispar}
826
827 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
828 {}%
```

See if currently allowing HTML paragraphs:

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

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

```
830 {%
831 \ClearPreHook{par}%
832 }%
```

Else: do nothing

833 {}%

no longer in paragraph mode

834 \global\setbool{LWR@doingstartpars}{false}%

no <p> tag to undo:

835 \global\boolfalse{LWR@doingapar}%

836 }

837 \end{warpHTML}

32 Page headers and footers

for HTML & PRINT: 838 \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.

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

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

841 \newcommand{\LWR@pagebottom}{}%

842

843 \newcommand{\LWR@setfirstpagetopb}[1]{%

844 \renewcommand{\LWR@firstpagetop}{#1}

845 \catcode'_ =8

846 }

\SetFirstPageTop {*{text and logos}*}

847 \newcommand{\SetFirstPageTop}{%

848 \catcode'_ =12

849 \LWR@setfirstpagetopb

850 }

851 \newcommand{\LWR@setpagetopb}[1]{%

852 \renewcommand{\LWR@pagetop}{#1}

853 \catcode'_ =8

854 }

\SetPageTop {*{text and logos}*}

```

855 \newcommand{\SetPageTop}{%
856 \catcode'\_ =12
857 \LWR@setpagetopb
858 }

859 \newcommand{\LWR@setpagebottomb}[1]{%
860 \renewcommand{\LWR@pagebottom}{#1}
861 \catcode'\_ =8
862 }

```

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

```

863 \newcommand{\SetPageBottom}{%
864 \catcode'\_ =12
865 \LWR@setpagebottomb
866 }

867 \end{warpall}

```

33 CSS

for HTML output: 868 \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.

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

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

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

```

870 \newcommand*{\LWR@newcssb}[1]{%
871 \renewcommand*{\LWR@currentcss}{#1}
872 \catcode'\_ =8
873 }
874
875 \newcommand*{\NewCSS}{
876 \catcode'\_ =12
877 \LWR@newcssb
878 }
879 \end{warpHTML}

```

for PRINT output: 880 \begin{warpprint}

```
881 \newcommand*{\NewCSS}[1]{}
882 \end{warpprint}
```

34 HTML meta description

for HTML output: 883 \begin{warppHTML}

\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.

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

```

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

Assigns the HTML file's description meta tag

```
885 \newcommand{\NewHTMLdescription}[1]{%
886 \renewcommand{\LWR@currentHTMLdescription}{#1}
887 }
888
889 \end{warppHTML}
```

for PRINT output: 890 \begin{warpprint}
891 \newcommand{\NewHTMLdescription}[1]{}
892 \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 $\text{\LaTeX 2}_{\epsilon}$ 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 62.2 for minipage footnotes.

35.3 Titlepage thanks

See section 42.6 for titlepage footnotes.

35.4 Regular page footnote implementation

for HTML output: 893 \begin{warpHTML}

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

894 \AtEndPreamble{

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

895 \renewcommand*{\notedivision}

896 {\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.

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

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

898 \let\footnote\LWR@origpagenote

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

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

```
899 \renewcommand{\notenuminnotes}[1]{%
900   \textsuperscript{#1} }
```

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

```
901 \let\LWR@orignoteentry\noteentry
902 \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`.

```
903 \begingroup%
904 \renewcommand*{\@currentlabel}{#1}%
905 \renewcommand*{\cref@currentlabel}{[pagenote] [#1] [] {#1}}%
```

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

```
906 \LWR@orignoteentry{#1}{#2}{#3}{#4}%
907 \endgroup%
908 }
```

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

```
909 \newcommand*{\LWR@printpendingnotes}{
```

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

```
910 \ifthenelse{\cnttest{\value{pagenote}}{>}{0}}{%
911 {
912 \begin{BlockClass}{footnotes}
```

Print any pending endnotes, allowing new endnotes to accumulate.

```
913 \LWR@origprintnotes*
```

Start over with new page notes.

```
914 \setcounter{pagenote}{0}
915 \end{BlockClass}
916 }
917 {}
918 }
```

```
919 }% \AtEndPreamble
```

```
920 \end{warpHTML}
```

36 Marginpars

for HTML output: 921 \begin{warpHTML}

```
\marginpar  [⟨left⟩] {⟨right⟩}
```

```
922 \renewcommand{\marginpar}[2] [] {%
923 \LWR@htmlspanclass{\marginpar}{#2}%
924 }
```

```
925 \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: 926 \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.

```
927 \newcounter{FileDepth}
928 \setcounter{FileDepth}{-5}
```

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

```
929 \newbool{CombineHigherDepths}
930 \booltrue{CombineHigherDepths}
```

```
931 \end{warpall}
```

for HTML output: 932 \begin{warpHTML}

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

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


`\LWR@thisnewfilename` The filename being sanitized.

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

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

Convert blanks into dashes, removes short words, store result in `\LWR@thisfilename`.



Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` L^AT_EX commands which appear in section names and TOC captions.

```
935 \newcommand*{\LWR@filenameno blanks}[1]{%
936 \begin{group}
```

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

```
937 \renewcommand*{\HTMLUnicode}[1]{%
938 \renewcommand*{\HTMLentity}[1]{##1}
939 \renewcommand*{\LWR@htmltagc}[1]{%
940 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}
```

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

```
941 \renewcommand*{\&}{and}
942 \renewcommand{\textit}[1]{##1}
943 \renewcommand{\textsc}[1]{##1}
944 \renewcommand{\textsl}[1]{##1}
945 \renewcommand{\textbf}[1]{##1}
946 \renewcommand{\texttt}[1]{##1}
947 \renewcommand{\textsf}[1]{##1}
948 \renewcommand{\textrm}[1]{##1}
949 \renewcommand{\textsuperscript}[1]{##1}
950 \renewcommand{\textsubscript}[1]{##1}
```

Replaces common symbols and short words with hyphens:

```
951 \edef\LWR@thisnewfilename{#1}
952 \fullexpandarg
```

Convert spaces into hyphens:

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

Convert punctuation into hyphens:

```

954 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
955 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
956 \StrSubstitute{\LWR@thisnewfilename}%
957 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
958 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
959 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
960 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
961 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
962 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
963 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
964 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
965 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
966 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
967 \StrSubstitute{\LWR@thisnewfilename}%
968 {\textless}{-}[\LWR@thisnewfilename]
969 \StrSubstitute{\LWR@thisnewfilename}%
970 {\textgreater}{-}[\LWR@thisnewfilename]
971 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
972 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
973 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
974 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
975 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
976 \StrSubstitute{\LWR@thisnewfilename}%
977 {\textbackslash}{-}[\LWR@thisnewfilename]
978 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
979 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
980 %      "~{" for babel
981 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
982 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
983 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

984 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
985 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
986 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
987 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
988 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
989 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
990 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
991 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
992 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
993 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
994 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
995 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
996 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
997 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
998 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
999 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
1000 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]

```

```
1001 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```
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 %      emdash
1008 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
1009 %      endash
1010 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
1011 \endgroup
1012 }
```

`\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.

```
1013 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```
1014 \LWR@html elementclassend{section}{textbody}
1015
1016 \LWR@printpendingnotes
1017
1018 \LWR@html element{footer}
1019
1020 \LWR@pagebottom
1021
1022 \LWR@html elementend{footer}
```

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

```
1023 \ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

End of this HTML file:

```
1024 \LWR@stoppars
1025 \LWR@htmltag{/body}\LWR@orignewline
1026 \LWR@htmltag{/html}\LWR@orignewline
1027
```

```
1028 \addtocounter{LWR@htmlfilename}{1}%
```

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

```
1029 \ifbool{FileSectionNames}%
1030 {\LWR@filenamenoblanks{#1}}
1031 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilename}}
```

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

```
1032 \LWR@htmlblockcomment{%
1033 \LWR@origtextttt{|Start file|
1034 \LWR@htmlsectionfilename{\LWR@thisfilename}}}%
1035 }
```

At the top of the starting file:

```
1036 \LWR@stoppars
1037
1038 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
1039
1040 \LWR@topnavigation
1041
1042 \LWR@html element{header}
1043
1044 \LWR@pagetop
1045
1046 \LWR@html elementend{header}
1047
1048 \ifcsvoid{thetitle}{}{\LWR@printthetitle}
1049
1050 \LWR@sidetoc
1051
1052 \LWR@html elementclass{section}{textbody}
1053
1054 \LWR@stoppars
1055
1056 \setcounter{LWR@latestautopage}{\value{page}}
1057 }

1058 \end{warpHTML}
```

38 Sectioning

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

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



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

for HTML output: 1059 `\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.



```
1060 \newbool{LWR@mainmatter}
1061 \DeclareDocumentCommand{\mainmatter}{}{%
1062 \booltrue{LWR@mainmatter}%
1063 }
```

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

```
1064 \DeclareDocumentCommand{\frontmatter}{}{%
1065 \boolfalse{LWR@mainmatter}%
1066 }
```

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

```
1067 \DeclareDocumentCommand{\backmatter}{}{%
1068 \boolfalse{LWR@mainmatter}
1069 }
```

38.2 Sectioning support macros

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

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

```
1070 \newcommand*{\LWR@sectionnumber}[1]{%
1071 \InlineClass{sectionnumber}{#1} %
1072 }
```


autosec A tag used by the TOC and index.

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

Create an autosection tag.

```
1073 \newcommand*{\LWR@createautosec}[1]{%
1074 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
1075 }
```


\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.

```
1076 \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.

```
1077 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
1078 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
1079 \LWR@pushoneclose{#1}{#2}%
1080 }
```

Ctrl **\LWR@prevFileDepth** Remembers the previous **\LWR@FileDepth**.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
1081 \newcounter{\LWR@prevFileDepth}
1082 \setcounter{\LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\LWR@section * [(<TOC name>)] {<name>} {<sectiontype>}`

The common actions for the high-level sectioning commands.

```
1083 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
1084 \LWR@traceinfo{\LWR@section}%
1085 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
1086 \global\boolfalse{\LWR@minipagethispar}%
```

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

```
1087 \LWR@orignewpage%
1088
```

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

```
1089 \LWR@traceinfo{\LWR@section: testing about to start a new HTML file}%
1090 \IfBooleanTF{#1}{}{% not starred
1091 \ifthenelse{%
1092 \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{FileDepth}}%
1093 \AND%
1094 \(%
1095 \NOT\boolean{CombineHigherDepths}\OR%
1096 \cnttest{\csuse{\LWR@depth#4}}{<=}{\value{\LWR@prevFileDepth}}%
1097 \)%
1098 }%
```

If so: start a new HTML file:

```
1099 {% new file
1100 \LWR@traceinfo{\LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
1101 \IfNoValueTF{#2}%
```

If no optional entry

```
1102 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
1103 {\LWR@newhtmlfile{#2}}%
1104 }% new file
```

Else: No new HTML file:

```
1105 {}% not new file
1106 {}% not starred
```

Remember this section's name for \nameref:

```
1107 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
1108 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
```

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

```
1109
1110 \LWR@htmlcomment{Opening #4 ‘‘#3’’{}}
1111
```

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

```
1112 \ifthenelse{%
1113 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
1114 }%
1115 {\LWR@startpars}
1116 {}
```

Create the opening tag with an autosec:

```
1117 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

If not starred, step counter and add to TOC:

```
1118 \IfBooleanTF{#1}%
1119 {}% starred
1120 {}% not starred
```

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

```
1121 \ifthenelse{%
1122 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1123 }%
1124 {}% if secnumdepth
```

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

```
1125 \LWR@traceinfo{LWR@section: about to test main matter}%
1126 \ifbool{LWR@mainmatter}%
1127 {%
1128 \LWR@traceinfo{LWR@section: yes mainmatter}%
1129 \refstepcounter{#4}%
```


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

```
1130 \LWR@traceinfo{LWR@section: about to addcontentsline}%
1131 \addcontentsline{toc}{#4}%
1132 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1133 \IfValueTF{#2}{#2}{#3}%
1134 \LWR@traceinfo{LWR@section: finished addcontentsline}%
1135 }% end of if main matter
```

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

```
1136 {% not main matter
1137 \LWR@traceinfo{LWR@section: no main matter}%
1138 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1139 }% end of not main matter
1140 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
1141 {%
1142 \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
1143 }%
```

For part, print the section type:

```
1144 \ifbool{LWR@mainmatter}%
1145 {%
1146 \ifthenelse{%
1147 \(\cnttest{\csuse{LWR@depth#4}}{<=}%
1148 {\value{secnumdepth}}\)\ AND
1149 \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)
1150 }%
1151 {\csuse{#4name}~{}}%
1152 {}%
```

Print the section number:

```
1153 \LWR@traceinfo{LWR@section: about to print section number}%
1154 \ifthenelse{%
1155 \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
1156 }%
1157 {\protect\LWR@sectionnumber{\csuse{the#4}}}%
1158 {}%
1159 \LWR@traceinfo{LWR@section: finished print section number}%
1160 }{}%
1161 }% end of not starred
```

Print the section name:

1162 #3

close the heading tag, such as /H2

1163 \LWR@htmltag{\csuse{LWR@tag#4end}}%

Generate a L^AT_EX label:

1164 \label{autopage-\thepage}%

Start paragraph handing unless is an inline paragraph or subparagraph:

1165 \ifthenelse{%

1166 \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}}%

1167 {\LWR@startpars}}%

1168 {}%

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.

1169 \IfBooleanTF{#1}{}{% not starred

1170 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%

1171 }% not starred

1172 \LWR@traceinfo{LWR@section: done}%

1173 }

38.3 \section and friends

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

1174 \@ifundefined{chapter}

1175 {}

1176 {%

1177 \DeclareDocumentCommand{\part}{s o m}{%

1178 \LWR@stoppars%

1179

1180 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}}%

1181

1182 \LWR@section{#1}{#2}{#3}{part}}%

1183 }

1184 }

`\chapter * [⟨TOC name⟩] {⟨name⟩}`

```

1185 \@ifundefined{chapter}
1186 {}
1187 {%
1188 \DeclareDocumentCommand{\chapter}{s o m}{%
1189 \LWR@traceinfo{chapter #3}%
1190 \LWR@stoppars%
1191 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
1192
1193 \LWR@traceinfo{chapter: about to LWR@section}%
1194 \LWR@section{#1}{#2}{#3}{chapter}%
1195 \LWR@traceinfo{chapter: done}%
1196 }
1197 }

```

`\section * [⟨TOC name⟩] {⟨name⟩}`

```

1198 \DeclareDocumentCommand{\section}{s o m}{%
1199 \LWR@stoppars%
1200
1201 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
1202
1203 \LWR@section{#1}{#2}{#3}{section}%
1204 }

```

`\subsection * [⟨TOC name⟩] {⟨name⟩}`

```

1205 \DeclareDocumentCommand{\subsection}{s o m}{%
1206 \LWR@stoppars%
1207
1208 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
1209
1210 \LWR@section{#1}{#2}{#3}{subsection}%
1211 }

```

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

```

1212 \DeclareDocumentCommand{\subsubsection}{s o m}{%
1213 \LWR@stoppars%
1214
1215 \LWR@startnewdepth{\LWR@depthsubsubsection}%
1216 {\LWR@printclosesubsubsection}%
1217
1218 \LWR@section{#1}{#2}{#3}{subsubsection}%
1219 }

```

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

```
1220 \DeclareDocumentCommand{\paragraph}{s o m}{%
1221 \LWR@stoppars%
1222
1223 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
1224
1225 \LWR@section{#1}{#2}{#3}{paragraph}%
1226 }
```

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

```
1227 \DeclareDocumentCommand{\subparagraph}{s o m}{%
1228 \LWR@stoppars%
1229
1230 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
1231
1232 \LWR@section{#1}{#2}{#3}{subparagraph}%
1233 }

1234 \end{warpHTML}
```

39 Starting a new file

for HTML output: 1235 \begin{warpHTML}

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

```
1236 \newcommand*{\MetaLanguage}{en-US}
```

`\LWR@filestart` {*title__suffix*}

Creates the opening HTML tags.

```
1237 \newcommand*{\LWR@filestart}[1]{
```

Locally temporarily disable direct-formatting commands:

```
1238 \begingroup
1239 \renewcommand{\textit}[1]{##1}% not used in filenames
1240 \renewcommand{\textsc}[1]{##1}
1241 \renewcommand{\textsl}[1]{##1}
1242 \renewcommand{\textbf}[1]{##1}
1243 \renewcommand{\texttt}[1]{##1}
```

```

1244 \renewcommand{\textsf}[1]{##1}
1245 \renewcommand{\textrm}[1]{##1}
1246 \renewcommand{\textsuperscript}[1]{##1}
1247 \renewcommand{\textsubscript}[1]{##1}
1248 \renewcommand*{\HTMLUnicode}[1]{ }
1249 \renewcommand*{\HTMLentity}[1]{ }
1250 \RenewDocumentCommand{\LWR@htmlspanclass}{m o +m}{##3}
1251 \DeclareExpandableDocumentCommand{\InlineClass}{m o m}{##3}

```

Create the page's HTML header:

```
1252 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
1253 \LWR@htmltag{html lang="\MetaLanguage"{}\LWR@orignewline
```

Start of the meta data:

```
1254 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
1255 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

lwarp is the generator:

```

1256 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\%
1257     \LWR@orignewline

```

If there is a description, add it now:

```

1258 \ifdefempty{\LWR@currentHTMLdescription}{\LWR@orignewline}{}{\%
1259 \LWR@htmltag{
1260 meta name="description" content="\LWR@currentHTMLdescription" /\%
1261     \LWR@orignewline
1262 }%

```

Mobile-friendly viewport:

```

1263 \LWR@htmltag{meta name="viewport" %
1264 content="width=device-width, initial-scale=1.0" /\%
1265     \LWR@orignewline

```

IE patch:

```

1266 \LWR@htmltag{!{-}{-}[if lt IE 9]\LWR@orignewline
1267 \LWR@htmltag{

```

```

1268 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
1269 \LWR@htmltag{/script}\LWR@orignewline
1270 \LWR@htmltag{![endif]{}{-}}\LWR@orignewline

```

The page's title:

```

1271 \ifcvoid{thetitle}{}{%
1272 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
1273 }%

```

The page's stylesheet:

```

1274 \LWR@htmltag{%
1275 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
1276 \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.

```

1277 \ifbool{mathjax}%
1278 {%
1279 \boolfalse{LWR@verbtags}
1280 \VerbatimInput{lwarp_mathjax.txt}%
1281 \booltrue{LWR@verbtags}
1282 \LWR@stoppars
1283 }% end of mathjax
1284 {}%

```

End of the header:

```

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

```

Start of the body:

```

1286 \LWR@htmltag{body}\LWR@orignewline
1287 \endgroup
1288 }

```

```

1289 \end{warpHTML}

```

40 Starting HTML output

for HTML output: 1290 \begin{warpHTML}

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

```
1291 \newcommand*{\LWR@LwarpStart}
1292 {%
```

Expand and detokenize `\HomeHTMLFileName` and `\HTMLFileName`:

```
1293 \edef\LWR@strresult{\HomeHTMLFileName}
1294 \edef\HomeHTMLFileName{\detokenize\expandafter{\LWR@strresult}}
1295 \edef\LWR@strresult{\HTMLFileName}
1296 \edef\HTMLFileName{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn:

```
1297 \LWR@origonecolumn%
```

Reduce chance of line overflow in verbatim environments:

```
1298 \LWR@origscriptsize%
```

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

```
1299 \LWR@origraggedright%
1300 \let\\\LWR@endoffline%
```

Spread the lines for `pdftotext` to read them well:

```
1301 \linespread{1.3}%
```

For `pdftotext` to reliably identify paragraph splits:

```
1302 \setlength{\parindent}{0pt}
1303 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
1304 \immediate\openout\LWR@file=lateximages.txt
```

Removes space after the caption in the HTML:

```
1305 \setlength{\belowcaptionskip}{-3ex}
```

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

```
1306 \renewcommand{\ps@plain}{}
```

`\centering` Not used in the HTML environment:
`\raggedleft`
`\raggedright`

```

1307 \renewcommand*{\centering}{}
1308 \renewcommand*{\raggedleft}{}
1309 \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:

```

1310 \let\LWR@origtabular\tabular
1311 \let\LWR@origendtabular\endtabular
1312 \let\tabular\LWR@tabular
1313 \let\endtabular\endLWR@tabular

```

Float captions:

```

1314 \let\LWR@origcaption\caption

```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

[Label in HTML](#)

```

1315 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

1316 \let\TeX\LWR@TeX
1317 \let\LaTeX\LWR@LaTeX
1318 \let\LuaTeX\LWR@LuaTeX
1319 \let\LuaLaTeX\LWR@LuaLaTeX
1320 \let\XeTeX\LWR@XeTeX
1321 \let\XeLaTeX\LWR@XeLaTeX
1322 \let\ConTeXt\LWR@ConTeXt

```

Graphics:

```

1323 \let\textcolor\LWR@textcolor
1324 \let\pagecolor\LWR@pagecolor
1325 \let\colorbox\LWR@colorbox
1326 \let\fcolorbox\LWR@fcolorbox
1327 \let\rotatebox\LWR@rotatebox
1328 \let\scalebox\LWR@scalebox
1329 \let\reflectbox\LWR@reflectbox

```

Not yet started any paragraph handling:

```

1330 \global\boolfalse{LWR@doingapar}
1331 \global\boolfalse{LWR@doingstartpars}

```


Start a new HTML file and a header:

```
1332 \LWR@filestart{}
1333 \LWR@htmltag{header}\LWR@orignewline
1334 \LWR@startpars
1335 \LWR@firstpagetop
1336 \LWR@stoppars
1337 \LWR@htmltag{/header}\LWR@orignewline
1338 \LWR@htmltag{section class="textbody"{} }
1339 \LWR@origpagestyle{empty}
```

Document and page settings:

```
1340 \mainmatter
1341 \LWR@origpagenumbering{arabic}
```

Set default titlepage thanks footnote marks. See section 42.6.

```
1342 \if@titlepage
1343   \thanksmarkseries{arabic}
1344 \else
1345   \thanksmarkseries{fnsymbol}
1346 \fi
```

Initial default patch for fancyvrb:

```
1347 \fvset{frame=none}%
```

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

```
1348 \catcode'\&=\active
```

Allow HTML paragraphs to begin:

```
1349 \LWR@startpars
1350 }
```

```
1351 \end{warpHTML}
```

41 Ending HTML output

for HTML output: 1352 \begin{warpHTML}

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

```

1353 \newcommand*{\LWR@requesttoc}[2]{%
1354 \ifbool{#1}
1355 {
1356 \expandafter\newwrite\csuse{tf@#2}
1357 \immediate\openout \csuse{tf@#2} \jobname.#2\relax
1358 }{}
1359 }

```

\LWR@LwarpEnd Final stop of all HTML output:

```

1360 \newcommand*{\LWR@LwarpEnd}
1361 {
1362 \LWR@stoppars
1363 \LWR@closeprevious{\LWR@depthfinished}

```

At the bottom of the ending file:

Close the textbody:

```

1364 \LWR@htmlelementclassend{section}{textbody}

```

Print any pending footnotes:

```

1365 \LWR@printpendingnotes

```

Create the footer:

```

1366 \LWR@htmlelement{footer}
1367
1368 \LWR@pagebottom
1369
1370 \LWR@htmlelementend{footer}

```

No bottom navigation if are finishing the home page:

Presumably has a table-of-contents printed.

```

1371 \ifnumcomp{\value{\LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}

```

```

1372 \LWR@stoppars% final stop of all paragraphs

```

Finish the HTML file:

```

1373 \LWR@htmltag{/body}\LWR@orignewline
1374 \LWR@htmltag{/html}\LWR@orignewline

```

Seems to be required sometimes:

```

1375 \LWR@orignewpage

```

For `lateximage` commands:

```
1376 \immediate\closeout\LWR@file
1377 }

1378 \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.

<code>\published</code> <code>\title</code> <code>\subtitle</code> <code>\author</code> <code>\date</code>	<code>\@title</code> , <code>\@subtitle</code> , <code>\@author</code> , etc. store the values as originally assigned, including any <code>\thanks</code> , <code>\and</code> , or <code>\affiliation</code> . These are low-level macros intended to be used by other macros only inside a <code>titlepage</code> or <code>titlingpage</code> , and are used by <code>\maketitle</code> . The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\printpublished</code> <code>\printtitle</code> <code>\printsubtitle</code> <code>\printauthor</code> <code>\printdate</code>	<code>\printtitle</code> , <code>\printsubtitle</code> , etc. are user-level macros intended to be used in <code>titlepage</code> and <code>titlingpage</code> environments in cases where <code>\maketitle</code> is not desired. These commands preserve the <code>\thanks</code> , etc., and should not be used in the main text. The author is printed inside a single-column table, which becomes multiple single-column tables if multiples authors are included.
<code>\thepublished</code> <code>\thetitle</code> <code>\thesubtitle</code> <code>\theauthor</code> <code>\thedate</code>	<code>\thetitle</code> , <code>\thesubtitle</code> , <code>\theauthor</code> , etc. are user-level sanitized versions which have removed the <code>\thanks</code> and <code>\affiliation</code> , and <code>\and</code> is changed for inline text usage. The author is printed inline without <code>\affiliation</code> or <code>\thanks</code> , with <code>\and</code> placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:
<code>\SetPageBottom</code>	<code>{\langle text \rangle}</code>

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

`\author` `{\langle author \rangle}` While using `\maketitle`, the author is treated as a single-column table

`\and`

and the `\and` feature finishes the current table then starts a new one for the next author. Each author thus is placed into its own table, and an affiliation may be placed on its own line such as

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

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

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

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

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

```
for HTML output: 1379 \begin{warpHTML}
                  1380 \newrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{#1}}
                  1381 \end{warpHTML}
```

```
for PRINT output: 1382 \begin{warpprint}
                  1383 \newrobustcmd{\affiliation}[1]{\\ \textsc{small#1}}
                  1384 \end{warpprint}
```

The following are based on the original titling code:

```
for HTML & PRINT: 1385 \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.

```
1386 \renewcommand{\author}[1]{%
1387 \gdef\@author{#1}
1388 \begingroup
1389   \renewcommand{\thanks}[1]{ }
1390   \renewcommand{\and}{\unskip, }
1391   \renewcommand{\thanksmark}[1]{ }
1392   \renewcommand{\thanksgap}[1]{ }
1393   \renewcommand{\affiliation}[1]{ }
```

```

1394 \protected@xdef\theauthor{#1}
1395 \endgroup

\published {(text)}

1396 \newcommand{\published}[1]{%
1397 \gdef\@published{#1}
1398 \begingroup
1399 \renewcommand{\thanks}[1]{ }
1400 \renewcommand{\thanksmark}[1]{ }
1401 \renewcommand{\thanksgap}[1]{ }
1402 \protected@xdef\thepublished{#1}
1403 \endgroup
1404 }
1405 \newcommand{\@published}{}
1406 \newcommand{\thepublished}{}

\subtitle {(text)}

1407 \newcommand{\subtitle}[1]{%
1408 \gdef\@subtitle{#1}
1409 \begingroup
1410 \renewcommand{\thanks}[1]{ }
1411 \renewcommand{\thanksmark}[1]{ }
1412 \renewcommand{\thanksgap}[1]{ }
1413 \protected@xdef\thesubtitle{#1}
1414 \endgroup
1415 }
1416 \newcommand{\@subtitle}{}
1417 \newcommand{\thesubtitle}{}

1418 \end{warpall}

```

42.2 Changes to HTML titlepage and titlingpage

for HTML output: 1419 \begin{warpHTML}

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

```

1420 \renewenvironment*{titlepage}
1421 {\BlockClass{titlepage}\LWR@subminipage}
1422 {\LWR@endsubminipage\endBlockClass}

```

Env titlingpage

```
1423 \renewenvironment*{titlingpage}
1424 {%
```

Start an HTML titlepage div:

```
1425 \begin{titlepage}
```

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

```
1426 \LWR@maketitlesetup
1427 \let\maketitle\LWR@titlingmaketitle
1428 }
1429 {
```

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

```
1430 \end{titlepage}
```

Reset the footnote counter:

```
1431 \@bscontmark
1432 }
```

```
1433 \end{warpHTML}
```

for HTML & PRINT: 1434 `\begin{warpall}`

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

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

```
1435 \newcommand*{\printthanks}{\@thanks}
```

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

```
1436 \AtEndEnvironment{titlingpage}{\@bscontmark}
```

```
1437 \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: 1438 `\begin{warpHTML}`

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

```

1439 \newcommand{\prepublished}[1]{%
1440 \def\@bsprepublished{\BlockClass{published}#1}%
1441 }
1442
1443 \newcommand{\postpublished}[1]{%
1444 \def\@bspostpublished{#1\endBlockClass}%
1445 }
1446
1447 \renewcommand{\pretitle}[1]{%
1448 \def\@bspretitle{#1\LWR@stoppars\LWR@htmltag{h1}}}%
1449 }
1450
1451 \renewcommand{\posttitle}[1]{%
1452 \def\@bsposttitle{\LWR@htmltag{/h1}\LWR@startpars#1}%
1453 }
1454
1455 \newcommand{\presubtitle}[1]{%
1456 \def\@bspresubtitle{\BlockClass{subtitle}#1}%
1457 }
1458
1459 \newcommand{\postsubtitle}[1]{%
1460 \def\@bspostsubtitle{#1\endBlockClass}%
1461 }
1462
1463 \renewcommand{\preauthor}[1]{%
1464 \def\@bspreauthor{\BlockClass{author}#1}%
1465 }
1466
1467 \renewcommand{\postauthor}[1]{%
1468 \def\@bspostauthor{#1\endBlockClass}%
1469 }
1470
1471 \renewcommand{\predate}[1]{%
1472 \def\@bspredate{#1\BlockClass{titledate}}}%
1473 }
1474
1475 \renewcommand{\postdate}[1]{%
1476 \def\@bspostdate{\endBlockClass#1}%
1477 }
1478
1479 \prepublished{\begin{center}}
1480 \postpublished{\par\end{center}}
1481
1482 \pretitle{\begin{center}}
1483 \posttitle{\par\end{center}}
1484
1485 \presubtitle{\begin{center}}
1486 \postsubtitle{\par\end{center}}

```



```

1487
1488 \preauthor{\begin{center}%
1489 \begin{tabular}{t}{c}%
1490 }
1491 \postauthor{\end{tabular}\par\end{center}}
1492
1493 \predate{\begin{center}}
1494 \postdate{\par\end{center}}

```

\printpublished

```

1495 \newcommand*{\printpublished}{
1496 \ifthenelse{\equal{\thepublished}{}}{
1497 {}
1498 {
1499 \begin{BlockClass}{published}
1500 \@published
1501 \end{BlockClass}
1502 }
1503 }

```

\printtitle

```

1504 \newcommand*{\printtitle}
1505 {
1506 \LWR@stoppars
1507 \LWR@htmltag{h1}%
1508 \@title%
1509 \LWR@htmltag{/h1}
1510 \LWR@startpars
1511 }

```

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

```

1512 \newcommand*{\LWR@printthetitle}
1513 {
1514 \LWR@stoppars
1515 \LWR@htmltag{h1}%
1516 \thetitle%
1517 \LWR@htmltag{/h1}
1518 \LWR@startpars
1519 }

```

\printsubtitle

```

1520 \newcommand*{\printsubtitle}{

```

```

1521 \ifthenelse{\equal{\thesubtitle}{}}
1522 {}
1523 {
1524 \begin{BlockClass}{subtitle}
1525 \@subtitle
1526 \end{BlockClass}
1527 }
1528 }

```

`\printauthor`

```

1529 \newcommand*\printauthor{%
1530 \begin{BlockClass}{author}
1531 \begin{tabular}{c}\@author\end{tabular}
1532 \end{BlockClass}
1533 }

```

`\printdate`

```

1534 \newcommand*\printdate{%
1535 \begin{BlockClass}{titledate}
1536 \@date
1537 \end{BlockClass}
1538 }

```

```

1539 \end{warpHTML}

```

42.4 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

for PRINT output: 1540 `\begin{warpprint}`

`\printpublished`

```

1541 \newcommand*\printpublished{{\Large\scshape\@published}}

```

`\printtitle`

```

1542 \newcommand*\printtitle{{\Huge\@title}}

```

`\printssubtitle`

```

1543 \newcommand*\printssubtitle{{\Large\itshape\@subtitle\bigskip}}

```

`\printauthor`

```
1544 \newcommand*{\printauthor}
1545   {\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

`\printdate`

```
1546 \newcommand*{\printdate}{\small\textit{\@date}}
```

42.5 `\maketitle` for print output

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

```
1547 \providecommand{\maketitle}{}
1548 \if@titlepage
1549   \renewcommand{\maketitle}{\begin{titlepage}%
1550     \let\footnotesize\small
1551     \let\footnoterule\relax
1552     \let \footnote \thanks
1553     \bsmarkseries
1554     \def\@makefnmark{\rlap{\@textsuperscript{%
1555       \normalfont\bsthanksheadpre \tamark \bsthanksheadpost}}}%
1556     \long\def\@makefntext##1{\makethanksmark ##1}
1557     \null\vfil
1558     \vskip 60\p@
1559     \vspace*{\droptitle}
1560     \maketitlehooka
1561     \ifcsempy{\@published}
1562   {}
1563   {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1564   {\@bsprettitle \@title \@bsposttitle}
1565   \ifcsempy{\@subtitle}
1566   {}
1567   {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1568   \maketitlehookb
1569   {\@bspreauthor \@author \@bspostauthor}
1570   \maketitlehookc
1571   {\@bspredate \@date \@bspostdate}
1572   \maketitlehookd
1573   \par
1574   \@thanks
1575   \vfil\null
1576   \end{titlepage}%
1577   \@bscontmark % \setcounter{footnote}{0}%
1578   %%% \bsmttitleempty
1579 } % end titlepage defs
```

```

1580 \else
1581   \renewcommand{\maketitle}{\par
1582     \begin{group}
1583       \@bsmarkseries
1584       \def\@makefnmark{\rlap{\@textsuperscript{%
1585         \normalfont\@bsthanksheadpre \tamark \@bsthanksheadpost}}}%
1586       \long\def\@makefntext##1{\makethanksmark ##1}
1587       \if@twocolumn
1588         \ifnum \col@number=\@ne
1589           \@maketitle
1590         \else
1591           \twocolumn[\@maketitle]%
1592         \fi
1593       \else
1594         \newpage
1595         \global\@topnum\z@
1596         \@maketitle
1597       \fi
1598       \thispagestyle{plain}\@thanks
1599     \end{group}
1600     \@bscontmark % \setcounter{footnote}{0}%
1601     %% \@bsmtitleempty
1602   } % end non-titlepage
1603
1604   \def\@maketitle{%
1605     \newpage
1606     \null
1607     \vskip 2em%
1608     \vspace*{\droptitle}
1609     \maketitlehooka
1610     \ifcseempty{@published}
1611   {}
1612   {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1613   {\@bspretitle \@title \@bsposttitle}
1614   \ifcseempty{@subtitle}
1615   {}
1616   {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1617   \maketitlehookb
1618   {\@bspreauthor \@author \@bspostauthor}
1619   \maketitlehookc
1620   {\@bspredate \@date \@bspostdate}
1621   \maketitlehookd
1622   \par
1623   \vskip 1.5em}
1624 \fi
1625
1626 \providecommand{\maketitlehookaa}{}
1627
1628 \providecommand{\maketitlehookaaa}{}
1629

```

```

1630 \newcommand{\prepublished}[1]{%
1631 \def\@bsprepublished{#1}%
1632 }
1633
1634 \newcommand{\postpublished}[1]{%
1635 \def\@bspostpublished{#1}%
1636 }
1637
1638 \newcommand{\presubtitle}[1]{%
1639 \def\@bspresubtitle{#1}%
1640 }

```

`\presubtitle` Hook after printing the subtitle.

```

1641 \newcommand{\postsubtitle}[1]{%
1642 \def\@bspostsubtitle{#1}%
1643 }

```

Initial settings:

```

1644 \if@titlepage
1645 \prepublished{
1646 \vspace*{-\baselineskip}\vspace*{-\medskipamount}\vspace*{-2em}
1647 \begin{center}}
1648 \postpublished{\par\end{center}\vskip 2em}
1649
1650 \presubtitle{\unskip\begin{center}\unskip}
1651 \postsubtitle{\par\end{center}\vskip 2em}
1652 \else
1653 \prepublished{\begin{center}}
1654 \postpublished{\par\end{center}\vskip 0.5em}
1655
1656 \presubtitle{\begin{center}\unskip}
1657 \postsubtitle{\par\end{center}\vskip 0.5em}
1658 \fi
1659 \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: 1660 `\begin{warpHTML}`

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

1661 `\newcommand*{\LWR@maketitlesetup}{%`

Select which kind of footnote marks to use:

1662 `\@bsmarkseries`

1663 `\@mpbsmarkseries`

Redefine the footnote mark:

1664 `\def\@makefnmark{\thefootnote}%`

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

Redefine the footnote text:

1665 `\long\def\@makefntext##1{%`

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

1666 `\makethanksmark \LWR@origspace{1in}`

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

Print the text and a closing paragraph tag:

1667 `##1\LWR@htmltagc{/\LWR@tagregularparagraph}%`

1668 `}%`

1669 `}`

`\@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`.

1670 `\def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or`

1671 `\mathsection\or \mathparagraph\or \text{\HTMLunicode{2016}}\or`

1672 `**\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.

```
1673 \renewcommand*\maketitle}{%
```

An HTML titlepage div is used for all classes.

```
1674 \begin{titlepage}
```

Set up special patches:

```
1675 \LWR@maketitlesetup
```

Typeset the title, etc:

```
1676 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
1677 \@thanks
```

Close the HTML titlepage div:

```
1678 \end{titlepage}
```

Reset the footnote counter:

```
1679 \@bscontmark
1680 }
```

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

```
1681 \DeclareDocumentCommand{\@maketitle}{}{%
1682 \maketitlehooka
1683 \ifcempty{@published}
1684 {}
1685 {\@bsprepublished \@published \@bspostpublished}\maketitlehookaa}
1686 {\@bsprettitle \@title \@bsposttitle}
1687 \ifcempty{@subtitle}
1688 {}
1689 {\maketitlehookaaa{\@bspresubtitle \@subtitle \@bspostsubtitle}}
1690 \maketitlehookb
1691 {\@bspreauthor \@author \@bspostauthor}
1692 \maketitlehookc
1693 {\@bspredate \@date \@bspostdate}
1694 \maketitlehookd
1695 }
```

```
1696 \providecommand{\maketitlehookaa}{}
1697 \providecommand{\maketitlehookaaa}{}

```

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

```
1698 \newcommand*{\LWR@titlingmaketitle}{%

```

Typeset the title, etc:

```
1699 \@maketitle

```

Immediately generate any `\thanks` footnotes:

```
1700 \@thanks
1701 }

```

`\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.

```
1702 \let\LWR@origthanksmarkseries\thanksmarkseries
1703 \renewcommand{\thanksmarkseries}[1]{%
1704 \def\@mpbsmarkseries{%
1705 \renewcommand*{\thempfootnote}{\@nameuse{#1}{mpfootnote}}}
1706 \LWR@origthanksmarkseries{#1}
1707 }

```

```
1708 \end{warpHTML}

```

43 Abstract

for HTML output: 1709 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the `babel` package.

```
1710 \providecommand*{\abstractname}{Abstract}

```

Env `abstract`


```

1711 \DeclareDocumentEnvironment{abstract}{}
1712 {
1713 \BlockClass{abstract}
1714 \BlockClassSingle{abstracttitle}{\abstractname}
1715 }
1716 {
1717 \endBlockClass
1718 }

1719 \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:

```

1720 \begin{warpHTML}
1721 \newcommand{\attribution}[1]{%
1722 \InlineClass{attribution}{--\,#1}}% emdash
1723 \end{warpHTML}

```

for PRINT output:

```

1724 \begin{warpprint}
1725 \newcommand{\attribution}[1]{\textsc{---\,#1}}
1726 \end{warpprint}

```

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

for HTML output:

```

1727 \begin{warpHTML}
1728 \newcommand{\citetitle}[1]{%
1729 \InlineClass{citetitle}{--\,#1}}% emdash
1730 \end{warpHTML}

```

for PRINT output:

```

1731 \begin{warpprint}
1732 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
1733 \end{warpprint}

```

44.2 Quotes, quotations

for HTML output:

```

1734 \begin{warpHTML}

```

Env quote

```

1735 \renewenvironment*{quote}
1736 {\LWR@htmlblocktag{blockquote}}
1737 {\LWR@htmlblocktag{/blockquote}}
1738
1739 \renewenvironment*{quotation}
1740 {\LWR@htmlblocktag{blockquotation}}
1741 {\LWR@htmlblocktag{/blockquotation}}

1742 \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. `lwrap` 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: 1743 `\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.

```
1744 \newlength{\HTMLvleftskip}
1745 \setlength{\HTMLvleftskip}{1em}
```

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

```
1746 \newlength{\HTMLleftmargini}
1747 \setlength{\HTMLleftmargini}{4.5em}

1748 \end{warpall}
```

for HTML output: 1749 `\begin{warpHTML}`

Env `verse` The `verse` environment will be placed inside a HTML `pre`.

```
1750 \AfterEndPreamble{
```

At the beginning of the `verse` environment:

```
1751 \AtBeginEnvironment{verse}
1752 {%
```

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 1753 \ifdef{\vleftskip}{%
1754 \setlength{\vleftskip}{\HTMLvleftskip}
1755 \setlength{\leftmargini}{\HTMLleftmargini}
1756 }{}
1757 \LWR@atbeginverbatim{verse}
1758 \unskip\vspace{-\baselineskip}
1759 }
```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```
1760 \AfterEndEnvironment{verse}{
1761 \unskip\vspace{-\baselineskip}
1762 \LWR@afterendverbatim
1763 }
```

Patch to place `poemtitle` inside an HTML span of class `poemtitle`:

```
1764 \ifdef{\poemtitle}{
1765 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
```

```

1766 \vspace{\beforepoemtitleskip}%
1767 {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
1768 \vspace{\afterpoemtitleskip}%
1769 }
1770 }{}
1771
1772 }
1773 \end{warpHTML}

```

45 Verbatim

for HTML output: 1774 \begin{warpHTML}

Env verbatim

```

1775 \AfterEndPreamble{
1776 \AtBeginEnvironment{verbatim}{\LWR@atbeginverbatim{verbatim}\unskip\vspace*{-\baselineskip}}
1777 \AfterEndEnvironment{verbatim}{\unskip\vspace*{-\baselineskip}\LWR@afterendverbatim}
1778 }

1779 \end{warpHTML}

```

46 Fancyvrb

for HTML & PRINT: 1780 \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.

```

1781 \newlength{\VerbatimHTMLWidth}
1782 \setlength{\VerbatimHTMLWidth}{4in}
1783 \end{warpall}

```

for HTML output: 1784 \begin{warpHTML}

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

```

1785 \newbool{LWR@verbtags}
1786 \booltrue{LWR@verbtags}

```

For `\VerbatimFootnotes`:

```
1787 \renewcommand{\VerbatimFootnotes}{
1788 \PackageError{lwarp}
1789 {Verbatim footnotes are not yet supported by lwarp.}
1790 {This may be improved when lwarp uses LaTeX footnotes instead of pagenotes.}
1791 }
```

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

Encloses a verbatim environment with the given CSS class.

```
1792 \newcommand*{\LWR@atbeginverbatim}[1]
1793 {%
```

Avoid excessive space between lines:

```
1794 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
1795 \LWR@stoppars%
```

Create a new `pre` of the given class:

```
1796 \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.

```
1797 \LWR@origttfamily%
```

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

```
1798 \let\hspace\LWR@orighspace%
1799 }
```

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

```
1800 \newcommand*{\LWR@afterendverbatim}{%
```

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

```
1801 \unskip%
```

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

```

1802 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
1803
1804 }{}%

```

Resume regular paragraph handling:

```

1805 \LWR@startpars%
1806 }

```

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

```

1807 \newcommand*{\LWR@Verbatimclass}{fancyvrb}

```

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

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

```

1808 \NewDocumentEnvironment{VerbatimClass}{m 0{}}
1809 {%
1810 \renewcommand*{\LWR@Verbatimclass}{#1}%
1811 \LWR@origVerbatim[#2]%
1812 }
1813 {\endVerbatim}

```

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

```

1814 \AfterEndPreamble{

```

Remember the original definition of `Verbatim`:

```

1815 \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.

```

1816 \renewcommand*{\Verbatim}{%
1817 \renewcommand*{\LWR@Verbatimclass}{fancyvrb}%
1818 \LWR@origVerbatim%
1819 }

```

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.

```

1820 \newcommand*{\LWR@fvstartnone}{%
1821 \LWR@traceinfo{fvstartnone}%
1822 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{div class="\LWR@Verbatimclass"}}}{}%
1823 \hbox to\z@{\LWR@atbeginverbatim{verbatim}}}%
1824 }
1825
1826 \newcommand*{\LWR@fvendnone}{%
1827 \LWR@traceinfo{fvendnone}%
1828 \hbox to\z@{\LWR@afterendverbatim}%
1829 \ifbool{LWR@verbtags}{\hbox to\z@{\LWR@htmltagc{/div}}}{}%
1830 }
1831
1832 \newcommand*{\LWR@fvstartsingle}{%
1833 \LWR@traceinfo{fvstartsingle}%
1834 \LWR@fvstartnone%
1835 \FV@BeginListFrame@Single%
1836 }
1837
1838 \newcommand*{\LWR@fvendsingle}{%
1839 \LWR@traceinfo{fvendsingle}%
1840 \FV@endListFrame@Single%
1841 \LWR@fvendnone%
1842 }
1843
1844 \newcommand*{\LWR@fvstartline}{%
1845 \LWR@traceinfo{fvstartline}%
1846 \LWR@fvstartnone%
1847 \FV@BeginListFrame@Lines%
1848 }
1849
1850 \newcommand*{\LWR@fvendline}{%
1851 \LWR@traceinfo{fvendline}%
1852 \FV@endListFrame@Lines%
1853 \LWR@fvendnone%
1854 }

```

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

```

1855 \def\FV@Frame@none{%
1856 \let\FV@BeginListFrame\LWR@fvstartnone%
1857 \let\FV@LeftListFrame\relax%
1858 \let\FV@RightListFrame\relax%
1859 \let\FV@endListFrame\LWR@fvendnone%
1860
1861 \def\FV@Frame@single{%
1862 \let\FV@BeginListFrame\LWR@fvstartsingle%
1863 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1864 \let\FV@RightListFrame\FV@RightListFrame@Single%

```

```

1865 \let\FV@EndListFrame\LWR@fvendsingle}
1866
1867 \def\FV@Frame@lines{%
1868 \let\FV@BeginListFrame\LWR@fvstartline%
1869 \let\FV@LeftListFrame\relax%
1870 \let\FV@RightListFrame\relax%
1871 \let\FV@EndListFrame\LWR@fvendline}
1872
1873 \def\FV@Frame@topline{%
1874 \let\FV@BeginListFrame\LWR@fvstartline%
1875 \let\FV@LeftListFrame\relax%
1876 \let\FV@RightListFrame\relax%
1877 \let\FV@EndListFrame\LWR@fvendnone}
1878
1879 \def\FV@Frame@bottomline{%
1880 \let\FV@BeginListFrame\LWR@fvstartnone%
1881 \let\FV@LeftListFrame\relax%
1882 \let\FV@RightListFrame\relax%
1883 \let\FV@EndListFrame\LWR@fvendline}
1884
1885 \def\FV@Frame@leftline{%
1886 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
1887 \ifx\FancyVerbFillColor\relax%
1888 \let\FV@FrameFillLine\relax%
1889 \else%
1890 \@tempdima\FV@FrameRule\relax%
1891 \multiply\@tempdima-\tw@%
1892 \edef\FV@FrameFillLine{%
1893 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
1894 \kern-\number\@tempdima sp}}%
1895 \fi%
1896 \let\FV@BeginListFrame\LWR@fvstartnone%
1897 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
1898 \let\FV@RightListFrame\relax%
1899 \let\FV@EndListFrame\LWR@fvendnone}

```

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

```

1900 \def\FV@SingleFrameLine#1{%
1901 \hbox to\z@{%
1902 % \kern\leftmargin
1903 \ifnum#1=\z@\relax
1904 \let\FV@Label\FV@LabelBegin
1905 \else
1906 \let\FV@Label\FV@LabelEnd
1907 \fi
1908 \ifx\FV@Label\relax
1909 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%

```



```

1910     \else
1911         \ifnum#1=\z@
1912 %         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
1913         \ifx\FV@LabelPositionTopLine\relax
1914 \else
1915         \LWR@htmltagc{div class="fancyvrblabel"}
1916 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
1917 \LWR@htmltagc{/div}
1918 \fi
1919     \else
1920 %         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
1921         \ifx\FV@LabelPositionBottomLine\relax
1922 \else
1923         \LWR@htmltagc{div class="fancyvrblabel"}
1924 \LWR@origtextrm{\FV@LabelEnd}
1925 \LWR@htmltagc{/div}
1926 \fi
1927     \fi
1928
1929     \fi
1930     \hss
1931 }
1932 }

```

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

```

1933 \def\FV@ListProcessLine#1{%
1934     \hbox to \hsize{%
1935 %         \kern\leftmargin
1936         \hbox to \VerbatimHTMLWidth {%
1937             \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
1938 \FV@LeftListNumber%
1939 %         \FV@LeftListFrame
1940         \FancyVerbFormatLine{#1}%
1941         \hss%
1942 %         \FV@RightListFrame
1943         \FV@RightListNumber%
1944 }%
1945         \hss% required to avoid underfull hboxes
1946 }
1947 }

```

Env BVerbatim

```

1948 \AtBeginEnvironment{BVerbatim}
1949 {
1950 \LWR@atbeginverbatim{bverbatim}
1951

```

```

1952 }
1953
1954 \AfterEndEnvironment{BVerbatim}
1955 {
1956 \leavevmode\par\vspace{-\baselineskip}
1957 \LWR@afterendverbatim
1958 }

```

Env **LVerbatim** No changes required.

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

```

1959 } % \AfterEndPreamble

```

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

No changes required.

```

1960 \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: 1961 \begin{warpHTML}

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

```

1962 \renewcommand{\@begintheorem}[2]{%
1963 \LWR@htmldivclass{theorem}
1964 \InlineClass{theoremlabel}{#1\ #2\ }
1965 \LWR@htmldivclass{theoremcontents}
1966 }

```

```

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

1967 \renewcommand{\@opargbegintheorem}[3]{%
1968 \BlockClass{theorem}
1969 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
1970 \BlockClass{theoremcontents}
1971 }

\@endtheorem


1972 \renewcommand*\@endtheorem{%
1973 \endBlockClass% theoremcontents
1974 \endBlockClass% theorem
1975 }

1976 \end{warpHTML}

```

48 Lists

If using `babel` with French, use

 **French** `\frenchbsetup{StandardLists=true}` 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 [\langle label\rangle]
```

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.
- `\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@futurenonospacelet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonospacelet`

```
2063 \def\LWR@futurenonospacelet#1{\def\LWR@cs{#1}%
2064 \afterassignment\LWR@fns lone\let\nexttoken= }
2065 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
2066 \def\LWR@fns ltwo{%
2067 \expandafter\ifx\LWR@cs\@sptoken\let\next=\@BTfns lthree%
2068 \else\let\next=\nexttoken\fi\next}
2069 \def\@BTfns lthree{\afterassignment\LWR@fns lone\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@futurenonospacelet\LWR@mynexttoken\LWR@tabledatacolumn tag
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 &

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 @ column contents:

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

Close paragraphs:

```
2093 \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
2094 \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell:

```
2095 \unskip\LWR@htmltag{/td}\LWR@orignewline%
2096 }% not skipping mrowcell
2097 }% not exiting tabular
2098 \global\boolfalse{LWR@skippingmrowcell}%
2099 }
```

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

```
2100 \newcounter{LWR@tabulardepth}
2101 \setcounter{LWR@tabulardepth}{0}
2102
```

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

```
2103 \def\LWR@origampmacro{&}
```

See below for why the group is used.

```
2104 \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.

```
2105 \catcode'\&=\active
2106
2107 \gdef&{%
2108 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{%
2109 {%
```

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

```
2110 \unskip%
2111 \LWR@closetabledatacell%
```

Move to the next column.

```
2112 \addtocounter{LWR@tablecolspos}{1}%
```

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

```
2113 \LWR@getmynexttoken%
2114 }%
```

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

```
2115 {\LWR@origampmacro}%
2116 }
2117 \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`

```
2118 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
2119 {%
2120 \LWR@closetabledatacell%
```

Finish the previous row:

```
2121 \LWR@htmltag{/tr}\LWR@orignewline
2122 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
2123 \global\boolfalse{\LWR@startedrow}
```

Additional setup:

```
2124 \global\boolfalse{\LWR@doinghline}%
2125 \global\boolfalse{\LWR@doingtbrule}%
2126 \LWR@clearmidrules%
```

Start at first column:

```
2127 \setcounter{\LWR@tablecolspos}{1}
```

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

```
2128 \LWR@getmynexttoken%
2129 }
```

49.5 Variables

```

2130 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
2131 \newcommand*{\LWR@pposition}{}
2132 \newcommand*{\LWR@pleft}{}
2133 \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.

```

2134 \newcommand*{\LWR@tablecolspec}{}

```

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

```

2135 \newcommand*{\LWR@strresult}{}

```

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

```

2136 \newcommand*{\LWR@origcolspec}{}

```

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

(This is not the total # columns.)

```

2137 \newcounter{LWR@tablecolwidth}

```

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

```

2138 \newcounter{LWR@tablecolspos}

```

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

```

2139 \newcounter{LWR@tabletotalcols}

```

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

```

2140 \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@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.

```
2141 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
2142 \LWR@traceinfo{at column}%
2143 \addtocounter{LWR@tablecolspos}{1}%
```

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

```
2144 \LWR@traceinfo{about to read the next token}%
2145 \expandarg%
2146 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]
2147 \fullexpandarg%
```

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

```
2148 \LWR@traceinfo{have now read the next token}%
2149 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
2150 {% left edge of the table:
2151 \LWR@traceinfo{at the left edge}%
2152 \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@strresult}%
2153 \LWR@traceinfo{finished with the def}%
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: \LWR@colatspec(\theLWR@tabletotalcols)!}%
2159 }%
2160 \let\LWR@strresult\relax%
2161 }
```

`\LWR@parsebeforecolumn` Handles >{text} columns.

```
2162 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
2163 \addtocounter{LWR@tablecolspos}{1}%
```

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

```
2164 \expandarg%
2165 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2166 \fullexpandarg%
```

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

```
2167 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@strresult}%
2168 \let\LWR@strresult\relax%
2169 }
```

\LWR@parseaftercolumn Handles <{text} columns.

```
2170 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
2171 \addtocounter{LWR@tablecolspos}{1}%
```

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

```
2172 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]
2173 \expandarg%
2174 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@strresult]%
2175 \fullexpandarg%
```

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

```
2176 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@strresult}%
2177 \let\LWR@strresult\relax%
2178 }
```

\LWR@parseskipcolumn Handles columns to skip, such as !.

```
2179 \newcommand*{\LWR@parseskipcolumn}{%
2180 }
```

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

\LWR@parsenormalcolumn {<thiscolumn>}

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

```
2181 \newcommand*{\LWR@parsenormalcolumn}[1]{%
2182 \appto\LWR@tablecolspec{#1}%
2183 \addtocounter{LWR@tabletotalcols}{1}%
2184 \addtocounter{LWR@tabletotalcolsnext}{1}%
2185 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{\relax}%
2186 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\relax}%
2187 }
```

Table 5: Tabular baseline

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

```
2187 \LWR@setexpparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{\relax}%
2188 }
```

49.8 Parsing ‘p’, ‘m’, or ‘b’ columns

```
\LWR@parsepcolumn {<thiscolumn>} the width will be ignored

2189 \newcommand*{\LWR@parsepcolumn}[1]{%

converts to the given column type

2190 \LWR@parsenormalcolumn{#1}%

skips the following width

2191 \addtocounter{LWR@tablecolspos}{1}%
2192 }
```

49.9 Parsing the column specifications


 HTML CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 5 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 6 for details.

Table 6 describes how each kind of column is converted to HTML.

```
\LWR@parsetablecols {<colspecs>}

Scans the column specification left to right.
```

Table 6: 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 5, \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.
- @, >, <:** Ignored so far.
-

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

```
2193 \newcommand*{\LWR@parsetablecols}[1]{%
2194 \LWR@traceinfo{\LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
2195 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
2196 \renewcommand*{\LWR@tablecolspec}{}%
```

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

```
2197 \setcounter{LWR@tabletotalcols}{0}%
2198 \setcounter{LWR@tabletotalcolsnext}{1}%
2199 \LWR@setexparray{LWR@colatspec}{leftedge}{\relax}%
2200 \LWR@setexparray{LWR@colatspec}{1}{\relax}%
2201 \LWR@setexparray{LWR@colatspec}{2}{\relax}%
2202 \LWR@setexparray{LWR@colatspec}{3}{\relax}%
2203 \LWR@setexparray{LWR@colbeforespec}{1}{\relax}%
2204 \LWR@setexparray{LWR@colbeforespec}{2}{\relax}%
2205 \LWR@setexparray{LWR@colbeforespec}{3}{\relax}%
2206 \LWR@setexparray{LWR@colafterspec}{1}{\relax}%
2207 \LWR@setexparray{LWR@colafterspec}{2}{\relax}%
2208 \LWR@setexparray{LWR@colafterspec}{3}{\relax}%
```

Starting at the first column specification:

```
2209 \setcounter{LWR@tablecolspos}{1}%
```

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

```
2210 \LWR@traceinfo{about to StrLen}%
2211 \noexpandarg%
2212 \StrLen{#1}[\LWR@strresult]%
2213 \fullexpandarg%
2214 \LWR@traceinfo{finished StrLen}%
2215 \setcounter{LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
2216 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolswidth}}{%
```

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

```
2217 \noexpandarg%
2218 \StrChar{#1}{\the\LWR@tablecolspos}[\LWR@strresult]%
2219 \fullexpandarg%
```



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

```
2220 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
2221 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
2222 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
2223 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
2224 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
2225 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
2226 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
2227 \IfStrEq{\LWR@strresult}{!}{\LWR@parseskipcolumn}{}%
2228 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
2229 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
2230 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnstype{P}[1]{>\centering\arraybackslash}p{#1}}
```

```
2231 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
2232 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
2233 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
2234 \addtocounter{\LWR@tablecolspos}{1}%
2235 }%
2236 }%
```

49.10 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.

```
2237 \newcommand*{\LWR@maybenewtablerow}
2238 {%
2239 \ifbool{\LWR@startedrow}%
2240 {}% started the row
2241 {% not started the row
```

Remember that now have started the row:

```
2242 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```

2243 \global\booltrue{LWR@intabularmetadata}%
2244 \ifbool{LWR@doinghline}%
2245 {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
2246 {% not doing hline
2247 \ifbool{LWR@doingtbrule}%
2248 {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
2249 {\LWR@htmltag{tr}\LWR@orignewline}%
2250 }% end of not doing hline
2251 }% end of not started the row
2252 }
```

49.11 Data opening tag

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

```

2253 \newcommand*{\LWR@tabledatasinglecolumntag}%
2254 {%
2255 \LWR@maybe newtable row%
```

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

```

2256 \ifbool{LWR@exitingtabular}{}%
2257 {% not exiting tabular
```

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

```

2258 \StrChar{\LWR@tablecolspec}{\the\LWR@tablecolspos}[\LWR@strresult]%
```

print the start of a new table data cell:

```

2259 \LWR@htmltag{td class="td%
```

append this column's spec:

```

2260 \LWR@strresult%
```

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

```

2261 \ifthenelse{\equal{\LWR@gettexpparray{LWR@midrules}{\the\LWR@tablecolspos}}{Y}}{rule}{}%
2262 "{}}%
```

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

```

2263 \ifthenelse{%
```

```

2264 \equal{\LWR@strresult}{p}\OR%
2265 \equal{\LWR@strresult}{m}\OR%
2266 \equal{\LWR@strresult}{b}\OR%
2267 \equal{\LWR@strresult}{P}\OR%
2268 \equal{\LWR@strresult}{M}\OR%
2269 \equal{\LWR@strresult}{B}%
2270 }%
2271 {% allow pars
2272 \LWR@startpars%
2273 \global\booltrue{\LWR@tableparcell}%
2274 }% allow pars
2275 {}% no pars

```

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

```

2276 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}%
2277 {\LWR@getexparray{\LWR@colatspec}{leftedge}}% left edge
2278 {}% not left edge
2279 \LWR@getexparray{\LWR@colbeforespec}{\the\LWR@tablecolspos}%
2280 \global\boolfalse{\LWR@intabularmetadata}%
2281 }% not exiting tabular
2282 }%

```

49.12 Midrules

LWR@midrules LWR@midrules is a data array (section 23) of columns containing Y if a midrule should be created for each column.

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

```

2283 \newcounter{\LWR@midrulecounter}

```

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

Clears all LWR@midrules markers for this line.

```

2284 \newcommand*{\LWR@clearmidrules}
2285 {%
2286 \setcounter{\LWR@midrulecounter}{1}%
2287 \whiledo{%
2288 \cnttest{\value{\LWR@midrulecounter}}{<=}{\value{\LWR@tablecolswidth}}%
2289 }%
2290 {%
2291 \LWR@setexparray{\LWR@midrules}{\the\LWR@midrulecounter}{\relax}%
2292 \addtocounter{\LWR@midrulecounter}{1}%
2293 }%
2294 }

```

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

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

```
2295 \newcommand*\LWR@subcmidrule[4]{%
2296 \setcounter{LWR@midrulecounter}{#3}%
2297 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
2298 {%
2299 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
2300 \addtocounter{LWR@midrulecounter}{1}%
2301 }% end of the whiledo
2302 }
```

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

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

```
2303 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
2304 {\LWR@subcmidrule{#1}{#2}{#3}}
```

49.13 Multicolumns

49.13.1 Parsing multicolumns

```
2305 \newcounter{LWR@tablemulticolwidth}
2306 \newcounter{LWR@tablemulticolspos}
```

`\LWR@printmccoltype` [*colspec*] Print any valid column type found. Does not print @, >, or < columns or their associated tokens.

```
2307 \newcommand*\LWR@printmccoltype[1]{%
2308 \LWR@traceinfo{lw@printmccoltype -#1-}%
```

Get one token of the column spec:

```
2309 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

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

```
2310 \IfStrEq{\LWR@strresult}{l}{l}{}%
2311 \IfStrEq{\LWR@strresult}{c}{c}{}%
2312 \IfStrEq{\LWR@strresult}{r}{r}{}%
2313 \IfStrEq{\LWR@strresult}{p}{p}{}%
2314 \IfStrEq{\LWR@strresult}{m}{m}{}%
2315 \IfStrEq{\LWR@strresult}{b}{b}{}%
2316 \IfStrEq{\LWR@strresult}{P}{P}{}%
```

```

2317 \IfStrEq{\LWR@strresult}{M}{M}{}%
2318 \IfStrEq{\LWR@strresult}{B}{B}{}%
2319 \IfStrEq{\LWR@strresult}{S}{r}{}%
2320 \LWR@traceinfo{lw@printmccoltype done}%
2321 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

2322 \newcommand*{\LWR@multicolpartext}{%
2323 \LWR@startpars%
2324 \LWR@multicoltext%
2325 \LWR@stoppars%
2326 }

```

`\LWR@multicolother` `{\colspec}` For @, >, <, print the next token without paragraph tags:

```

2327 \newcommand*{\LWR@multicolother}[1]{%
2328 \addtocounter{LWR@tablemulticolspos}{1}%
2329 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
2330 \LWR@strresult%
2331 }

```

`\LWR@printmccoldata` `{\colspec}` Print the data for any valid column type found.

```

2332 \newcommand*{\LWR@printmccoldata}[1]{%
2333 \LWR@traceinfo{lw@printmccoldata -#1}%

```

Get one token of the column spec:

```

2334 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```

2335 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
2336 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
2337 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
2338 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}{}%
2339 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}{}%
2340 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}{}%
2341 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}{}%
2342 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}{}%
2343 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}{}%
2344 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}{}%
2345 \IfStrEq{\LWR@strresult}{\detokenize@}{\LWR@multicolother{#1}}{}%
2346 \IfStrEq{\LWR@strresult}{\detokenize>}{\LWR@multicolother{#1}}{}%
2347 \IfStrEq{\LWR@strresult}{\detokenize<}{\LWR@multicolother{#1}}{}%

```

```

2348 % \addtocounter{LWR@tablemulticolspos}{1}%
2349 % \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
2350 % \LWR@strresult%
2351 % }{}%
2352 \LWR@traceinfo{lwr@printmccoldata done}%
2353 }

```

`\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`.

```

2354 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
2355 \setcounter{LWR@tablemulticolspos}{1}%
2356 \StrLen{#1}[\LWR@strresult]%
2357 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

2358 \whiledo{%
2359 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
2360 }%
2361 {%

```

Execute the assigned print function for each token in the column spec:

```

2362 #2{#1}%

```

Move to the next token in the column spec:

```

2363 \addtocounter{LWR@tablemulticolspos}{1}%
2364 }%
2365 }

```

49.13.2 High-level multicolumn interface

`\LWR@domulticolumn` $\{\langle 1: \textit{numcols} \rangle\} \{\langle 2: \textit{colspec} \rangle\} \{\langle 3: \textit{text} \rangle\}$

```

2366 \newcommand{\LWR@multicoltext}{}
2367
2368 \NewDocumentCommand{\LWR@domulticolumn}{m m +m}{%
2369 \LWR@traceinfo{lwr@domulticolumn -#1- -#2-}%
2370 \renewcommand{\LWR@multicoltext}{#3}%
2371 \LWR@maybenewtablerow%

```

Begin the opening table data tag:

```
2372 \LWR@htmltag{td colspan="#1"
2373 class="td%
```

Print the column type:

```
2374 \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”.

```
2375 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

Close the class tag’s opening quote:

```
2376 "%
2377 }% end of the opening table data tag
2378 \global\boolfalse{\LWR@intabularmetadata}%
2379 \LWR@parsemulticolumnalignment{#2}{\LWR@printmccoldata}%
2380 }
```

49.13.3 Longtable captions

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```
2381 \newbool{\LWR@starredlongtable}
2382 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
2383 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] {*caption*}

```
2384 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
2385 {%
```

Remember the latest name for `\nameref`:

```
2386 \IfValueTF{#2}{% optional given?
2387 \ifthenelse{\equal{#2}{}}{% optional empty?
2388 {\LWR@setlatestname{#3}}% empty
2389 {\LWR@setlatestname{#2}}% given and non-empty
2390 }% optional given
2391 {\LWR@setlatestname{#3}}% no optional
```


create a multicolumn across all the columns

```

2392 \LWR@domulticolumn{\theLWR@tabletotalcols}{P}{% \LWR@domulticolumn
2393 % \IfBooleanTF{#1}% star?
2394 % {\IfValueTF{#2}{\LWR@origcaption*{#2}{#3}}{\LWR@origcaption*{#3}}}
2395 % {\IfValueTF{#2}{\LWR@origcaption{#2}{#3}}{\LWR@origcaption{#3}}}
2396 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

2397 {% yes star
2398 \LWR@htmlblocktag{figcaption}%
2399 #3%
2400 \LWR@htmlblocktag{/figcaption}%
2401 }%
2402 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

2403 \ifbool{LWR@starredlongtable}%
2404 {%
2405 \ifthenelse{\equal{#2}{}}% TOC entry
2406 {}%
2407 {%
2408 \refstepcounter{\LTcapttype}%
2409 \protected@edef\@currentlabel{%
2410 \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
2411 }%
2412 }{}%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

2413 \LWR@htmlblocktag{figcaption}%
2414 \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
2415 \LWR@htmlblocktag{/figcaption}%

```

See if an optional caption was given:

```

2416 \ifthenelse{\equal{#2}{}}% TOC entry empty

```

if the optional caption was given, but empty, do not form a TOC entry

```

2417 {}%

```

If the optional caption was given, but might only be []:

```

2418 {% TOC entry not empty

```

2419 \IfNoValueTF{#2}% No TOC entry?

The optional caption is []:

```
2420 {% No TOC entry
2421 \addcontentsline%
2422 {\csuse{ext@LTcapttype}}%
2423 {LTcapttype}%
2424 {%
2425 \protect\numberline%
2426 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2427 {\ignorespaces #3\protect\relax}%
2428 }%
2429 }% end of No TOC entry
```

The optional caption has text enclosed:

```
2430 {% yes TOC entry
2431 \addcontentsline%
2432 {\csuse{ext@LTcapttype}}%
2433 {LTcapttype}%
2434 {%
2435 \protect\numberline%
2436 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
2437 {\ignorespaces #2\protect\relax}%
2438 }%
2439 }% end of yes TOC entry
2440 }% end of TOC entry not empty
2441 }% end of no star
2442 }% end of \LWR@domulticolumn
2443
2444 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
2445 \addtocounter{LWR@tablecolspos}{-1}
2446
2447 }
```

49.13.4 \tabledatamulticolumntag

`\LWR@tabledatamulticolumntag` $\{ \langle numcols \rangle \} \{ \langle alignment \rangle \} \{ \langle text \rangle \}$

```
2448 \NewDocumentCommand{\LWR@tabledatamulticolumntag}{m m +m}%
2449 {%
2450 \LWR@domulticolumn{#1}{#2}{#3}%
2451 \addtocounter{LWR@tablecolspos}{#1}%
2452 \addtocounter{LWR@tablecolspos}{-1}%
2453 }
```

49.14 Multirow

Pkg multirow

```
\LWR@tabledatamultirowtag {\numrows} [\bigstruts] {\width} [\fixup] {\text}
```

```
2454 \NewDocumentCommand{\LWR@tabledatamultirowtag}{m o m o m}%
```

```
2455 {%
```

```
2456 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
2457 \LWR@htmltag{td rowspan="#1" class="td%
```

Append this column's spec:

```
2458 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

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

```
2459 \ifthenelse{\equal{\LWR@gettexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
```

```
2460 "{}}
```

While printing the text, redefine \\\ to generate a new line

```
2461 \begingroup \let\\\LWR@endoffline #5 \endgroup
```

```
2462 \LWR@stoppars%
```

```
2463 \global\boolfalse{\LWR@intabularmetadata}%
```

```
2464 }%
```

49.15 Utility macros inside a table

```
2465 \newcommand*{\LWR@donothing}{}%
```

```
2466 \newcommand*{\LWR@domidrule}{\booltrue{\LWR@doinghline}}%
```

```
2467 \newcommand*{\LWR@dotbrule}{\booltrue{\LWR@doingtbrule}}%
```

49.16 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:

```
2468 \newbool{\LWR@exitingtabular}
```

```
2469 \newcommand*{\LWR@tabledatacolumnntag}%
```

```
2470 {%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
2471 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
2472 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
2473 {\booltrue\LWR@exitingtabular}}{ }%
```

`longtable` can have a caption in a cell

```
2474 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
2475 {\let\mynext\LWR@donothing}}{ }%
```

Look for other things which would not start a table cell:

```
2476 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
2477 {\let\mynext\LWR@donothing}}{ }%
2478 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
2479 {\let\mynext\LWR@donothing}}{ }%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
2480 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
2481 {\let\mynext\LWR@donothing}}{ }%
2482 %
2483 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
2484 {\let\mynext\LWR@donothing}}{ }%
2485 %
2486 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
2487 { }%
2488 \let\mynext\LWR@donothing}}{ }%
2489 %
2490 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
2491 {\let\mynext\LWR@donothing}}{ }%
2492 %
2493 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
2494 {\let\mynext\LWR@donothing}}{ }%
2495 %
2496 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
2497 {\let\mynext\LWR@donothing}}{ }%
2498 %
2499 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
2500 {\let\mynext\LWR@donothing}}{ }%
2501 %
2502 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprintonly}}
```

```

2503 {\let\mynext\LWR@donothing}{}%
2504 %
2505 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}
2506 {\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:


```

2507 \mynext%
2508 }

2509 \end{warpHTML}

```

49.17 `\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:

```

2510 \begin{warpall}
2511 \newcommand*{\mrowcell}{}
2512 \end{warpall}

```

49.18 New `\tabular` definition

for HTML output:

```

2513 \begin{warpHTML}

```

Env `LWR@tabular` [*⟨verticalposition⟩*] {*⟨colspecs⟩*}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

2514 \newenvironment*{LWR@tabular}[2]{}
2515 {%
2516 \LWR@traceinfo{tabular started}%
2517 \begingroup%
2518 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

2519 \global\boolfalse{LWR@startedrow}%

```

Not yet doing an hline:

```
2520 \global\boolfalse{LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
2521 \global\boolfalse{LWR@doingtbrule}%
```

Have not yet found the end of tabular command:

```
2522 \boolfalse{LWR@exitingtabular}%
```

Create the `table` tag:

```
2523 \global\booltrue{LWR@intabularmetadata}%
2524 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
2525 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
2526 \LWR@stoppars%
```

Track column # for setting text-align:

```
2527 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
2528 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
2529 \let\\\LWR@tabularendoffline%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynexttoken` to see if the next token might create a new data cell:

```
2530 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
2531 \newcommand*{\midrule}{\LWR@domidrule\LWR@getmynexttoken}%
2532 \NewDocumentCommand{\cmidrule}{o d() m}%
2533 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
2534 \RenewDocumentCommand{\cline}{m}%
```

```

2535 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
2536 \newcommand*{\toprule}{\LWR@dotbrule\LWR@getmynexttoken}%
2537 \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.

```

2538 \let\multicolumn\LWR@tabledatamulticolumntag%
2539 \let\multirow\LWR@tabledatamultirowtag%
2540 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
2541 \let\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

2542 \global\boolfalse{\LWR@tableparcell}%
2543 \global\boolfalse{\LWR@skippingmrowcell}%

```

Look ahead for a possible table data cell:

```

2544 \LWR@getmynexttoken%
2545 }%

```

Ending the environment:

```

2546 {%
2547 \LWR@closetabledatacell%
2548 \LWR@htmlblocktag{/tr}%
2549 \LWR@htmlblocktag{/table}%
2550 \global\boolfalse{\LWR@intabularmetadata}%

2551 \addtocounter{\LWR@tabulardepth}{-1}%
2552 \endgroup%
2553 }

2554 \end{warpHTML}

```

49.19 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 7 shows the data structures related to cross-referencing.

for HTML output: 2555 `\begin{warpHTML}`

50.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
2556 \newcommand*{\@currentlabelname}{}%
```

`\LWR@stripperperiod` `{\text}` [`\langle`.`\rangle`]

Removes a trailing period.

```
2557 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

```
2558 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period.
See `zref-titleref` and `getttitlestring`.

```
2559 \GetTitleStringExpand{#1}%
2560 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
2561 \edef\@currentlabelname{%
2562 \expandafter\LWR@stripperperiod\@currentlabelname%
2563 \ltx@empty.\ltx@empty\@nil%
2564 }%
2565 }
```


Table 7: Cross-referencing data structures

Original L^AT_EX:	(print and HTML)
\refstepcounter: Steps the counter and sets \@currentlabel.	
\@currentlabel: \p@<ctr>\the<ctr> Updated by \refstepcounter.	
\label: Writes to the .aux file: \newlabel{<label>}{\@currentlabel}{\thepage}}	
\newlabel: When the .aux file is read, sets \r@<label>.	
\r@<label>: Set to: {\@currentlabel}{\thepage}}	
\ref: Returns the first part of \r@<label>.	
\pageref: Returns the second part of \r@<label>.	
Added by amsmath:	(print and HTML)
\label: Delayed until math is completed.	
Added by lwarp:	(HTML only)
\nameref: Emulated from hyperref for lwarp. See section 50.4.	
\ref and \nameref: Adds HTML tags. See section 50.4.	
Added by cleverref:	(print and HTML)
\refstepcounter: Added: sets \cref@currentlabel.	
\cref@currentlabel: (<type>=<ctr> unless an alias is used): [<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>} Also see section 35.4 for use with footnotes.	
\label: Writes to the .aux file: \newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}	
\newlabel: (Unchanged.) When the .aux file is read, sets \r@<label>@cref.	
\r@<label>@cref: Set to: {\cref@currentlabel}{\thepage}}	
Utility functions: See \cref@getlabel, \cref@gettype, \cref@getcounter, \cref@getprefix.	
Cross-referencing names: \crefname and \Crefname assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
\cref, etc.: Modified for lwarp. See section 59.	
\label: Adds HTML tags (section 50.3), plus \splabel data (section 50.2): zLWR@name: The section name for this label. zLWR@htmlfilenumber: The filenumber or name for this label. zLWR@lateximagedepth: The lateximagedepth for this label. zLWR@lateximagenumber: The lateximagenumber for this label.	
\label inside math: See section 54.4.1.	
Footnotes: See \noteentry in section 35.4.	

50.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called special:

```
2566 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
2567 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
2568 \zref@newprop{zLWR@htmlfilenumber}{%
2569 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
2570 }%
```

Additional properties for lateximages:

```
2571 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
2572 \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:

```
2573 \zref@addprop{special}{zLWR@name}
2574 \zref@addprop{special}{zLWR@htmlfilenumber}
2575 \zref@addprop{special}{zLWR@lateximagedepth}
2576 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
2577 \newcommand*{\LWR@spreff}[2]{%
2578 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```
2579 \newcommand*{\LWR@nameref}[1]{%
2580 \LWR@spreff{#1}{zLWR@name}%
2581 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```

2582 \newcommand*{\LWR@htmlfileref}[1]{%
2583 \LWR@traceinfo{\LWR@htmlfileref A: !#1!}%
2584 \LWR@sprep{#1}{z\LWR@htmlfilenumber}%
2585 \LWR@traceinfo{\LWR@htmlfileref B}%
2586 }

```

`\LWR@lateximagedepthref` `{\label}` Returns the `lateximagedepth` for this label:

```

2587 \newcommand*{\LWR@lateximagedepthref}[1]{%
2588 \LWR@sprep{#1}{z\LWR@lateximagedepth}%
2589 }

```

`\LWR@lateximagenumberref` `{\label}` Returns the `lateximagenumber` for this label:

```

2590 \newcommand*{\LWR@lateximagenumberref}[1]{%
2591 \LWR@sprep{#1}{z\LWR@lateximagenumber}%
2592 }

```

`\LWR@splabel` `{\label}` Sanitize the name and then creates the label:

```

2593 \newcommand*{\LWR@splabel}[1]{%
2594 \LWR@setlatestname{@currentlabelname}%
2595 \zref@labelbylist{#1}{special}}

```

50.3 Labels

`\LWR@subsublabel` `{\label}` Creates an HTML id tag.

```

2596 \newcommand*{\LWR@subsublabel}[1]{%

```

Create an HTML id tag unless are inside a `lateximage`, since it would appear in the image:

```

2597 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
2598 {}%
2599 {% not lateximage

```

If not doing a `lateximage`, create an HTML ID tag: (To be factored...)

```

2600 \ifbool{\LWR@doingstartpars}%
2601 {% pars allowed
2602 \ifbool{\LWR@doingapar}
2603 {% par started
2604 \LWR@htmltag{a id="#1"}{\LWR@htmltag{/a}%
2605 }% par started

```

```

2606 {% par not started
2607 \LWR@stoppars%
2608 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2609 \LWR@startpars%
2610 }% par not started
2611 }% pars allowed
2612 {% pars not allowed
2613 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
2614 }% pars not allowed
2615 }% not lateximage
2616 }

```

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

`\label` during HTML output. Creates a traditional L^AT_EX label, adds a special label with additional information, and creates an HTML id tag.

Used by `\LWR@htmllabel` and `\LWR@htmlmathlabel`.

```

2617 \newcommand*{\LWR@sublabel}[1]{%

```

Create a traditional LaTeX label, as modified by `cleveref`:

```

2618 \LWR@traceinfo{\LWR@sublabel: !#1!}%
2619 \LWR@origlabel{#1}%

```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

2620 \LWR@traceinfo{about to splabel}%
2621 \LWR@traceinfo{filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
2622 \LWR@traceinfo{\LWR@thisfilename is !\LWR@thisfilename!}%
2623 \LWR@traceinfo{\LWR@htmlfilenumber is \the\LWR@htmlfilenumber}%
2624 \LWR@splabel{#1}%
2625 \LWR@subsublabel{#1}%
2626 \LWR@traceinfo{finished with \LWR@sublabel}%
2627 }

```

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

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular L^AT_EX `\label`.

```

2628 \newcommand*{\LWR@newlabel}[1]{%
2629 \LWR@traceinfo{\LWR@newlabel: !#1!}%
2630 % \@bsphack%
2631 \LWR@sublabel{#1}%
2632 % \@esphack%

```

```
2633 \LWR@traceinfo{lwr@newlabel: done}%
2634 }
```

50.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
2635 \newcommand*{\LWR@startref}[1]
2636 {%
2637 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}
2638 \LWR@traceinfo{***LWR@startref A: !#1!}%
```

Create the tag:

```
2639 \LWR@htmltag{a href="%
2640 \LWR@traceinfo{***LWR@startref B}%
2641 \LWR@htmlrefsectionfilename{#1}%
2642 \LWR@traceinfo{***LWR@startref C}%
2643 \#%
```

See if `LWR@lateximagedepth` is unknown:

```
2644 \LWR@traceinfo{***LWR@startref D: !#1!}%
2645 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown:

```
2646 {??}%
```

If `LWR@lateximagedepth` is known:

```
2647 {%
2648 \LWR@traceinfo{***LWR@startref D1}%
2649 \LWR@traceinfo{***depthref is \LWR@lidref}%
2650 \ifthenelse{\cinttest{\LWR@lidref}{>}{0}}%
2651 {%
2652 \LWR@traceinfo{***LWR@startref D2}%
2653 lateximage\LWR@lateximagenumberref{#1}%
2654 }%
2655 {%
2656 \LWR@traceinfo{***LWR@startref D3}%
2657 #1%
2658 }%
2659 }%
2660 \LWR@traceinfo{***LWR@startref E}%
```

closing quote:

```
2661 "{}}%
2662 \LWR@traceinfo{***LWR@startref F}%
2663 }
```

`\ref * {<label>}` Create an internal document reference link, or without a link if starred per hyperref.

```
2664 \NewDocumentCommand{\LWR@newref}{s m}{%
2665 \LWR@traceinfo{LWR@newref #2}%
2666 \IfBooleanTF{#1}%
2667 {\LWR@origref{#2}}%
2668 {%
2669 \LWR@startref{#2}%
2670 \LWR@origref{#2}%
2671 \LWR@htmltag{/a}%
2672 }%
2673 }
2674
```

`\pagerefPageFor` Text for starred page references.

```
2675 \newcommand*{\pagerefPageFor}{page for}
```

`\pageref * {<label>}` Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
2676 \NewDocumentCommand{\LWR@newpageref}{s m}{%
2677 \IfBooleanTF{#1}%
2678 {\(\pagerefPageFor\ \LWR@origref{#2})}%
2679 {\(\cpageref{#2})}%
2680 }
```

`\nameref {<label>}`

```
2681 \newcommand*{\nameref}[1]{%
2682 \LWR@traceinfo{nameref A}%
2683 \LWR@startref{#1}%
2684 \LWR@traceinfo{nameref B}%
2685 \LWR@nameref{#1}%
2686 \LWR@traceinfo{nameref C}%
2687 \LWR@htmltag{/a}%
2688 \LWR@traceinfo{nameref D}%
2689 }
```

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.

```
2690 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF:
2691 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {\URL} {\text}
```

```
2692 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
2693 \LWR@htmltag{a href="#1" target="_{}blank"\LWR@orignewline}#2\LWR@htmltag{/a}%
2694 \LWR@ensuredoingapar%
2695 }
```

```
\LWR@subhyperrefclass {\URL} {\text} {\htmlclass}
```

```
2696 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
2697 \LWR@htmltag{a href="#1"
2698 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
2699 \LWR@ensuredoingapar%
2700 }
```

```
\href [{options}] {\URL} {\text}
```

Create a link with accompanying text:

```
2701 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
2702 \LWR@subhyperref{#2}{#3}%
2703 \endgroup%
2704 \LWR@ensuredoingapar%
2705 }
2706
2707 \newcommand{\href}{%
2708 \LWR@ensuredoingapar%
```

```

2709 \begingroup%
2710 \catcode'\_ =12
2711 \LWR@hrefb%
2712 }

```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```

2713 \newcommand*\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
2714
2715 \newcommand{\nolinkurl}{%
2716 \LWR@ensuredoingapar%
2717 \begingroup\catcode'\_ =12
2718 \LWR@nolinkurlb%
2719 }

```

`\url` $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link:

```

2720 \newcommand*\LWR@urlb}[1]{%
2721 \href{#1}{#1}%
2722 \endgroup%
2723 \LWR@ensuredoingapar%
2724 }
2725
2726 \newcommand{\url}{%
2727 \LWR@ensuredoingapar%
2728 \begingroup\catcode'\_ =12
2729 \LWR@urlb%
2730 }

```

`\LWR@subinlineimage` $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

2731 \newcommand*\LWR@subinlineimage}[5][]{%
2732 \ifthenelse{\equal{#1}{}}{%
2733 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
2734 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
2735 }
2736 \end{warpHTML}

```


Table 8: 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 8 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: 2737 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
2738 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
2739 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
2740 \addtocounter{LWR@thisfloat}{1}%
2741 \booltrue{LWR@freezethisfloat}%
```

```
2742 \begingroup
```

Settings while inside the environment:

```
2743 \LWR@origraggedright
```

Open an HTML figure tag:

```
2744 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
2745 \renewcommand*{\@capttype}{#1}
2746 \caption@settype{#1}
2747 \LWR@startpars
2748 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
2749 \let\@float\LWR@floatbegin
2750 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
2751 \newcommand*{\LWR@floatend}{%
2752 \LWR@stoppars%
2753 \LWR@htmllementend{figure}%
2754 \endgroup%
2755 \boolfalse{LWR@freezethisfloat}%
2756 \LWR@startpars%
2757 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```
2758 \let\end@float\LWR@floatend
2759 \let\end@dblfloat\LWR@floatend
```

Ctr `LWR@thisfloat` A sequential counter for all floats. This is used to identify the float then reference it from the List of Figures and List of Tables.

```
2760 \newcounter{LWR@thisfloat}
```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```
2761 \newbool{LWR@freezethisfloat}
2762 \boolfalse{LWR@freezethisfloat}
```

`\LWR@maybeinthisfloat`

```
2763 \newcommand*{\LWR@maybeinthisfloat}{%
2764 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
2765 }
```

`\@capttype` Remembers which float type is in use.

```
2766 \newcommand*{\@capttype}{}
```

51.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
2767 \AtBeginDocument{\providecommand*\CaptionSeparator}{:~}}
```

`\@makecaption` $\{\langle name \text{ and } num \rangle\} \{\langle text \rangle\}$

Prints the float type and number, the caption separator, and the caption text.

```
2768 \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.

```
2769 \newcounter{LWR@nextautofloat}
2770 \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.

```
2771 \newcommand*{\LWRsetnextfloat}[2]{%
2772 \setcounter{LWR@nextautopage}{#1}%
2773 \setcounter{LWR@nextautofloat}{#2}%
2774 }
```

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.

```
2775 \newcounter{LWR@latestautopage}% updated each new HTML file
2776 \setcounter{LWR@latestautopage}{1}
```

```
2777 \let\LWR@origcaption@begin\caption@begin
2778 \let\LWR@origcaption@end\caption@end
2779 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
2780 \newcommand{\LWR@caption@begin}
2781 {
2782 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
2783 \begingroup%
```

The `caption` code was not allowing the closing `par` tag:

```
2784 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage inside the caption:

```
2785 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}{-}{-}%
```

Enclose the original caption code inside an HTML tag:

```
2786 \LWR@htmlblocktag{figcaption}%
2787 \LWR@origcaption@begin%
2788 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
2789 \newcommand{\LWR@caption@end}
2790 {%
2791 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
2792 \vspace*{\baselineskip}%
```

Closing tag:

```
2793 \LWR@htmlblocktag{/figcaption}%
2794 \endgroup%
2795 % \leavevmode% avoid bad space factor (0) error
2796 \LWR@traceinfo{\LWR@caption@end: done}%
2797 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
2798 \let\caption@begin\LWR@caption@begin
2799 \let\caption@end\LWR@caption@end
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
2800 \let\LWR@origcaptionlistentry\captionlistentry
2801
2802 \renewcommand*{\captionlistentry}{%
2803 \LWR@maybeinthisfloat%
2804 \LWR@ensuredoingapar%
2805 \LWR@htmltag{a id="autofloat-\arabic{\LWR@thisfloat}}{\LWR@htmltag{/a}%
2806 \LWR@origcaptionlistentry%
2807 }
2808
2809 \def\LWR@LTcaptionlistentry{%
2810 \LWR@ensuredoingapar%
```

```

2811 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2812 \bgroup
2813 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
2814 {\egroup\LWR@LT@captionlistentry}}%
2815 \def\LWR@LT@captionlistentry#1{%
2816 \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autofloat before each float's entry. No changes if writing `.toc`.

```

2817 \let\LWR@origaddcontentsline\addcontentsline
2818
2819 \renewcommand*\addcontentsline[3]{%
2820 \ifthenelse{\equal{#1}{toc}}{ }{%
2821 \addtocontents{\@nameuse{ext@#2}}{%
2822 \protect\LWRsetnextfloat%
2823 {\arabic{LWR@latestautopage}}%
2824 {\arabic{LWR@thisfloat}}%
2825 }%
2826 }%
2827 \LWR@origaddcontentsline{#1}{#2}{#3}%
2828 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

2829 \AtBeginDocument{
2830 \let\LWR@origcaptionof\captionof
2831
2832 \renewcommand*\captionof{%
2833 \LWR@maybeinthisfloat%
2834 \LWR@stoppars
2835 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
2836 \LWR@origcaptionof%
2837 }
2838 }
2839 \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: 2840 \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.

```
2841 \newcommand*\LWR@myshorttoc}[1]{
2842 \LWR@ensuredoingapar
```

Only if the file exists:

```
2843 \IfFileExists{\jobname.#1}{
```



Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.



For 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.

```
2844 \begingroup
2845 % \ifxetexorluatex%
2846 % \else
2847 % \inputencoding{latin1}% currently disabled
2848 % \fi
2849 \makeatletter
```

Read in the TOC file:

```
2850 \@input{\jobname.#1}
2851 % \makeatother
2852 \endgroup
2853 }%
2854 {}%
```

2855 }

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

2856 `\NewDocumentCommand{\LWR@subtableofcontents}{m m}{%`

Closes previous levels:

2857 `\@ifundefined{chapter}`

2858 `{\LWR@closeprevious{\LWR@depthsection}}`

2859 `{\LWR@closeprevious{\LWR@depthchapter}}`

Prints any pending footnotes so that they appear above the potentially large TOC:

2860 `\LWR@printpendingnotes`

Place the list into its own chapter (if defined) or section:

2861 `\@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}`

Create a new HTML `nav` containing the TOC/LOF/LOT:

2862 `\LWR@htmlclass{nav}{#1}`

Create the actual list:

2863 `\LWR@myshorttoc{#1}`

Close the `nav`:

2864 `\LWR@htmlclassend{nav}{#1}`

2865 }

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

2866 `\let\LWR@orig@starttoc\@starttoc`

2867

2868 `\renewcommand{\@starttoc}[1]{`

2869 `\LWR@htmlclass{nav}{#1}`

2870 `\LWR@orig@starttoc{#1}`

2871 `\LWR@htmlclassend{nav}{#1}`

2872 }

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.


```

2873 \let\LWR@origtableofcontents\tableofcontents
2874 \let\LWR@origlistoffigures\listoffigures
2875 \let\LWR@origlistoftables\listoftables
2876 \renewcommand*{\tableofcontents}{%

```

Copy the .toc file to .sidetoc for printing the sidetoc. The original .toc file is renewed when \tableofcontents is finished.

```

2877 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
2878 \LWR@printpendingnotes%
2879 \LWR@origtableofcontents%
2880 }
2881 \renewcommand*{\listoffigures}{\LWR@printpendingnotes\LWR@origlistoffigures}
2882 \renewcommand*{\listoftables}{\LWR@printpendingnotes\LWR@origlistoftables}

```

52.2 High-level TOC commands

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

Emulate the `\listof` command from the `float` package (section 82). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```

2883 \NewDocumentCommand{\listof}{m +m}{%
2884 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
2885 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
2886 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
2887 \jobname.\csuse{ext@#1}\relax
2888 }

```

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.

```
2889 \end{warpHTML}
```

for HTML & PRINT: 2890 \begin{warpall}

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to tocdepth.

```
2891 \newcounter{SideTOCDepth}
2892 \setcounter{SideTOCDepth}{1}
```

\sidetocname Holds the default name for the sidetoc.

```
2893 \newcommand{\sidetocname}{Contents}
```

```
2894 \end{warpall}
```

for HTML output: 2895 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
2896 \newcommand*{\LWR@sidetoc}{
2897 \LWR@stoppars
2898
```

The entire sideTOC is placed into a nav of class sidetoc.

```
2899 \LWR@html elementclass{nav}{sidetoc}
2900
2901 \setcounter{tocdepth}{\value{SideTOCDepth}}
2902
```

The title is placed into a div of class sidetoc title, and may contain paragraphs.

```
2903 \begin{BlockClass}{sidetoc title}
2904 \sidetocname
2905 \end{BlockClass}
```

The table of contents is placed into a div of class sidetoc contents.

```
2906 \begin{BlockClass}{sidetoc contents}
2907 \LinkHome
2908
2909 \LWR@myshorttoc{sidetoc}
2910 \end{BlockClass}
2911 \LWR@html elementclassend{nav}{sidetoc}
2912 }
```

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:

```
2913 \renewcommand*\numberline[1]{%
2914 \LWR@sectionnumber{#1}%
2915 }
```

`\hypertoc` $\{\langle 1: depth \rangle\} \{\langle 2: class \rangle\} \{\langle 3: name \rangle\} \{\langle 4: page \rangle\}$

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 HTML CSS class is `tocsection`, `tocsubsection`, etc.

#3 the text of the caption

#4 page number

```
2916 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to `tocdepth`:

```
2917 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
2918 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
2919 \LWR@subhyperrefclass{%
2920 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{#2}%
2921 \LWR@stoppars%
2922 }
2923 {}
2924 }
```

`\hypertocfloat` $\{\langle 1: depth \rangle\} \{\langle 2: class \rangle\} \{\langle 3: caption \rangle\} \{\langle 4: page \rangle\}$

#1 is depth

#2 HTML CSS class is tocfigure, toctable

#3 the text of the caption

#4 page number

```
2925 \newcommand{\hypertocfloat}[4]{%
2926 \LWR@startpars
```

Create an HTML link to filename#autofloat-(float number), with text of the caption, of the given HTML class.

```
2927 \LWR@subhyperrefclass{%
2928 \LWR@htmlrefsectionfilename{autopage-\arabic{\LWR@nextautopage}}%
2929 \#autofloat-\arabic{\LWR@nextautofloat}}%
2930 {#3}{#2}%
2931 \LWR@stoppars%
2932 }
```

Automatically called by \contentsline:

```
2933 \renewcommand{\l@part}[2]{\hypertoc{-1}{tocpart}{#1}{#2}}
2934 \DeclareDocumentCommand{\l@chapter}{m m}
2935 {\hypertoc{0}{tocchapter}{#1}{#2}}
2936 \renewcommand{\l@section}[2]{\hypertoc{1}{tocsection}{#1}{#2}}
2937 \renewcommand{\l@subsection}[2]{\hypertoc{2}{tocsubsection}{#1}{#2}}
2938 \renewcommand{\l@subsubsection}[2]
2939 {\hypertoc{3}{tocsubsubsection}{#1}{#2}}
2940 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{tocparagraph}{#1}{#2}}
2941 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{tocsubparagraph}{#1}{#2}}
2942 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{tocfigure}{#1}{#2}}
2943 \renewcommand{\l@table}[2]{\hypertocfloat{1}{toctable}{#1}{#2}}

2944 \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: 2945 \begin{warpHTML}

\LWR@indexsection Controls whether the index will be in a section or a chapter, depending on the documentclass.

```

2946 \@ifundefined{chapter}
2947 {\newcommand*{\LWR@indexsection}{\section{\indexname}}}
2948 {\newcommand*{\LWR@indexsection}{\chapter{\indexname}}}

```

`\printindex`

```

2949 \let\LWR@origprintindex\printindex
2950
2951 \renewcommand*{\printindex}
2952 {
2953   \LWR@indexsection
2954   \LWR@startpars
2955   \LWR@origprintindex
2956 }

```

Env `theindex`

```

2957 \renewenvironment*{theindex}{%
2958 \let\item\LWR@indexitem%
2959 \let\subitem\LWR@indexsubitem%
2960 \let\subsubitem\LWR@indexsubsubitem%
2961 }{}

```

`\LWR@indexitem`

```

2962 \newcommand{\LWR@indexitem}{
2963
2964   \InlineClass{indexitem}{}
2965 }

```

`\LWR@indexitem`

```

2966 \newcommand{\LWR@indexsubitem}{
2967
2968   \InlineClass{indexsubitem}{}
2969 }

```

`\LWR@indexitem`

```

2970 \newcommand{\LWR@indexsubsubitem}{
2971
2972   \InlineClass{indexsubsubitem}{}
2973 }

```

`\hyperindexref` $\{(\textit{autosecnumber})\}$

`\hyperindexref{web address}` is inserted into `*.ind` by the xindy style file `lwarp_html.xdy`

```
2974 \newcommand*\hyperindexref}[1]{
2975 \LWR@htmltag{a href="%
```

Create an HTML reference to the autosection:

```
2976 \LWR@htmlrefsectionfilename{autopage-#1}\#autosec-#1"{}%}
```

Add the section number and section name, then close the tag:

```
2977 \LWR@origref{autopage-#1}\enskip\LWR@nameref{autopage-#1}%
2978 \LWR@htmltag{/a}%
2979 }
```

```
2980 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case `hyperref` was not used:

```
2981 \begin{warpprint}
2982 \newcommand*\hyperindexref}[1]{#1}
2983 \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 font⁸, 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

⁸See section 119 regarding fonts and fractions.

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 \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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<p>https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx</p>
L ^A T _E X macros	<ul style="list-style-type: none"> 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 ^A T _E X 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: 2984 \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.

```

2985 \let\LWR@origtextdollar\$
2986
2987 \renewcommand*{\$}{%
2988 \LWR@htmltagc{span}%
2989 \LWR@origtextdollar%
2990 \LWR@htmltagc{/span}%
2991 }

```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```

2992 \newcounter{LWR@externalfilecnt}

```


54.1 Inline and display math

```

2993 \let\LWR@origdollar=$
2994 \let\secondorigdollar=$% balance for editor syntax highlighting

2995 \let\LWR@origopenparen\l
2996 \let\LWR@origcloseparen\r

```

\$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
\$\$

```

2997 \begingroup
2998 \catcode'\$=\active%
2999 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```

3000 \gdef\LWR@doubledollar$#1$$ {
3001 \ifbool{mathjax}%

```

For MathJax, print the math between `\[` and `\]`:

```

3002 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

3003 {% not mathjax
3004
3005 \begin{lateximage}%
3006 [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
3007 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
3008 \end{lateximage}
3009
3010 }
3011 }%

```

\LWR@singledollar Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```

3012 \gdef\LWR@singledollar#1$ {%
3013 \ifbool{mathjax}%

```

For MathJax, print the math between `\(` and `\)`:

```

3014 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}

```

For SVG, print the math inside a `lateximage`, with an ALT tag of the \LaTeX code:

```

3015 {% not mathjax
3016 \begin{lateximage}%
3017 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%

```

```

3018 \LWR@origdollar#1\LWR@origdollar%
3019 \end{lateximage}%
3020 }%
3021 }%

```

\(Redefine to the above dollar macros.

```

\
3022 \gdef\(#1\){$#1$}
3023 \gdef\[#1\]{$$#1$$}
3024
3025 \endgroup

```

Remove the old `math` and `displaymath` environments:

```

3026 \let\math\relax
3027 \let\endmath\relax
3028 \let\displaymath\relax
3029 \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`.

```

3030 \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`.

```

3031 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

When the document begins, the dollar sign must be made active to trigger the new math macros:

```

3032 \AtBeginDocument{\catcode'\$=\active}

```

54.2 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```

3033 \newcounter{LWR@nextequation}

```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\($\)$ ” and “\($\)$ ” characters. They are printed to HTML output, not interpreted by \LaTeX .

```
3034 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
3035 \ifcsdef{thechapter}{
3036 \BlockClassSingle{hidden}{
3037 \textbackslash(
3038 \textbackslash{}seteqsection \{\thechapter\}
3039 \textbackslash)
3040 }
3041 }
3042 {}% not using chapters
3043
```

MathJax doesn't allow setting the equation number to 1:

```
3044 \ifthenelse{\cnttest{\value{equation}}>0}
3045 {
```

Tell MathJax that the next set of equations begins with the current L^AT_EX equation number, plus one.

```
3046 \setcounter{\LWR@nextequation}{\value{equation}}
3047 \addtocounter{\LWR@nextequation}{1}
```

Place the MathJax command inside “\ (“ and “\)” characters, to be printed to HTML, not interpreted by L^AT_EX.

```
3048 \BlockClassSingle{hidden}{
3049 \textbackslash(
3050 \textbackslash{}seteqnumber \{\arabic{\LWR@nextequation}\}
3051 \textbackslash)
3052 }
3053 }{}% not eq > 1
3054 }
```

\LWR@restoremathlatexformatting While producing math, use regular L^AT_EX formatting instead of HTML tags.

```
3055 \newcommand*{\LWR@restoremathlatexformatting}{%
3056 \let\hspace\LWR@origspace%
3057 \let\rule\LWR@origrule%
3058 \let\,\LWR@origcomma% disable HTML short unbreakable space
3059 \let\textit\LWR@origtextit%
3060 \let\textbf\LWR@origtextbf%
3061 \let\texttt\LWR@origtexttt%
3062 \let\textsc\LWR@origtextsc%
3063 \let\textsf\LWR@origtextsf%
```

```

3064 \let\textrm\LWR@origtextrm%
3065 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3066 \let\textsuperscript\LWR@origtextsuperscript%
3067 \let\textsubscript\LWR@origtextsubscript%
3068 \let~\LWR@origtilde%
3069 \let\enskip\LWR@origenskip%
3070 \let\quad\LWR@origquad%
3071 \let\qquad\LWR@origqquad%
3072 }

```

`\LWR@hidelatexequation` $\{\langle environment \rangle\}$ $\{\langle contents \rangle\}$

Creates the L^AT_EX version of the equation inside an HTML comment.

```

3073 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%

```

Stop HTML paragraph handling and open an HTML comment:

```

3074 \LWR@stoppars
3075 \LWR@htmlopencomment
3076

```

Start the LaTeX math environment inside the HTML comment:

```

3077 \begingroup
3078 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their L^AT_EX meanings.

```

3079 \LWR@restoremathlatexformatting

```

When a label is encountered, the HTML comment is closed, the label is generated in an HTML-output context, then the HTML comment is opened again. See `\LWR@htmlmathlabel` in section 54.4.1.

Print the contents of the equation:

```

3080 #2

```

End the L^AT_EX math environment inside the HTML comment:

```

3081 \csuse{\LWR@origend#1}
3082 \endgroup
3083

```

Close the HTML comment and resume HTML paragraph handling:

```

3084 \LWR@htmlclosecomment

```

```
3085 \LWR@startpars
3086 }
```

```
\LWR@addmathjax {<environment>} {<contents>}
```

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by L^AT_EX.

```
3087 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
3088
```

Enclose the MathJax environment inside printed “\(" and “\)” characters.

```
3089 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
3090 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
3091 \textbackslash{}end\{#1\}
3092
3093 }
```

54.3 Equation environment

Remember existing `equation` environment:

```
3094 \let\LWR@origequation\equation
3095 \let\LWR@origendequation\endequation
```

Remove existing `equation` environment:

```
3096 \let\equation\relax
3097 \let\endequation\relax
```

Env **equation** The new `equation` environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the L^AT_EX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
3098 \NewEnviron{equation}{%
3099
3100 \ifbool{mathjax}
```

MathJax output:

```
3101 {
```

Print commands to synchronize MathJax's equation number and format to the current \LaTeX chapter/section and equation number:

```
3102 \LWR@syncmathjax
```

Print the \LaTeX math inside an HTML comment:

```
3103 \LWR@hidelatexequation{equation}{\BODY}
3104 }
```

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`.

```
3105 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
3106 \begin{lateximage}[(\theequation) \textbackslash{begin\{equation\}}}%
3107 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
3108 \textbackslash{end\{equation\}}}% alt tag
```

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
3109 \LWR@origequation
3110 \BODY% contents collected by NewEnviron
3111 \LWR@origendequation
3112 \end{lateximage}%
3113 }
3114
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
3115 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}}]
```

54.4 AMS Math environments

54.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for `multline`-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multline` environment and not an `equation`.

```
3116 \newbool{LWR@amsmultline}
3117 \boolfalse{LWR@amsmultline}
```

`\LWR@htmlmathlabel` $\langle label \rangle$

`\label` when called inside a \LaTeX AMS math environment's math mode which is encased inside an HTML comment while creating a HTML document.

```
3118
3119 \newcommand*{\LWR@htmlmathlabel}[1]{%
3120 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
3121 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
3122 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
3123 \ifbool{LWR@amsmultline}{\hspace*{\totwidth0}}
```

Temporarily end the HTML comment, insert the \LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
3124 \LWR@htmlclosecomment\LWR@sublabel{\@firstofone#1}\LWR@htmlopencomment}%
3125 }%
3126 {\LWR@sublabel{\@firstofone#1}}%
3127 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
3128 \newcommand*{\LWR@beginhideamsmath}{
3129 \LWR@stoppars
3130
3131 \LWR@htmlopencomment
```

```

3132
3133 \begingroup
3134 \LWR@restoremathlatexformatting
3135 }

```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```

3136 \newcommand*{\LWR@endhideamsmath}{
3137 \endgroup
3138
3139 \LWR@htmlclosecomment
3140
3141 \LWR@startpars
3142 }

```

54.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the $\mathscr{A}\mathscr{M}\mathscr{S}$ environment.

For MathJax: Each environment is syched with \LaTeX 's equation numbers, typeset with \LaTeX inside an HTML comment, then printed to HTML output for MathJax to process.

Env `multline`

```

3143 \BeforeBeginEnvironment{multline}{
3144 \ifbool{mathjax}
3145 {
3146 \LWR@syncmathjax
3147 \booltrue{LWR@amsmultline}
3148 \LWR@beginhideamsmath
3149 }
3150 {
3151 \lateximage[LWRAMSMATHBODY]
3152 }
3153 }
3154
3155 \AfterEndEnvironment{multline}{
3156
3157 \ifbool{mathjax}
3158 {

```



```

3159 \LWR@endhideamsmath
3160 \boolfalse{LWR@amsmultline}
3161 \LWR@addmathjax{multline}{\the\@envbody}
3162 }
3163 {\endlateximage}
3164
3165 }

```

Env **multline***

```

3166 \BeforeBeginEnvironment{multline*}{
3167 \ifbool{mathjax}
3168 {
3169 \LWR@syncmathjax
3170 \booltrue{LWR@amsmultline}
3171 \LWR@beginhideamsmath
3172 }
3173 {
3174 \lateximage[LWR@MSMATHBODY]
3175 }
3176 }
3177
3178 \AfterEndEnvironment{multline*}{
3179
3180 \ifbool{mathjax}
3181 {
3182 \LWR@endhideamsmath
3183 \boolfalse{LWR@amsmultline}
3184 \LWR@addmathjax{multline*}{\the\@envbody}
3185 }
3186 {\endlateximage}
3187
3188 }
3189

```

Env **gather**

```

3190 \BeforeBeginEnvironment{gather}{
3191 \ifbool{mathjax}
3192 {
3193 \LWR@syncmathjax
3194 \boolfalse{LWR@amsmultline}
3195 \LWR@beginhideamsmath
3196 }
3197 {
3198 \lateximage[LWR@MSMATHBODY]
3199 }
3200 }
3201

```

```

3202 \AfterEndEnvironment{gather}{
3203
3204 \ifbool{mathjax}
3205 {
3206 \LWR@endhideamsmath
3207 \LWR@addmathjax{gather}{\the\@envbody}
3208 }
3209 {\endlateximage}
3210
3211 }

```

Env **gather***

```

3212 \BeforeBeginEnvironment{gather*}{
3213 \ifbool{mathjax}
3214 {
3215 \LWR@syncmathjax
3216 \boolfalse{LWR@amsmultline}
3217 \LWR@beginhideamsmath
3218 }
3219 {
3220 \lateximage[LWRAMSMATHBODY]
3221 }
3222 }
3223
3224 \AfterEndEnvironment{gather*}{
3225
3226 \ifbool{mathjax}
3227 {
3228 \LWR@endhideamsmath
3229 \LWR@addmathjax{gather*}{\the\@envbody}
3230 }
3231 {\endlateximage}
3232
3233 }

```

Env **align**

```

3234 \BeforeBeginEnvironment{align}{
3235 \ifbool{mathjax}
3236 {
3237 \LWR@syncmathjax
3238 \boolfalse{LWR@amsmultline}
3239 \LWR@beginhideamsmath
3240 }
3241 {
3242 \lateximage[LWRAMSMATHBODY]
3243 }
3244 }

```

```

3245
3246 \AfterEndEnvironment{align}{
3247
3248 \ifbool{mathjax}
3249 {
3250 \LWR@endhideamsmath
3251 \LWR@addmathjax{align}{\the\@envbody}
3252 }
3253 {\endlateximage}
3254
3255 }

```

Env **align***

```

3256 \BeforeBeginEnvironment{align*}{
3257 \ifbool{mathjax}
3258 {
3259 \LWR@syncmathjax
3260 \boolfalse{LWR@amsmultiline}
3261 \LWR@beginhideamsmath
3262 }
3263 {
3264 \lateximage[LWRAMSMATHBODY]
3265 }
3266 }
3267
3268 \AfterEndEnvironment{align*}{
3269
3270 \ifbool{mathjax}
3271 {
3272 \LWR@endhideamsmath
3273 \LWR@addmathjax{align*}{\the\@envbody}
3274 }
3275 {\endlateximage}
3276
3277 }

```

Env **flalign**

```

3278 \BeforeBeginEnvironment{flalign}{
3279 \ifbool{mathjax}
3280 {
3281 \LWR@syncmathjax
3282 \boolfalse{LWR@amsmultiline}
3283 \LWR@beginhideamsmath
3284 }
3285 {
3286 \lateximage[LWRAMSMATHBODY]
3287 }

```

```

3288 }
3289
3290 \AfterEndEnvironment{flalign}{
3291
3292 \ifbool{mathjax}
3293 {
3294 \LWR@endhideamsmath
3295 \LWR@addmathjax{flalign}{\the\@envbody}
3296 }
3297 {\endlateximage}
3298
3299 }

```

Env **flalign***

```

3300 \BeforeBeginEnvironment{flalign*}{
3301 \ifbool{mathjax}
3302 {
3303 \LWR@syncmathjax
3304 \boolfalse{LWR@amsmultiline}
3305 \LWR@beginhideamsmath
3306 }
3307 {
3308 \lateximage[LWRAMSMATHBODY]
3309 }
3310 }
3311
3312 \AfterEndEnvironment{flalign*}{
3313
3314 \ifbool{mathjax}
3315 {
3316 \LWR@endhideamsmath
3317 \LWR@addmathjax{flalign*}{\the\@envbody}
3318 }
3319 {\endlateximage}
3320
3321 }

3322 \end{wrapHTML}

```

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 `lwrapmk` 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: 3323 `\begin{warpHTML}`

`LWR@commentfooterstyle`

```
3324 \newpagestyle{LWR@commentfooterstyle}{
3325 \sethead{}{}{}
3326 \setfoot{\LWR@htmlopencomment}{}{}
3327 }
```

Ctr LWR@lateximagenumber Sequence the images.

```
3328 \newcounter{LWR@lateximagenumber}
3329 \setcounter{LWR@lateximagenumber}{0}
```

Ctr LWR@lateximagedepth Do not create `\lateximage` inside of `\lateximage`.

```
3330 \newcounter{LWR@lateximagedepth}
3331 \setcounter{LWR@lateximagedepth}{0}
```

Declare the `\LWR@file` for writing to generate file `lateximages.txt`:

```
3332 \ifcsdef{\LWR@file}{}{\newwrite{\LWR@file}}
```

A few utility macros to write special characters:

```
3333 \edef\LWR@hashmark{\string#} % for use in \write
3334 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr LWR@nextpage Used to count ahead to the next page:

```
3335 \newcounter{LWR@nextpage}

3336 \end{warpHTML}
```

for HTML & PRINT: 3337 \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.

3338 \newcommand*{\LateximageFontSizeName}{large}

3339 \end{warpall}

for HTML output: 3340 \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...

```
3341 \newcommand{\LWR@HTMLsanitize}[1]{%
3342 \protect\StrSubstitute{\detokenize{#1}}%
3343 {\detokenize{&}}%
3344 {\detokenize{&}}[\LWR@strresult]%
3345 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
3346 {\detokenize{<}}%
3347 {\detokenize{<}}[\LWR@strresult]%
3348 [\LWR@strresult]%
3349 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
3350 {\detokenize{>}}%
3351 {\detokenize{>}}[\LWR@strresult]%
3352 [\LWR@strresult]%
3353 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
3354 {\detokenize{##}}%
3355 {\detokenize{##}}[\LWR@strresult]%
3356 [\LWR@strresult]%
3357 \LWR@strresult%
3358 }
```

`\LWR@HTMLsanitizeexpand` $\{ \langle text \rangle \}$

This version expands the argument before sanitizing it.

```
3359 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
3360 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
3361 {\detokenize{&}}%
```

```

3362 {\detokenize{&}}}%
3363 [\LWR@strresult]%
3364 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}}%
3365 {\detokenize{<}}}%
3366 {\detokenize{&lt;}}}%
3367 [\LWR@strresult]%
3368 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}}%
3369 {\detokenize{>}}}%
3370 {\detokenize{&gt;}}}%
3371 [\LWR@strresult]%
3372 \LWR@strresult%
3373 }

```

Env `lateximage` [*alttag*]

```

3374 \NewDocumentEnvironment{lateximage}{0{image}}{%
3375 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}%

```

If nesting inside an already-existing `lateximage`, simply record one more level:

```

3376 {%
3377 \addtocounter{\LWR@lateximagedepth}{1}%
3378 }%

```

Otherwise, this is the outer-most `lateximage`:

```

3379 {% start of outer-most lateximage

```

Starting a new `lateximage`:

```

3380 \addtocounter{\LWR@lateximagenumber}{1}%

```

While inside a `lateximage`, do not use `mathjax`:

```

3381 \boolfalse{mathjax}

```

Be sure that are doing a paragraph:

```

3382 \LWR@ensuredoingapar%

```

Next file:

```

3383 \addtocounter{\LWR@externalfilecnt}{1}%

```

Figure out what the next page number will be:

```

3384 \setcounter{\LWR@nextpage}{\thepage}%
3385 \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:

```
3386 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
3387 class="lateximagesource">{}} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
3388 \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`.

```
3389 \LWR@origthispagestyle{LWR@commentfooterstyle}%
3390 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
3391 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
3392 \LWR@origminipage{6in}%
3393 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a lateximage.

```
3394 \let\hspace\LWR@orighspace%
3395 \let\rule\LWR@origrule%
3396 \let\,\LWR@origcomma% disable HTML short unbreakable space
3397 \let\textit\LWR@origtextit%
3398 \let\textbf\LWR@origtextbf%
3399 \let\texttt\LWR@origtexttt%
3400 \let\textsc\LWR@origtextsc%
3401 \let\textsf\LWR@origtextsf%
3402 \let\textrm\LWR@origtextrm%
3403 \renewcommand*{\thefootnote}{\fnsymbol{footnote}}%
3404 \let\textsuperscript\LWR@origtextsuperscript%
3405 \let\textsubscript\LWR@origtextsubscript%
3406 \let~\LWR@origtilde%
3407 \let\enskip\LWR@origenskip%
3408 \let\quad\LWR@origquad%
3409 \let\qquad\LWR@origqqquad%
3410 \let\tabular\LWR@origtabular%
3411 \let\endtabular\LWR@origendtabular%
3412 \let\newline\LWR@orignewline%
3413 }% end of outer-most lateximage
```



```

3414 }% end of \begin{lateximage}
3415 {% start of \end{lateximage}
3416 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

3417 {%
3418 \addtocounter{LWR@lateximagedepth}{-1}%
3419 }%

```

if this is the outer-most lateximage:

```

3420 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

3421 \LWR@origendminipage%
3422 \LWR@orignewpage%
3423 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by `pdftotext`:

```

3424 \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.

```

3425 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}{%
3426 {%
3427 \LWR@subinlineimage[%
3428 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
3429 ]%
3430 {lateximage}%
3431 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
3432 {svg}%
3433 {}%
3434 }%
3435 {%
3436 \LWR@subinlineimage[#1]{lateximage}%
3437 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
3438 }%
3439 % \LWR@orignewline% Removed to prevent extra space.

```

Be sure that are doing a paragraph:

```

3440 \LWR@ensuredoingapar%

```

Close the HTML span which has the `pdftotext` comment and also the link to the `.svg` image:

```
3441 \LWR@htmltag{/span}%
3442 \LWR@htmlcomment{End of lateximage}%
3443 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one `lateximage` level:

```
3444 \addtocounter{LWR@lateximagedepth}{-1}%
3445 }% end of outer-most lateximage
3446 }%
3447 \end{warpHTML}
```

```
for PRINT output: 3448 \begin{warpprint}
3449 \newenvironment{lateximage}[1][\begin{group}]{\end{group}}
3450 \end{warpprint}
```

56 center, flushleft, flushright

```
for HTML output: 3451 \begin{warpHTML}
```

Env **center** Replace `center` functionality with CSS tags:

```
3452 \renewenvironment*{center}
3453 {\BlockClass{center}}
3454 {\endBlockClass}
```

Env **flushright**

```
3455 \renewenvironment*{flushright}
3456 {\BlockClass{flushright}}
3457 {\endBlockClass}
```


Env **flushleft**

```
3458 \renewenvironment*{flushleft}
3459 {\BlockClass{flushleft}}
3460 {\endBlockClass}
```

```
3461 \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: 3462 `\begin{warpHTML}`

Options for siunitx:

```
3463 \PassOptionsToPackage{
3464 detect-mode=true,
3465 per-mode=symbol,% fraction is not seen by pdftotext
3466 text-celsius = {\HTMLentity{deg}C},
3467 text-degree = {\HTMLentity{deg}},
3468 }{siunitx}

3469 \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: 3470 `\begin{warpHTML}`

58.1 `\graphicspath`

`\graphicspath` `{\langle path \rangle}`

3471 `\newcommand*\thisgraphicspath{}`

3472 `\renewcommand*\graphicspath[1]{\renewcommand*\thisgraphicspath{#1}}`


`\DeclareGraphicsExtensions` `{\langle list \rangle}`

`\DeclareGraphicsRule` `{\langle \rangle} {\langle \rangle} {\langle \rangle} {\langle \rangle}`

3473 `\renewcommand*\DeclareGraphicsExtensions[1]{}`

3474 `\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.

3475 `\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.

3476 `\newlength{\LWR@igwidth}`

3477 `\newlength{\LWR@igheight}`

3478 `\newcommand*\LWR@igwidthstyle{}`

3479 `\newcommand*\LWR@igheightstyle{}`

3480 `\newcommand*\LWR@igorigin{}`

```

3481 \newcommand*{\LWR@igangle}{}
3482 \newcommand*{\LWR@igxscale}{1}
3483 \newcommand*{\LWR@igyscale}{1}
3484 \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:

```

3485 \define@key{igraph}{width}{%
3486 \setlength{\LWR@igwidth}{#1}%
3487 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
3488 {%

```

Default to use the converted fixed length given:

```

3489 \uselengthunit{PT}%
3490 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

3491 \IfEndWith{#1}{ex}%
3492 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
3493 {}% not ex
3494 \IfEndWith{#1}{em}%
3495 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
3496 {}% not em
3497 \IfEndWith{#1}{\}%
3498 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
3499 {}% not percent
3500 \IfEndWith{#1}{px}%
3501 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
3502 {}% not px
3503 }{}% end of length > 0pt
3504 }

```

If an optional height was given, set an HTML style:

```

3505 \define@key{igraph}{height}{%
3506 \setlength{\LWR@igheight}{#1}%
3507 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}{%
3508 {%

```

Default to use the converted fixed length given:

```

3509 \uselengthunit{PT}%
3510 \renewcommand*{\LWR@igheightstyle}{%
3511 height:\rndprintlength{\LWR@igheight} %

```

3512 }%

If ex or em dimensions were given, use those instead:

```

3513 \IfEndWith{#1}{ex}%
3514 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
3515 {}% not ex
3516 \IfEndWith{#1}{em}%
3517 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
3518 {}% not em
3519 \IfEndWith{#1}{\}%
3520 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
3521 {}% not percent
3522 \IfEndWith{#1}{px}%
3523 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
3524 {}% not px
3525 }{}% end of length > Opt
3526 }

```

Handle origin key:

```

3527 \define@key{igraph}{origin}{%
3528 \renewcommand*{\LWR@igorigin}{#1}%
3529 }

```

Handle angle key:

```

3530 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

3531 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
3532

```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```

3533 \define@key{igraph}{scale}{%
3534 \renewcommand*{\LWR@igxscale}{#1}%
3535 \renewcommand*{\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

3536 \define@key{igraph}{bb}{}
3537 \define@key{igraph}{bblx}{}
3538 \define@key{igraph}{bblly}{}
3539 \define@key{igraph}{bburx}{}
3540 \define@key{igraph}{bbury}{}
3541 \define@key{igraph}{natwidth}{}

```

```

3542 \define@key{igraph}{natheight}{}
3543 \define@key{igraph}{hiresbb}{}
3544 \define@key{igraph}{viewport}{}
3545 \define@key{igraph}{trim}{}
3546 \define@key{igraph}{totalheight}{}
3547 \define@key{igraph}{keepaspectratio}{}
3548 \define@key{igraph}{clip}{}
3549 \define@key{igraph}{draft}{}
3550 \define@key{igraph}{type}{}
3551 \define@key{igraph}{ext}{}
3552 \define@key{igraph}{read}{}
3553 \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.

```

3554 \newcommand*{\LWR@rotstyle}[2]{%
3555   #1transform:rotate(-#2deg);
3556 }

```

`\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.

```

3557 \newcommand*{\LWR@scalestyle}[3]{%
3558   #1transform:scale(#2,#3);
3559 }

```

58.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

3560 \newbool{LWR@infloatrow}
3561 \boolfalse{LWR@infloatrow}

3562 \newcommand*{\LWR@imageextension}{}
3563 \newcommand*{\LWR@expgraphicsfilename}{}

```

`\LWR@includegraphicsb` [*options*] {*filename*}

```
3564
3565 \newcommand*\LWR@includegraphicsb[2] []
3566 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
3567 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
3568 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3569 \ifbool{LWR@infloatrow}%
3570 {}
3571 {% not in a minipage or a floatrow:
3572 \setlength{\linewidth}{6in}%
3573 \setlength{\textwidth}{6in}%
3574 \setlength{\textheight}{9in}%
3575 }%
3576 }{}%
```

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.

```
3577 \edef\LWR@expgraphicsfilename{#2}
3578 \renewcommand*\LWR@imageextension{}%
3579 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.jpg}%
3580 {\renewcommand*\LWR@imageextension}{.jpg}}{}%
3581 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.JPG}%
3582 {\renewcommand*\LWR@imageextension}{.JPG}}{}%
3583 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.png}%
3584 {\renewcommand*\LWR@imageextension}{.png}}{}%
3585 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.PNG}%
3586 {\renewcommand*\LWR@imageextension}{.PNG}}{}%
3587 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.gif}%
3588 {\renewcommand*\LWR@imageextension}{.gif}}{}%
3589 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.GIF}%
3590 {\renewcommand*\LWR@imageextension}{.GIF}}{}%
3591 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svg}%
3592 {\renewcommand*\LWR@imageextension}{.svg}}{}%
3593 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVG}%
3594 {\renewcommand*\LWR@imageextension}{.SVG}}{}%
3595 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.svgz}%
3596 {\renewcommand*\LWR@imageextension}{.svgz}}{}%
```



```

3597 \IfFileExists{\detokenize\expandafter\thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
3598 {\renewcommand*{\LWR@imageextension}{.SVGZ}}{}%

```

Convert a PDF extension to SVG, leave the result in \LWR@strresult:

Must also \detokenize .pdf and .svg comparison strings.

```

3599 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
3600 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
3601 %
3602 \StrSubstitute{\LWR@strresult}%
3603 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%

```

For correct em sizing during the width and height conversions:

```

3604 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

3605 \setlength{\LWR@igwidth}{0pt}%
3606 \setlength{\LWR@igheight}{0pt}%
3607 \renewcommand*{\LWR@igwidthstyle}{}%
3608 \renewcommand*{\LWR@igheightstyle}{}%
3609 \renewcommand*{\LWR@igorigin}{}%
3610 \renewcommand*{\LWR@igangle}{}%
3611 \renewcommand*{\LWR@igxscale}{1}%
3612 \renewcommand*{\LWR@igyscale}{1}%
3613 \renewcommand*{\LWR@igclass}{inlineimage}%

```

Options are in #1

```

3614 \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.

```

3615 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
3616 {% start of href
3617 \LWR@htmltag{% start of image tags
3618 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
3619 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline

```

Only include a style tag if a width, height, angle, or scale was given:

```

3620 \ifthenelse{

```

```

3621 \NOT\equal{\LWR@igwidthstyle}{ } \OR
3622 \NOT\equal{\LWR@igheightstyle}{ } \OR
3623 \NOT\equal{\LWR@igorigin}{ } \OR
3624 \NOT\equal{\LWR@igangle}{ } \OR
3625 \NOT\equal{\LWR@igxscale}{1} \OR
3626 \NOT\equal{\LWR@igyscale}{1}
3627 }%
3628 {\LWR@origtilde{ } style="%
3629 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
3630 {\LWR@igwidthstyle; }}%
3631 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
3632 {\LWR@igheightstyle; }}%
3633 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
3634 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@originnewline}{ }%
3635 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
3636 {%
3637 \LWR@rotstyle{-ms-}{\LWR@igangle}%
3638 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
3639 \LWR@rotstyle{ }{\LWR@igangle%
3640 } }%
3641 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
3642 \NOT\equal{\LWR@igyscale}{1}}%
3643 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
3644 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
3645 \LWR@scalestyle{ }{\LWR@igxscale}{\LWR@igyscale}}}%
3646 " \LWR@originnewline}{ }%

```

Set the class:

```

3647 \LWR@origtilde{ } class="\LWR@igclass" \LWR@originnewline%
3648 }% end of image tags
3649 }% end of href
3650 \endgroup

```

Return to small-sized output:

```

3651 \LWR@origscriptsize
3652 }

```

`\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.

```

3653 \renewcommand*{\includegraphics}
3654 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
3655 \LWR@ensuredoingapar%
3656 \begin{group}%
3657 \LWR@includegraphicsb%
3658 }
```

```
3659 \end{warpHTML}
```

for PRINT output: For print output, accept and then discard the new `class` key:

```
3660 \begin{warpprint}
3661 \define@key{Gin}{class}{}
3662 \end{warpprint}
```

58.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 3663 \begin{warpHTML}

`\LWR@rotboxorigin` Holds the origin key letters.

```
3664 \newcommand*{\LWR@rotboxorigin}{}%
```

`\LWR@originname` $\{\langle letter \rangle\}$

Given one L^AT_EX origin key value, translate into an HTML origin word:

```
3665 \newcommand*{\LWR@originname}[1]{%
3666 \ifthenelse{\equal{#1}{t}}{top}{}%
3667 \ifthenelse{\equal{#1}{b}}{bottom}{}%
3668 \ifthenelse{\equal{#1}{c}}{center}{}%
3669 \ifthenelse{\equal{#1}{l}}{left}{}%
3670 \ifthenelse{\equal{#1}{r}}{right}{}%
3671 }
```

`\LWR@originnames` $\{\langle letters \rangle\}$

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```
3672 \newcommand*{\LWR@originnames}[1]{%
3673 \StrChar{#1}{1}[\LWR@strresult]%
3674 \LWR@originname{\LWR@strresult}%
3675 \StrChar{#1}{2}[\LWR@strresult]%
3676 \LWR@originname{\LWR@strresult}%
3677 }
```

Handle the origin key for `\rotatebox`:

```
3678 \define@key{krotbox}{origin}{%
3679 \renewcommand*{\LWR@rotboxorigin}{#1}%
3680 }
```

These keys are ignored:

```
3681 \define@key{krotbox}{x}{}
3682 \define@key{krotbox}{y}{}
3683 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```
3684 \NewDocumentCommand{\LWR@rotatebox}{0{} m +m}{%
```

Reset the origin to “none-given”:

```
3685 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
3686 \setkeys{krotbox}{#1}%

```

Select `inline-block` so that HTML will transform this span:

```
3687 \LWR@htmltagc{span style="display: inline-block;

```

If an origin was given, translate and print the origin information:

```
3688 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
3689 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
3690 \LWR@rotstyle{-ms-}{#2} %
3691 \LWR@rotstyle{-webkit-}{#2} %
3692 \LWR@rotstyle{}{#2} %
3693 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
3694 \begin{LWR@nestspan}%
3695 #3%

```

Close the span:

```
3696 \LWR@htmltagc{/span}%
3697 \end{LWR@nestspan}%
3698 }
```

`\scalebox` $\{\langle h\text{-scale}\rangle\} [\langle v\text{-scale}\rangle] \{\langle text\rangle\}$

Will `\let\scalebox\LWR@scalebox` at `\LWR@LwarpStart`, in case `\scalebox` was over-written by a later package load.

```
3699 \NewDocumentCommand{\LWR@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
3700 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
3701 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3702 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3703 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
3704 "}}%
```

Print the text to be scaled:

```
3705 \begin{LWR@nestspan}%
3706 #3%
```

Close the span:

```
3707 \LWR@htmltagc{/span}%
3708 \end{LWR@nestspan}%
3709 }
```

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

Will `\let\reflectbox\LWR@reflectbox` at `\LWR@LwarpStart`, in case `\reflectbox` was over-written by a later package load.

```
3710 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
3711 \end{warpHTML}
```

58.5 Null functions

These functions are not supported by `lwarp`'s HTML conversion.

for HTML output: 3712 \begin{warpHTML}

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
3713 \renewcommand{\resizebox}[3]{#3}
```

```
3714 \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 7 on 185 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: 3715 \begin{warpHTML}

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
3716 \AtEndPreamble{
3717 \RequirePackage{cleveref}
3718 \crefname{pagenote}{note}{notes}
3719 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
3720 \AfterEndPreamble{
```

```
\@@setcref {<kindofref>} {<label>}
```

```
3721 \renewcommand*\@@setcref[2]{#1{\ref{#2}}{}{}}
```

```

\@@setcrefrange  {\langle text \rangle} {\langle label \rangle} {\langle label \rangle}

3722 \renewcommand{\@@setcrefrange}[3]{%
3723 #1{\ref{#2}}{\ref{#3}}{}{}{}}

\cpagerefFor  Redefinable word between “page(s)” and the page numbers.

3724 \newcommand*{\cpagerefFor}{for}

\@@setcpageref  {\langle typeofref \rangle} {\langle label \rangle}, where typeofref is “page” or “pages”

3725 \renewcommand*{\@@setcpageref}[2]{%
3726 #1{\cpagerefFor\ \cref{#2}}{}{}%
3727 }

3728 \renewcommand{\@@setcpagerefrange}[3]{%
3729 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}}

Remember and patch some label-related definitions.

3730 \let\LWR@origlabel\label
3731 \let\label\LWR@newlabel
3732 \let\LWR@origref\ref
3733 \let\ref\LWR@newref% \end{ syntax highlighting
3734 \let\LWR@origpageref\pageref
3735 \let\pageref\LWR@newpageref
3736
3737
3738 }% AfterEndPreamble
3739
3740 \end{warpHTML}

```

60 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 3741 `\begin{warpHTML}`

```

Env  picture

3742 \BeforeBeginEnvironment{picture}{
3743
3744 \lateximage}
3745

```

```

3746 \AfterEndEnvironment{picture}{\endlateximage
3747
3748 }

3749 \end{warpHTML}


```

61 Tikzpicture

Pkg **tikz**

Env **tikzpicture** May be used as-is, and its contents will be converted to an image.

For recent versions of Tikz (v3.0.0 — 2013/12/20 and later), include in the document preamble:

 **preamble** `\usetikzlibrary{babel} % v3.0.0 and up`

For older versions of Tikz without the Tikz **babel** library, **lwarp** automatically handles the catcode change for the $\$$ math shift.

catcodes (**lwarp** changes the catcode of $\$$ for its own use. The Tikz **babel** library temporarily changes catcodes back to normal for Tikz's use, restoring them to their **lwarp** purpose while inside each Tikz node.)

for HTML output: `3750 \begin{warpHTML}`

Bool **LWR@tikzbabel** **tikz** v3.0.0 introduced the **babel** library which handles catcode changes. For older versions, **lwarp** must change $\$$'s catcode itself.

```

3751 \newbool{LWR@tikzbabel}
3752 \boolfalse{LWR@tikzbabel}
3753
3754 \AtBeginDocument{
3755 \ifpackageloaded{tikz}{
3756 \ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
3757 {\booltrue{LWR@tikzbabel}}
3758 {\boolfalse{LWR@tikzbabel}}
3759 }{}
3760 }

```

Env **tikzpicture** **tikzpicture** environment is enclosed inside a `\lateximage`.

```

3761 \BeforeBeginEnvironment{tikzpicture}{%
3762
3763 \lateximage
3764 \ifbool{LWR@tikzbabel}
3765 {}

```



```

3766 {\catcode'\$=3} % dollar sign is math shift
3767 }
3768
3769 \AfterEndEnvironment{tikzpicture}{%
3770 \endlateximage
3771
3772 \ifbool{LWR@tikzbabel}
3773 {}
3774 {\catcode'\$=\active}
3775 }


3776 \end{warpHTML}

```

62 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 `\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`.

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.

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

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

for HTML output: 3777 `\begin{warpHTML}`

62.1 Counters and lengths

Ctrl LWR@minipagedepth Used to only reset the line width at the outermost minipage.

```

3778 \newcounter{LWR@minipagedepth}
3779 \setcounter{LWR@minipagedepth}{0}

```

Len \WR@minipagewidth Used to convert the width into printable units.

```

3780 \newlength{\LWR@minipagewidth}

```

Len \WR@minipageheight Used to convert the height into printable units.

```

3781 \newlength{\LWR@minipageheight}

```

Remember the original definitions:

```

3782 \let\LWR@origminipage\minipage
3783 \let\LWR@origendminipage\endminipage

```

62.2 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.

```

3784 \NewDocumentCommand{\LWR@nullrule}{o m m}{ }

```

\LWR@makefntext [*text*]

Adds a paragraph closing tag to the end of each footnote.

```

3785 \newcommand{\LWR@makefntext}[1]{%
3786 \begingroup%
3787 \let\rule\LWR@nullrule%
3788 \textsuperscript{\@thefnmark} #1%
3789 \endgroup%
3790 \LWR@htmltagc{/LWR@tagregularparagraph}%
3791 }

```

\LWR@makefnmark Creates an HTML superscript footnote mark:

```

3792 \newcommand*{\LWR@makefnmark}{\textsuperscript{\@thefnmark}}

```

62.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the L^AT_EX definition and modified to create a `mpfootnotes` div class:

```

3793 \def\LWR@endminipage{%
3794     \par
3795     \unskip
3796     \ifvoid\@mpfootins\else
3797         \vskip\skip\@mpfootins
3798         \normalcolor
3799 \LWR@htmldivclass{mpfootnotes}
3800     \unvbox\@mpfootins
3801 \LWR@htmldivclassend{mpfootnotes}
3802     \fi
3803     \@minipagefalse
3804     \color@endgroup
3805     \egroup
3806     \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.

```

3807 \newcommand*{\LWR@subminipage}{%
3808 \LWR@stoppars
3809 \LWR@origminipage{6in}

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

3810 \LWR@origraggedright%

```

Minipage footnotes appear the bottom of the minipage.

```

3811 \let\footnote\LWR@origfootnote%
3812 \let\@makefnmark\LWR@makefnmark%
3813 \let\@makefnmark\LWR@makefnmark%

```

Resume paragraph tag handling for the contents of the minipage:

```

3814 \LWR@startpars%
3815 }

```

`\LWR@endsubminipage` Closes the subminipage.

```

3816 \newcommand*{\LWR@endsubminipage}{%

```

```

3817 \LWR@stoppars%
3818 \LWR@endminipage% The following empty line is required:
3819
3820 }

```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```

3821 \newbool{LWR@minipagefullwidth}
3822 \boolfalse{LWR@minipagefullwidth}

```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

for HTML output:

```

3823 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
3824 \end{warpHTML}

```

for PRINT output:

```

3825 \begin{warpprint}
3826 \newcommand*{\minipagefullwidth}{}
3827 \end{warpprint}

```

for HTML output:

```

3828 \begin{warpHTML}

```

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```

3829 \newbool{LWR@minipagethispar}
3830 \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.

```

3831 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}
3832 {%

```

Pre-compute the given width and height:

Reset the text area if are starting the outer-most minipage:

```

3833 \LWR@traceinfo{starting minipage}%
3834 \uselengthunit{in}%
3835 \setlength{\LWR@minipagewidth}{#4}%

```

```

3836 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
3837 \addtolength{LWR@minipagewidth}{3em}% room for frames
3838 \setlength{\linewidth}{6in}%
3839 \setlength{\textwidth}{6in}%
3840 \setlength{\textheight}{9in}%
3841 }{}%
3842 \addtocounter{LWR@minipagedepth}{1}%
3843 \setlength{LWR@minipageheight}{\textheight}% default unless specified
3844 \IfValueTF{#2}{\setlength{LWR@minipageheight}{#2}}{}%

```

L^AT_EX wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```

3845 \LWR@stoppars%

```

Create the <div> tag with optional alignment style:

```

3846 \LWR@traceinfo{minipage: creating div class}%
3847 \LWR@htmltag{div class="minipage" style="%
3848 \ifthenelse{\equal{#1}{t}}{vertical-align: bottom ; }{}%
3849 \ifthenelse{\equal{#1}{c}}{vertical-align: middle ; }{}%
3850 \ifthenelse{\equal{#1}{b}}{vertical-align: top ; }{}%
3851 \ifthenelse{\equal{#3}{t}}{justify-content: flex-start ; }{}%
3852 \ifthenelse{\equal{#3}{c}}{justify-content: center ; }{}%
3853 \ifthenelse{\equal{#3}{b}}{justify-content: flex-end ; }{}%
3854 \ifthenelse{\equal{#3}{s}}{justify-content: space-between ; }{}%

```

Print the width and optional height styles:

```

3855 \LWR@traceinfo{minipage: about to print the width}%
3856 \uselengthunit{PT}%
3857 \ifbool{LWR@minipagefullwidth}%
3858 {\boolfalse{LWR@minipagefullwidth}}%
3859 {%
3860 \ifthenelse{\lengthtest{#4}=\linewidth}%
3861 {}%
3862 {width:\rndprintlength{LWR@minipagewidth} ; }%
3863 }%
3864 \LWR@traceinfo{minipage: about to print the height}%
3865 \IfValueTF{#2}{height:\rndprintlength{LWR@minipageheight} ; }{}%
3866 "%}%

```

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.

```

3867
3868 \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.

```
3869 \setlength{\linewidth}{#4}% the original width
3870 \setlength{\textwidth}{6in}%
3871 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
3872 \LWR@origraggedright%
3873 \let\footnote\LWR@origfootnote%
3874 \let\@makefntext\LWR@makefntext%
3875 \let\@makefnmark\LWR@makefnmark%
```

Resume paragraph tag handling for the contents of the minipage:

```
3876 \LWR@startpars%
3877 \LWR@traceinfo{minipage: finished starting the minipage}%
3878 }
```

End the environment with L^AT_EX processing and closing tag:

```
3879 {%
3880 \LWR@stoppars%
3881 \LWR@endminipage% The following empty line is required:
3882
3883 \LWR@htmldivclassend{minipage}%
3884 \vspace{1\baselineskip}% required for subcaption
3885 \addtocounter{LWR@minipagedepth}{-1}%
3886 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
3887 \global\booltrue{LWR@minipagethispar}%
3888 }
```

62.4 Parbox, makebox, framebox, fbox, raisebox

`\parbox` [*<pos>*] [*<height>*] [*<inner-pos>*] {*<width>*} {*<text>*}

A parbox uses the minipage code:

```
3889 \RenewDocumentCommand{\parbox}{O{t} o O{t} m m}
```

```

3890 {
3891 \begin{minipage}[#1][#2][#3]{#4}
3892 #5
3893 \end{minipage}
3894 }

```

`\makebox` [*$\langle width \rangle$*] [*$\langle pos \rangle$*] { *$\langle text \rangle$* }

Width and position are ignored.

```

3895 \RenewDocumentCommand{\makebox}{o o m}{%
3896 \mbox{#3}
3897 }

```

`\framebox` [*$\langle width \rangle$*] [*$\langle pos \rangle$*] { *$\langle text \rangle$* }

Width and position are ignored.

```

3898 \RenewDocumentCommand{\framebox}{o o m}{%
3899 \fbox{#3}
3900 }

```

`\fbox` { *$\langle text \rangle$* }

```

3901 \let\LWR@origfbox\fbox
3902 %
3903 \renewcommand*{\fbox}[1]{%
3904 \InlineClass{framebox}{#1}%
3905 }

```

`\raisebox` { *$\langle raiselen \rangle$* } [*$\langle height \rangle$*] [*$\langle depth \rangle$*] { *$\langle text \rangle$* }

```

3906 \RenewDocumentCommand{\raisebox}{m o o m}{%
3907 #4%
3908 }

```

```

3909 \end{warpHTML}

```

63 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: 3910 \begin{warpHTML}

```
\emph    {\langle text\rangle}

3911 \renewcommand{\emph}[1]{\LWR@htmlspan{em}{#1}}

\textmd   {\langle text\rangle}

3912 \renewcommand{\textmd}[1]{\LWR@htmlspan{textmd}{#1}}

\textbf   {\langle text\rangle}

3913 \renewcommand{\textbf}[1]{\LWR@htmlspan{b}{#1}}

\textrm   {\langle text\rangle}

3914 \renewcommand{\textrm}[1]{\InlineClass{textrm}{#1}}

\textsf   {\langle text\rangle}

3915 \renewcommand{\textsf}[1]{\InlineClass{textsf}{#1}}

\texttt   {\langle text\rangle}

3916 \renewcommand{\texttt}[1]{\LWR@htmlspan{kbd}{#1}}

\textup   {\langle text\rangle}

3917 \renewcommand{\textup}[1]{\LWR@htmlspan{textup}{#1}}

\textit   {\langle text\rangle}

3918 \renewcommand{\textit}[1]{\LWR@htmlspan{i}{#1}}

\textsc   {\langle text\rangle}

3919 \renewcommand{\textsc}[1]{\InlineClass{textsc}{#1}}

\textnormal {\langle text\rangle}

3920 \renewcommand{\textnormal}[1]{\textmd{\textrm{\textup{#1}}}}

\mdseries

3921 \renewcommand*{\mdseries}{}
```


`\bfseries`

3922 `\renewcommand*\bfseries{}`

`\rmfamily`

3923 `\renewcommand*\rmfamily{}`

`\sffamily`

3924 `\renewcommand*\sffamily{}`

`\ttfamily`

3925 `\renewcommand*\ttfamily{}`

`\upshape`

3926 `\renewcommand*\upshape{}`

`\itshape`

3927 `\renewcommand*\itshape{}`

`\scshape`

3928 `\renewcommand*\scshape{}`

`\scshape`

3929 `\renewcommand*\normalfont{}`

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

3930 `\renewcommand{\sp}[1]{\text{^{#1}}}`

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

3931 `\renewcommand{\sb}[1]{\text{_{#1}}}`

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

3932 `\renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}`

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

3933 `\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.

3934 `\AtBeginDocument{\let\up\textsuperscript}`

`\fup` $\{\langle text \rangle\}$ Prints superscript.

Supports `fmtcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

3935 `\AtBeginDocument{\let\fup\textsuperscript}`

`\hfill`

3936 `\renewcommand*{\hfill}{\quad}`

`\hrulefill`

3937 `\renewcommand*{\hrulefill}{\rule{1in}{1pt}}`

`\dotfill`

3938 `\renewcommand*{\dotfill}{\dots}`

3939 `\end{warpHTML}`

64 Skips, spaces, font sizes

for HTML output: 3940 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
3941 \let\LWR@origcomma\,
3942 \let\LWR@origtilde~
3943 \let\LWR@origenskip\enskip
3944 \let\LWR@origquad\quad
3945 \let\LWR@origqqquad\qqquad
3946 \let\LWR@orighspace\hspace
3947 \let\LWR@origrule\rule
```

Direct-formatting space commands become HTML entities:

```
3948 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space

3949 \renewcommand*{~}{\HTMLentity{nbsp}}
3950 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
3951 \let\LWR@orignormalsize\normalsize
3952 \let\LWR@origsmall\small
3953 \let\LWR@origfootnotesize\footnotesize
3954 \let\LWR@origscriptsize\scriptsize
3955 \let\LWR@origtiny\tiny
3956 \let\LWR@origlarge\large
3957 \let\LWR@origLarge\Large
3958 \let\LWR@origLARGE\LARGE
3959 \let\LWR@orighuge\huge
3960 \let\LWR@origHuge\Huge
3961 \renewcommand*{\normalsize}{}
3962 \renewcommand*{\small}{}
3963 \renewcommand*{\footnotesize}{}
3964 \renewcommand*{\scriptsize}{}
3965 \renewcommand*{\tiny}{}
3966 \renewcommand*{\large}{}
3967 \renewcommand*{\Large}{}
3968 \renewcommand*{\LARGE}{}
3969 \renewcommand*{\huge}{}
3970 \renewcommand*{\Huge}{}
3971
3972 \renewcommand*{\onecolumn}{}
3973
3974 \renewcommand{\twocolumn}[1][{}]{
3975
3976 #1
3977
3978 }
```

`\newline` Uses HTML `
` tag

```
3979 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
3980 \let\n newline\LWR@newlinebr
```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```
3981 \NewDocumentCommand{\LWR@endoffline}{s o}
3982 {%
3983 \newline%
3984 }
```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `div` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qqquad` until the end of the paragraph, when the closing `p` tag is created.

`\hspace`
`\enskip`
`\quad`
`\qqquad`

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 `\qqquad`'s HTML output.

```
3985 \newcommand*{\LWR@minipagestartpars}{%
3986 \ifbool{LWR@minipagethispar}%
3987 {%
3988 \LWR@startpars%
3989 }{}%
3990 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qqquad`'s HTML output.

```

3991 \newcommand*{\LWR@minipagestoppars}{%
3992 \ifbool{LWR@minipagethispar}%
3993 {%
3994 \LWR@stoppars%
3995 }{}%
3996 }

```

`\quad` Handles special minipage & horizontal space interactions.

```

3997 \renewcommand*{\quad}{%
3998 \LWR@minipagestoppars%
3999 \HTMLUnicode{2001}%
4000 \LWR@minipagestartpars%
4001 }

```

`\qqquad` Handles special minipage & horizontal space interactions.

```

4002 \renewcommand*{\qqquad}{\quad\quad}

```

`\enskip` Handles special minipage & horizontal space interactions.

```

4003 \renewcommand*{\enskip}{%
4004 \LWR@minipagestoppars%
4005 \HTMLUnicode{2000}%
4006 \LWR@minipagestartpars%
4007 }

```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```

Len \WR@tempheight 4008 \newlength{\LWR@tempwidth}
Len \WR@tempraise 4009 \newlength{\LWR@tempheight}
4010 \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.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```

4011
4012 \NewDocumentCommand{\LWR@hspace}{s m}{%
4013 \setlength{\LWR@tempwidth}{#2}%

```

Only if the width is not zero:

```

4014 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4015 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
4016 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
4017 {%
4018 \HTMLUnicode{2009}% thin breakable space
4019 }%
```

Print the span with the converted width. Not rounded.

```
4020 {%
4021 \uselengthunit{PT}%
4022 \LWR@htmltagc{%
4023 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
4024 }%
4025 \LWR@htmltagc{/span}%
4026 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4027 \LWR@minipagestartpars%
4028 }%
4029 }
```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
4030 \let\hspace\LWR@hspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML br tag.

```
4031 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
4032 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
4033 \renewcommand*{\pagebreak}[1] [] {
4034
4035 }
```

`\nopagebreak` [*num*]

4036 `\renewcommand*\nopagebreak[1] [] {}`

`\enlargethispage` * {*len*}

4037 `\RenewDocumentCommand{\enlargethispage}{s m}{}%`

`\LWR@rule` [*raise*] {*width*} {*height*}

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

4038 `\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:

4039 `\setlength{\LWR@tempwidth}{#2}%`

If it's zero-width then skip the entire rule:

4040 `\ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{}`

4041 `{}% zero- width`

4042 `{}% non-zero width`

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

4043 `\ifthenelse{\lengthtest{\LWR@tempwidth>0pt}\AND%`

4044 `\lengthtest{\LWR@tempwidth<1pt}}%`

4045 `{\setlength{\LWR@tempwidth}{1pt}}{}%`

Likewise with height:

4046 `\setlength{\LWR@tempheight}{#3}%`

4047 `\ifthenelse{\lengthtest{\LWR@tempheight>0pt}\AND%`

4048 `\lengthtest{\LWR@tempheight<1pt}}%`

4049 `{\setlength{\LWR@tempheight}{1pt}}{}%`

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

4050 `\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.

```
4051 \uselengthunit{PT}%
4052 \LWR@htmltagc{%
4053 span
4054 style=" %
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
4055 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
4056 width:\printlength{\LWR@tempwidth}; %
4057 height:\printlength{\LWR@tempheight}; %
```

The raise height is converted to a CSS transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a L^AT_EX length allows a typical L^AT_EX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a L^AT_EX length limits the allowable syntax. To do: A superior method would compute a ratio of L^AT_EX ex height, then print that to HTML with an ex unit.

```
4058 \IfValueTF{#1}%
4059 {%
4060 \setlength{\LWR@tempraise}{Opt-#1}%
4061 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
4062 \LWR@orignewline%
4063 -ms-transform: translate(Opt,\printlength{\LWR@tempraise}); %
4064 \LWR@orignewline%
4065 -webkit-transform: translate(Opt,\printlength{\LWR@tempraise}); %
4066 \LWR@orignewline%
4067 transform: translate(Opt,\printlength{\LWR@tempraise}); %
4068 \LWR@orignewline%
4069 }{}%
```

Display inline-block to place the span inline with the text:

```
4070 display:inline-block;"%
4071 }%
4072 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
4073 \LWR@minipagestartpars%
```



```
4074 }% non-zero width
4075 }
```

```
\rule  [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```
4076 \let\rule\LWR@rule
```

```
4077 \end{warpHTML}
```

65 \phantomsection

for HTML output: 4078 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
4079 \newcommand*{\phantomsection}{\section*{}}
```

```
4080 \end{warpHTML}
```

66 \LaTeX and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally `\let` in `\LWR@LwarpStart`.

For CSS conversions, see:

<http://edward.oconnor.cx/2007/08/tex-poshlet>

<http://nitens.org/taraborelli/texlogo>

66.1 HTML logos

for HTML output: 4081 \begin{warpHTML}

```
\TeX TEX
```

`latexlogo` is a CSS class used to properly typeset the E and A in \LaTeX and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in \LaTeX , \LuaTeX , \ConTeXt , etc.

```
4082 \newcommand*{\LWR@TeX}
4083 {\InlineClass{latexlogofont}%
4084 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

$\backslash\text{\LaTeX}$ \LaTeX , $\text{\LaTeX} 2_{\epsilon}$

$\backslash\text{\LaTeXe}$

```
4085 \newcommand*{\LWR@LaTeX}
4086 {\InlineClass{latexlogofont}%
4087 {\InlineClass{latexlogo}%
4088 {L\textsuperscript{a}T\textsubscript{e}X}}
4089
4090 \renewcommand*{\LaTeXe}
4091 {\LaTeX\InlineClass{latexlogofont}%
4092 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

$\backslash\text{\LuaTeX}$ \LuaTeX , \LuaLaTeX

$\backslash\text{\LuaLaTeX}$

```
4093 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
4094 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

$\backslash\text{\XeTeX}$ \XeTeX , \XeLaTeX

$\backslash\text{\XeLaTeX}$

`xetexlogo` is a CSS class which aligns the backwards E in \XeTeX and spaces \TeX appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in \XeLaTeX and spaces \LaTeX appropriately.

```
4095 \newcommand*{\Xe}
4096 {X\textsubscript{\HTMLUnicode{18e}}}
4097 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
4098 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

$\backslash\text{\ConTeXt}$ \ConTeXt

```
4099 \newcommand*{\LWR@ConTeXt}
4100 {\InlineClass{latexlogofont}{Con}\TeX}%
4101 \InlineClass{latexlogofont}{t}}
```

$\backslash\text{\BibTeX}$ \BibTeX , *MakeIndex*

$\backslash\text{\MakeIndex}$

```
4102 \providecommand*{\BibTeX}
4103 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
```

```

4104
4105 \newcommand*{\MakeIndex}
4106 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS` \mathcal{AMS}

`amslogo` is a CSS class used for the \mathcal{AMS} logo.

```

4107 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{\}
4108 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX` `MiKTeX`

```

4109 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK\TeX}}

```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the `LyX`logo.

```

4110 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}

```

```

4111 \end{warpHTML}

```

66.2 Print logos

```

for PRINT output: 4112 \begin{warpprint}
4113 \newcommand*{\XeTeXrevE}
4114 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
4115 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
4116 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
4117 \providecommand*{\AmS}{%
4118 \leavevmode\hbox{$\mathcal{A}$\kern-.2em\lower.376ex%
4119 \hbox{$\mathcal{M}$}\kern-.2em\mathcal{S}$}}
4120 \newcommand*{\LyX}{\textsf{LyX}}
4121 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
4122 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
4123 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
4124 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
4125 \providecommand*{\ConTeXt}{\mbox{Con\TeXt}}
4126 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
4127 \end{warpprint}

```

67 \AtBeginDocument, \AtEndDocument

for HTML output: 4128 \begin{warpHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the
\LWR@LwarpEnd document.

4129 \AfterEndPreamble{\LWR@LwarpStart}

4130 \AtEndDocument{\LWR@LwarpEnd}

4131 \end{warpHTML}

68 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:

```
4132 \providecommand*\HomeHTMLFileName{\BaseJobname}
4133 \providecommand*\HTMLFileName{}
4134 \providecommand*\UseLatexmk{false}
```

```
for PRINT output: 4135 \begin{warpprint}
4136 \ifcsdef{LWR@file}{\newwrite{LWR@file}}
4137 \immediate\openout\LWR@file=lwarpmk.conf
4138 \ifwindows
4139 \immediate\write\LWR@file{opsystem = "Windows"}
4140 \else
4141 \immediate\write\LWR@file{opsystem = "Unix"}
4142 \fi
4143 \ifPDFTeX
4144 \immediate\write\LWR@file{latexname = "pdflatex"}
4145 \fi
4146 \ifXeTeX
4147 \immediate\write\LWR@file{latexname = "xelatex"}
4148 \fi
4149 \ifLuaTeX
4150 \immediate\write\LWR@file{latexname = "lualatex"}
4151 \fi
4152 \immediate\write\LWR@file{sourcename = "\jobname"}
4153 \immediate\write\LWR@file{%
4154 homehtmlfilename = "\HomeHTMLFileName"%
4155 }
4156 \immediate\write\LWR@file{htmlfilename = "\HTMLFileName"}
4157 \immediate\write\LWR@file{uselatexmk = "\UseLatexmk"}
4158 \immediate\closeout\LWR@file
4159 \end{warpprint}
```

69 Trademarks

- 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

70 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}

```

70.1 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 }{}

```

70.2 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 }{}

```

70.3 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 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.toclstlisting {margin-left: 0em}
104
105
106 body {
107     font-family: "DejaVu Serif", "Bitstream Vera Serif",
108         "Lucida Bright", Georgia, serif;
109     background: #FAF7F4 ;
110     color: black ;
111     margin: 0em ;
112     padding: 0em ;
113     font-size: 100% ;
114     line-height: 1.2 ;
115 }
116
117 p {margin: 1.5ex 0em 1.5ex 0em ;}
118
119 /* Holds a section number to add space between it and the name */
120 span.sectionnumber { margin-right: .6em }
121
122 /* Inserted in front of index lines */
123 span.indexitem {margin-left: 0em}
124 span.indexsubitem {margin-left: 2em}
125 span.indexsubsubitem {margin-left: 4em}
126
127 div.hidden { display: none ; }
128
129 kbd {
130     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
131         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
132         "Courier New", monospace;
133     font-size: 100% ;
```

```
134 }
135
136 span.strong { font-weight: bold; }
137
138 span.textmd { font-weight: normal; }
139
140 span.textsc { font-variant: small-caps; }
141
142 span.textup { font-variant: normal; }
143
144 span.textrm {
145     font-family: "DejaVu Serif", "Bitstream Vera Serif",
146     "Lucida Bright", Georgia, serif;
147 }
148
149 span.textsf {
150     font-family: "DejaVu Sans", "Bitstream Vera Sans",
151     Geneva, Verdana, sans-serif ;
152 }
153
154 span.attribution {
155     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
156 }
157
158 span.citetitle {
159     margin-left: 1em ; font-size: 80% ; font-style: oblique;
160 }
161
162 span.poemtitle {
163     font-size: 120% ; font-weight: bold;
164 }
165
166 blockquote {
167     margin-left: 0px ;
168     margin-right: 0px ;
169 }
170
171 blockquote p {
172     line-height: 1.5;
173     text-align: left ;
174     font-size: .85em ;
175     margin-left: 3em ;
176     margin-right: 3em ;
177 }
178
179 blockquotation {
180     margin-left: 0px ;
181     margin-right: 0px ;
182 }
183
```

```
184 blockquotation p {
185   line-height: 1.5;
186   text-align: left ;
187   font-size: .85em ;
188   margin-left: 3em ;
189 margin-right: 3em ;
190 }
191
192 div.epigraph {
193   line-height: 1.2;
194   text-align: left ;
195   padding: 3ex 1em 0ex 1em ;
196 /*   margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
197   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
198 /*   margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
199   font-size: .85em ;
200   max-width: 27em ;
201 }
202
203
204
205 div.epigraphsource{
206   text-align:right ;
207   margin-left:auto ;
208 /*   max-width: 50% ; */
209   border-top: 1px solid #A0A0A0 ;
210   padding-bottom: 3ex ;
211   line-height: 1.2;
212 }
213
214 div.epigraph p { padding: .5ex ; margin: 0ex ;}
215 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
216
217
218
219
220 html body {
221   margin: 0 ;
222   line-height: 1.2;
223 }
224
225
226 body div {
227   margin: 0ex;
228 }
229
230
231 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
232 {
233   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
```

```
234         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
235         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
236         "Times New Roman", serif;
237     font-style: normal ;
238     font-weight: bold ;
239     text-align: left ;
240 }
241
242 h1 { /* title of the entire website, used on each page */
243     text-align: center ;
244     font-size: 2.5em ;
245     padding: .4ex 0em 0ex 0em ;
246 }
247 h2 { font-size: 2.25em }
248 h3 { font-size: 2em }
249 h4 { font-size: 1.75em }
250 h5 { font-size: 1.5em }
251 h6 { font-size: 1.25em }
252 span.paragraph {font-size: 1em ; font-variant: normal ;
253     margin-right: 1em ; }
254 span.subparagraph {font-size: 1em ; font-variant: normal ;
255     margin-right: 1em ; }
256
257
258
259 /* Title of the file */
260 h1 {
261     margin: 0ex 0em 0ex 0em ;
262     line-height: 1.3;
263     text-align: center ;
264 }
265
266 /* Part */
267 h2 {
268     margin: 1ex 0em 1ex 0em ;
269     line-height: 1.3;
270     text-align: center ;
271 }
272
273 /* Chapter */
274 h3 {
275     margin: 3ex 0em 1ex 0em ;
276     line-height: 1.3;
277 }
278
279 /* Section */
280 h4 {
281     margin: 3ex 0em 1ex 0em ;
282     line-height: 1.3;
283 }
```

```
284
285 /* Sub-Section */
286 h5 {
287     margin: 3ex 0em 1ex 0em ;
288     line-height: 1.3;
289 }
290
291 /* Sub-Sub-Section */
292 h6 {
293     margin: 3ex 0em 1ex 0em ;
294     line-height: 1.3;
295 }
296
297
298 div.titlepage {
299     text-align: center ;
300 }
301
302 .footnotes {
303     font-size: .85em ;
304     margin: 3ex 1em 0ex 1em ;
305     padding-bottom: 1ex ;
306 }
307
308 .marginpar {
309     width:20%; float:right;
310     text-align:left;
311     margin: 1ex 0.5em 1ex 1em ;
312     padding: 1ex 0.5em 1ex 0.5em ;
313     font-size: 85% ;
314     border-top: 1px solid silver ;
315     border-bottom: 1px solid silver ;
316     overflow-x: auto;
317 }
318
319 .marginpar br { margin-bottom: 2ex ; }
320
321 div.marginblock {
322     width:50%; float:right;
323     text-align:left;
324     margin: 1ex 0.5em 1ex 1em ;
325     padding: 1ex 0.5em 1ex 0.5em ;
326     overflow-x: auto;
327 }
328
329 div.marginblock div.minipage p { font-size: 85%}
330
331 div.marginblock br { margin-bottom: 2ex ; }
332
333
```

```
334 section.textbody div.footnotes{
335     margin: 3ex 0em 0ex 0em ;
336     border-bottom: 2px solid silver ;
337 }
338
339 .footnoteheader {
340     border-top: 2px solid silver ;
341     margin-top: 3ex ;
342     padding-top: 1ex ;
343     font-weight: bold ;
344 }
345
346 .mpfootnotes {
347     text-align: left ;
348     font-size: .85em ;
349     margin: 3ex 0em 3ex 0em ;
350     border-top: 1px solid silver ;
351     border-bottom: 1px solid silver ;
352 /*     padding-top: 1ex ; */
353 }
354
355 /* Remove footnote top border in the title page. */
356 div.titlepage div.mpfootnotes {
357     border-top: none ;
358 }
359
360
361
362 ol {
363     margin: 1ex 1em 1ex 0em;
364     line-height: 1.2;
365 }
366
367 ul, body dir, body menu {
368     margin: 1ex 1em 1ex 0em;
369     line-height: 1.2;
370 }
371
372 li { margin: 0ex 0em 1ex 0em; }
373
374 html {
375     margin: 0;
376     padding: 0;
377 }
378
379 .programlisting {
380     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
381                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
382                 "Courier New", monospace;
383     margin: 1ex 0ex 1ex 0ex ;
```



```
384 padding: .5ex 0pt .5ex 0pt ;
385 overflow-x: auto;
386 }
387
388 section.textbody>pre.programlisting {
389 border-top: 1px solid silver ;
390 border-bottom: 1px solid silver ;
391 }
392
393
394 .inlineprogramlisting {
395 font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
396             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
397             "Courier New", monospace;
398 overflow-x: auto;
399 }
400
401
402 div.abstract {
403 margin: 2em 5% 2em 5% ;
404 padding: 1ex 1em 1ex 1em ;
405 /* font-weight: bold ; */
406 font-size: 90% ;
407 }
408
409 div.abstract dl {line-height:1.5;}
410 div.abstract dt {color:#304070;}
411
412 div.abstracttitle{
413 font-family: "URW Classico", Optima, "Linux Biolinum O",
414             "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
415             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
416 font-weight:bold;
417 font-size:1.25em;
418 text-align: center ;
419 }
420
421
422 .verbatim {
423 overflow-x: auto ;
424 }
425
426 .alltt {
427 overflow-x: auto ;
428 }
429
430
431 .bverbatim {
432 margin: 1ex 0pt 1ex 0pt ;
433 padding: .5ex 0pt .5ex 0pt ;
```

```
434     overflow-x: auto ;
435 }
436
437 .lverbatim {
438     margin: 1ex 0pt 1ex 0pt ;
439     padding: .5ex 0pt .5ex 0pt ;
440     overflow-x: auto ;
441 }
442
443 .fancyvrb {
444     font-size:.85em ;
445     margin: 3ex 0pt 3ex 0pt
446 }
447
448 .fancyvrblabel {
449     font-weight:bold;
450     text-align: center ;
451 }
452
453
454 .verse {
455     font-family: "Linux Libertine Mono O", "Lucida Console",
456                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
457                 "Liberation Mono", "FreeMono", "Andale Mono",
458                 "Nimbus Mono L", "Courier New", monospace;
459     margin-left: 1em ;
460 }
461
462
463 div.singlespace { line-height: 1.2 ; }
464 div.onehalfspace { line-height: 1.5 ; }
465 div.doublespace { line-height: 2 ; }
466
467
468 .theorem {
469     background: #e0e0e0;
470     padding: 1ex ;
471     margin: 1em ;
472     border: 1px solid silver ;
473 }
474
475 .theoremlabel { font-weight: bold ; }
476
477 .theoremcontents { font-style: italic; margin-left: 1em ; }
478
479
480 /* Minipage environments, vertically aligned to top, center, bottom: */
481 .minipage {
482     /* display: inline-block ; */
483     /* Mini pages which follow each other will be tiled. */
```

```
484     margin: .25em .25em .25em .25em;
485     padding: .25em .25em .25em .25em;
486     display: inline-flex;
487     flex-direction: column ;
488     overflow: auto;
489 }
490
491 /* Paragraphs in the flexbox did not collapse their margins. */
492 /* Have not yet researched this. */
493 .minipage p {margin: .75ex 0em .75ex 0em ;}
494
495
496
497 .framebox {
498     margin: 0ex ;
499     padding: 0ex ;
500     border: 1px solid black;
501     border-radius: 0px ;
502     padding: .3ex .2em 0ex .2em ;
503     margin: .1ex ;
504     display: inline-block ;
505 }
506
507
508 .mdframed {
509 /*     padding: 0ex ; */
510 /*     border: 1px solid black; */
511 /*     border-radius: 0px ; */
512     padding: 0ex ;
513     margin: 3ex 5% 3ex 5% ;
514 /*     display: inline-block ; */
515 }
516
517 .mdframed p { padding: 0ex .5em 0ex .5em ; }
518
519 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
520
521 .mdframedtitle {
522     padding: .5em ;
523     display: block ;
524     font-size: 130%
525 }
526
527 .mdframedsubtitle {
528     padding: 0ex .5em 0ex .5em ;
529     display: block ;
530     font-size: 115% ;
531 }
532
533 .mdframedsubsubtitle {
```

```
534     padding: 0ex .5em 0ex .5em ;
535     display: block ;
536 }
537
538 .mdtheorem {
539     padding: 0ex .5em 0ex .5em ;
540     margin: 3ex 5% 3ex 5% ;
541     /* display: inline-block ; */
542 }
543
544
545
546
547 dl {
548     margin: 1ex 2em 1ex 0em;
549     line-height: 1.3;
550 }
551
552 dl dt {
553     margin-top: 1ex;
554     font-weight: bold;
555 }
556
557 dl dd p { margin-top: 0em; }
558
559
560 nav.toc, nav.lof, nav.lot, nav.lol {
561     font-family: "URW Classico", Optima, "Linux Biolinum O",
562         "DejaVu Sans", "Bitstream Vera Sans",
563         Geneva, Verdana, sans-serif ;
564     margin-bottom: 4ex ;
565 }
566
567 nav.toc p, nav.lof p, nav.lot p, nav.lol p {
568     line-height: 1.2 ;
569     margin-top:.5ex ;
570     margin-bottom:.5ex;
571     font-size: .9em ;
572 }
573
574
575
576 img, img.hyperimage, img.borderimage {
577     max-width: 600px;
578     border: 1px solid silver;
579     box-shadow: 3px 3px 3px #808080 ;
580     padding: .5% ;
581     margin: .5% ;
582     background: none ;
583 }
```

```
584
585 img.inlineimage{
586     padding: 0px ;
587     box-shadow: none ;
588     border: none ;
589     background: none ;
590     margin: 0px ;
591     display: inline-block ;
592     border-radius: 0px ;
593 }
594
595 img.logoimage{
596     max-width: 300px ;
597     box-shadow: 3px 3px 3px #808080 ;
598     border: 1px solid black ;
599     background:none ;
600     padding:0 ;
601     margin:.5ex ;
602     border-radius: 10px ;
603 }
604
605
606 .section {
607 /*
608     To have each section float relative to each other:
609 */
610 /*
611     display: block ;
612     float: left ;
613     position: relative ;
614     background: white ;
615     border: 1px solid silver ;
616     padding: .5em ;
617 */
618     margin: 0ex .5em 0ex .5em ;
619     padding: 0 ;
620 }
621
622
623 figure {
624     margin: 3ex auto 3ex auto ;
625     padding: 1ex 1em 1ex 1em ;
626     overflow-x: auto ;
627 }
628
629
630 /* To automatically center images in figures: */
631 /*
632 figure img.inlineimage {
633     margin: 0ex auto 0ex auto ;
```

```
634     display: block ;
635 }
636 */
637
638 /* To automatically center minipages in figures: */
639 /*
640 figure div.minipage, figure div.minipage div.minipage {
641     margin: 1ex auto 1ex auto ;
642     display: block ;
643 }
644 */
645
646 figure div.minipage p { font-size: 85% ; }
647
648 figure.subfigure, figure.subtable {
649     display: inline-block ; margin: 3ex 1em 3ex 1em ;
650 }
651
652 div.floatrow { text-align: center; }
653
654 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
655
656 div.floatfoot { font-size: .85em ;
657     border-top: 1px solid silver ; line-height: 1.2 ; }
658
659 figcaption , .lstlistingtitle {
660     font-size: .85em ;
661     text-align: center ;
662     font-weight: bold ;
663     margin-top: 1ex ;
664     margin-bottom: 1ex ;
665 }
666
667 figure.subfigure figcaption, figure.subtable figcaption {
668     border-bottom: none ; background: none ;
669 }
670
671 div.nonfloatcaption {
672     margin: 1ex auto 1ex auto ;
673     font-size: .85em ;
674     text-align: center ;
675     font-weight: bold ;
676 }
677
678 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
679 figure div.floatrow div.minipage figcaption {
680     border: none ;
681     background: none ;
682 }
683
```

```

684
685 table {
686     margin: 1ex auto 1ex auto ;
687     border-collapse: collapse ;
688     border-spacing: 0px ;
689     line-height: 1.3 ;
690 }
691
692 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
693     margin-bottom: 0ex ; } /* for \hline */
694
695 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
696     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
697
698 td {padding: 1ex .5em 1ex .5em ;}
699
700 table td.tdl { text-align: left ; vertical-align: middle ; }
701 table td.tdc { text-align: center ; vertical-align: middle ; }
702 table td.tdr { text-align: right ; vertical-align: middle ; }
703 table td.tdp { text-align: left ; vertical-align: bottom ; }
704 table td.tdm { text-align: left ; vertical-align: middle ; }
705 table td.tdb { text-align: left ; vertical-align: top ; }
706 table td.tdP { text-align: center ; vertical-align: bottom ; }
707 table td.tdM { text-align: center ; vertical-align: middle ; }
708 table td.tdB { text-align: center ; vertical-align: top ; }
709 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
710     vertical-align: middle ; } /* for cmdrule */
711 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
712     vertical-align: middle ; }
713 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
714     vertical-align: middle ; }
715 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
716     vertical-align: bottom ; }
717 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
718     vertical-align: middle ; }
719 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
720     vertical-align: top ; }
721 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
722     vertical-align: bottom ; }
723 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
724     vertical-align: middle ; }
725 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
726     vertical-align: top ; }
727
728 /* Margins of paragraphs inside table cells: */
729 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
730     padding-bottom: 1ex ; margin: 0ex ; }
731 td.tdm p , td.tdrrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
732     padding-bottom: 1ex ; margin: 0ex ; }
733 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;

```

```
734     padding-bottom: 1ex ; margin: 0ex ; }
735
736 td.tdp , td.tdprule , td.tdP , td.tdPrule
737     { padding: 0ex .5em 0ex .5em ; }
738 td.tdm , td.tdmrule , td.tdM , td.tdMrule
739     { padding: 0ex .5em 0ex .5em ; }
740 td.tdb , td.tdbrule , td.tdB , td.tdBrule
741     { padding: 0ex .5em 0ex .5em ; }
742
743
744 /* table notes: */
745 .tnotes {
746     margin: 0ex 5% 1ex 5% ;
747     padding: 0.5ex 1em 0.5ex 1em;
748     font-size:.85em;
749     text-align: left ;
750 }
751
752 .tnotes dl dt p {margin-bottom:0px;}
753
754 .tnoteitemheader {margin-right: 1em;}
755
756
757
758 /* center, flushleft, flushright environments */
759 div.center{text-align:center;}
760 div.center table {margin-left:auto;margin-right:auto;}
761 div.flushleft{text-align:left;}
762 div.flushleft table {margin-left:0em ; margin-right:auto;}
763 div.flushright{text-align:right;}
764 div.flushright table {margin-left:auto ; margin-right: 0em ;}
765
766
767
768
769 /* program listing callouts: */
770 span.callout {
771     font-family: "DejaVu Sans", "Bitstream Vera Sans",
772         Geneva, Verdana, sans-serif ;
773     border-radius: .5em;
774     background-color:black;
775     color:white;
776     padding:0px .25em 0px .25em;
777     margin: 0 ;
778     font-weight: bold;
779     font-size:.72em ;
780 }
781
782 div.programlisting pre.verbatim span.callout{
783 font-size: .85em ;
```



```
784 }
785
786
787
788
789
790 div.published
791 {
792     text-align: center ;
793     font-variant: normal ;
794     font-style: italic ;
795     font-size: 1em ;
796     margin: 3ex 0em 3ex 0em ;
797 }
798
799 div.subtitle
800 {
801     text-align: center ;
802     font-variant: normal ;
803     font-style: italic ;
804     font-size: 1.25em ;
805     margin: 3ex 0em 3ex 0em ;
806 }
807
808 div.subtitle p { margin: 1ex ; }
809
810 div.author
811 {
812     font-variant: normal ;
813     font-style: normal ;
814     font-size: 1em ;
815     margin: 3ex 0em 3ex 0em ;
816 }
817
818 div.author table {
819     margin: 3ex auto 0ex auto ;
820     background: none ;
821 }
822
823 div.author table tbody tr td { padding: .25ex ; }
824
825 span.affiliation {font-size: .85em ; font-variant: small-caps; }
826
827 div.titledate {
828     text-align: center ;
829     font-size: .85em ;
830     font-style: italic;
831     margin: 6ex 0em 6ex 0em ;
832 }
833
```

```
834
835 nav.topnavigation{
836     text-align: left ;
837     padding: 0.5ex 1em 0.5ex 1em ;
838 /*     margin: 2ex 0em 3ex 0em ; */
839     margin: 0 ;
840     border-bottom: 1px solid silver ;
841     border-top: 1px solid silver ;
842     clear:right ;
843 }
844
845 nav.botnavigation{
846     text-align: left ;
847     padding: 0.5ex 1em 0.5ex 1em ;
848 /*     margin: 3ex 0em 2ex 0em ; */
849     margin: 0 ;
850     border-top: 1px solid silver ;
851     border-bottom: 1px solid silver ;
852     clear:right ;
853 }
854
855
856 header{
857     line-height: 1.2 ;
858     font-size: 1em ;
859 /*     border-bottom: 2px solid silver ; */
860     margin: 0px ;
861     padding: 0ex 1em 0ex 1em ;
862     text-align:center ;
863 }
864
865 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
866
867
868 footer{
869     font-size: .85em ;
870     line-height: 1.2 ;
871     margin-top: 1ex ;
872     border-top: 2px solid silver ;
873     padding: 2ex 1em 2ex 1em ;
874     clear:right ;
875     text-align:left ;
876 }
877
878
879 a.linkhome { font-weight:bold ; font-size: 1em ;}
880
881
882 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
883
```

```
884 img.lateximage{
885     padding: 0px 0px 0px 0px ;
886     box-shadow: none ;
887     border: none ;
888     background: none ;
889     margin: 0px 0px -.15ex 0px ;
890     /* pdfcrop leaves a slight margin, adjust to baseline */
891     max-width: 100% ;
892     border-radius: 0ex ;
893     border: none ;
894 }
895
896
897
898 nav.sidetoc {
899     font-family: "DejaVu Serif", "Bitstream Vera Serif",
900         "Lucida Bright", Georgia, serif;
901     float:right ;
902     width: 20%;
903     border-left: 1px solid silver;
904     border-top: 1px solid silver;
905     border-bottom: 1px solid silver;
906 /*     border-top: 2px solid #808080 ; */
907     background: #FAF7F4 ;
908     padding: 2ex 0em 2ex 1em ;
909     margin: 0ex 0em 2ex 1em ;
910     font-size:.9em ;
911     border-radius: 20px 0px 0px 20px ;
912 }
913
914 div.sidetoccontents {
915 /*     border-top: 1px solid silver ; */
916     overflow-y: auto ;
917     width: 100% ;
918     text-align: left ;
919 }
920
921 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
922     text-indent: 0 ; }
923 nav.sidetoc p a {color:black ; font-size: .7em ;}
924 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
925     border-bottom: 1px solid silver ; }
926 nav.sidetoc a:hover {text-decoration: underline ; }
927
928
929
930 section.textbody { margin: 0ex 1em 0ex 1em ;}
931
932
933 div.multicolsheading { -webkit-column-span: all;
```

```
934     -moz-column-span: all; column-span: all; }
935 div.multicols { -webkit-columns: 3 380px ;
936     -moz-columns: 3 380px ; columns: 3 380px ; }
937 div.multicols p {margin-top: 0ex}
938
939 /* Used to support algorithmicx: */
940 span.floatright { float: right ; }
941
942
943
944 /*
945 For CSS LaTeX and related logos:
946 Based on:
947 http://edward.oconnor.cx/2007/08/tex-poshlet
948 http://nitens.org/taraborelli/texlogo
949 */
950
951 .latexlogofont {
952     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
953         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
954     font-variant: normal ;
955 }
956
957 .latexlogo {
958     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
959         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
960     letter-spacing: .03em ;
961     font-size: 1.1em;
962 }
963
964 .latexlogo sup {
965     text-transform: uppercase;
966     letter-spacing: .03em ;
967     font-size: 0.85em;
968     vertical-align: 0.15em;
969     margin-left: -0.36em;
970     margin-right: -0.15em;
971 }
972
973 .latexlogo sub {
974     text-transform: uppercase;
975     vertical-align: -0.5ex;
976     margin-left: -0.1667em;
977     margin-right: -0.125em;
978     font-size: 1em;
979 }
980
981 .xetexlogo {
982     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
983         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
```

```
984     letter-spacing: .03em ;
985     font-size: 1.1em;
986 }
987
988 /* A smaller gap between Xe and Tex v.s. LaTeX: */
989 .xetexlogo sub {
990     text-transform: uppercase;
991     vertical-align: -0.5ex;
992     margin-left: -0.0667em;
993     margin-right: -0.2em;
994     font-size: 1em;
995     letter-spacing: .03em ;
996 }
997
998 /* A large gap between Xe and LaTeX v.s. TeX: */
999 .xelatexlogo sub {
1000     text-transform: uppercase;
1001     vertical-align: -0.5ex;
1002     margin-left: -0.0667em;
1003     margin-right: -.05em;
1004     font-size: 1em;
1005     letter-spacing: .03em ;
1006 }
1007
1008 .amslogo {
1009     font-family: "TeXGyreChorus","URW Chancery L",
1010         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1011         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1012         "Hoefler Text", Times, "Times New Roman", serif;
1013     font-style: italic;
1014 }
1015
1016 .lyxlogo {
1017     font-family: "URW Classico", Optima, "Linux Biolinum O",
1018         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1019     Verdana, sans-serif ;
1020 }
1021
1022
1023
1024
1025 /* Only display top and bottom navigation if a small screen: */
1026 /* Hide the sidetoc if a small screen: */
1027 nav.topnavigation { display:none; }
1028 nav.botnavigation { display:none; }
1029
1030 @media screen and (max-width: 45em) {
1031     /*     nav.sidetoc {display:none;} */
1032     nav.sidetoc {
1033         float: none ;
```

```

1034         width: 100% ;
1035         margin: 5ex 0px 5ex 0px ;
1036         padding: 0 ;
1037         border-radius: 0 ;
1038         border-bottom: 1px solid black ;
1039         border-top: 1px solid black ;
1040         box-shadow: none ;
1041     }
1042 /*     nav.topnavigation { display:block } */
1043     nav.botnavigation { display:block }
1044 }
1045
1046 @media print {
1047     body {
1048         font-family: "Linux Libertine O",
1049             "DejaVu Serif", "Bitstream Vera Serif",
1050             "Liberation Serif", "Nimbus Roman No 9 L",
1051             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1052     }
1053     nav.sidetoc { display:none; }
1054     nav.topnavigation { display: none; }
1055     nav.botnavigation { display: none; }
1056 }
1057
1058 @media handheld {
1059     nav.sidetoc { display:none; }
1060     nav.topnavigation { display:block }
1061     nav.botnavigation { display:block }
1062 }
1063
1064 @media projection {
1065     nav.sidetoc { display:none; }
1066     nav.topnavigation { display:block }
1067     nav.botnavigation { display:block }
1068 }
1069 \end{VerbatimOut}
1070 % \end{Verbatim}% for syntax highlighting

```

70.4 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.

```

1071 \begin{VerbatimOut}{lwarp_sagebrush.css}
1072 @import url("lwarp.css") ;

```

```
1073
1074
1075 A:link {color:#105030 ; text-decoration: none ; }
1076 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1077 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1078 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1079
1080
1081
1082 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1083 {
1084     font-family: "URW Classico", Optima, "Linux Biolinum O",
1085                 "Linux Libertine O", "Liberation Serif",
1086                 "Nimbus Roman No 9 L", "FreeSerif",
1087                 "Hoefler Text", Times, "Times New Roman", serif;
1088     font-variant: small-caps ;
1089 font-weight: normal ;
1090     color: #304070 ;
1091     text-shadow: 2px 2px 3px #808080;
1092 }
1093
1094 h1 { /* title of the entire website, used on each page */
1095     font-variant: small-caps ;
1096     color: #304070 ;
1097     text-shadow: 2px 2px 3px #808080;
1098     background-color: #F7F7F0 ;
1099     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
1100 }
1101
1102 h1 {
1103     border-bottom: 1px solid #304070;
1104     border-top: 2px solid #304070;
1105 }
1106
1107 h2 {
1108     border-bottom: 1px solid #304070;
1109     border-top: 2px solid #304070;
1110     background-color: #F7F7F0 ;
1111     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
1112 }
1113
1114
1115
1116 div.abstract {
1117     background: #f5f5eb ;
1118     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1119
1120     border: 1px solid silver;
1121     border-radius: 1em ;
1122 }
```

```
1123
1124 div.abstract dl {line-height:1.5;}
1125 div.abstract dt {color:#304070;}
1126
1127 div.abstracttitle{
1128     font-family: "URW Classico", Optima, "Linux Biolinum O",
1129         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1130         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1131     font-weight:bold;
1132     font-variant: small-caps ;
1133     font-size:1.5em;
1134     border-bottom: 1px solid silver ;
1135     color: #304070 ;
1136     text-align: center ;
1137     text-shadow: 1px 1px 2px #808080;
1138 }
1139
1140
1141 div.epigraph {
1142     background: #f5f5eb ;
1143     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1144
1145     border: 1px solid silver ;
1146     border-radius: 1ex ;
1147     box-shadow: 3px 3px 3px #808080 ;
1148 }
1149
1150
1151 .example {
1152     background-color: #f5f5eb ;
1153     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
1154
1155 }
1156
1157 div.exampletitle{
1158     font-family: "URW Classico", Optima, "Linux Biolinum O",
1159         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1160         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1161     font-weight:bold;
1162     font-variant: small-caps ;
1163     border-bottom: 1px solid silver ;
1164     color: #304070 ;
1165     text-align: center ;
1166     text-shadow: 1px 1px 2px #808080;
1167 }
1168
1169
1170 .sidebar {
1171     background-color: #f5f5eb ;
1172     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
```



```
1173
1174 }
1175
1176 div.sidebartitle{
1177     font-family: "URW Classico", Optima, "Linux Biolinum O",
1178         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1179         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1180     font-weight:bold;
1181     font-variant: small-caps ;
1182     border-bottom: 1px solid silver ;
1183     color: #304070 ;
1184     text-align: center ;
1185     text-shadow: 1px 1px 2px #808080;
1186 }
1187
1188
1189 .fancyvrblabel {
1190     font-family: "URW Classico", Optima, "Linux Biolinum O",
1191         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1192         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1193     font-weight:bold;
1194     font-variant: small-caps ;
1195     font-size: 1.5em ;
1196     color: #304070 ;
1197     text-align: center ;
1198     text-shadow: 1px 1px 2px #808080;
1199 }
1200
1201
1202
1203 .minipage {
1204     background-color: #eeeeee7 ;
1205     border: 1px solid silver ;
1206     border-radius: 1ex ;
1207 }
1208
1209 figure.figure .minipage { border: none; }
1210
1211 div.marginblock div.minipage { border: none; }
1212
1213 figure , div.marginblock {
1214     background-color: #eeeeee7 ;
1215     border: 1px solid silver ;
1216     border-radius: 1ex ;
1217     box-shadow: 3px 3px 3px #808080 ;
1218 }
1219
1220 figure figure {
1221     border: 1px solid silver ;
1222     margin: 0em ;
```

```
1223 box-shadow: none ;
1224 }
1225
1226 /*
1227 figcaption {
1228     border-top: 1px solid silver ;
1229     border-bottom: 1px solid silver ;
1230     background-color: #e8e8e8 ;
1231 }
1232 */
1233
1234
1235 div.table {
1236     box-shadow: 3px 3px 3px #808080 ;
1237 }
1238
1239 /*
1240 .tnotes {
1241     background: #e8e8e8;
1242     border: 1px solid silver;
1243 }
1244 */
1245
1246
1247 nav.topnavigation{
1248     background-color: #b0b8b0 ;
1249     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
1250 }
1251
1252 nav.botnavigation{
1253     background-color: #b0b8b0 ;
1254     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
1255 }
1256
1257
1258
1259 header{
1260     background-color: #F7F7F0 ;
1261     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
1262 }
1263
1264 footer{
1265     background-color: #F7F7F0 ;
1266     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
1267 }
1268
1269
1270
1271 nav.sidetoc {
1272     background-color: #F7F7F0 ;
```

```

1273     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
1274     box-shadow: 3px 3px 3px #808080 ;
1275     border-radius: 0px 0px 0px 20px ;
1276 }
1277
1278 div.sidetocitle {color: #304070 ; }
1279 nav.sidetoc a:hover {color:#006000 ; text-decoration: none ; text-shadow:0px 0px 2px #a0a0a0;}
1280
1281
1282 @media screen and (max-width: 45em) {
1283     nav.sidetoc { border-radius: 0 ; }
1284 }
1285
1286
1287 \end{VerbatimOut}
1288 % \end{Verbatim}% for syntax highlighting

```

70.5 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.

```

1289 \begin{VerbatimOut}{lwarp_formal.css}
1290 @import url("lwarp.css") ;
1291
1292
1293
1294 A:link {color:#802020 ; text-decoration:none; }
1295 A:visited {color:#802020 ; text-shadow:none ;}
1296 A:hover {color:#400000 ; text-shadow:none ;}
1297 A:active {color:#C00000 ; text-shadow:none ;}
1298
1299
1300 body {
1301     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1302         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1303         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1304         "Times New Roman", serif;
1305     background: #fffcf5;
1306 }
1307
1308 span.textrm {
1309     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1310         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1311         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,

```

```
1312         "Times New Roman", serif;
1313     }
1314
1315     span.textsf {
1316         font-family: "DejaVu Sans", "Bitstream Vera Sans",
1317             Geneva, Verdana, sans-serif ;
1318     }
1319
1320
1321
1322     h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1323     {
1324         font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1325             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1326             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1327             "Times New Roman", serif;
1328         color: #800000 ;
1329         text-shadow: none ;
1330     }
1331
1332     h1, h2 {
1333         background-color: #fffcf5 ;
1334         background-image: none ;
1335         border-bottom: 1px solid #808080;
1336         border-top: 2px solid #808080;
1337     }
1338
1339     div.abstracttitle {
1340         font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1341             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1342             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1343             "Times New Roman", serif;
1344         color: black ;
1345         text-shadow: none ;
1346     }
1347
1348     div.abstract { font-size: 100% }
1349
1350     .sidebar {
1351         background: #fffcf5;
1352         background-image: none ;
1353         margin: 2em 5% 2em 5%;
1354         padding: 0.5em 1em;
1355         border: none ;
1356         border-top : 1px solid silver;
1357         border-bottom : 1px solid silver;
1358         font-size: 90% ;
1359     }
1360
1361     div.sidebartitle{
```

```
1362     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1363         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1364         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1365         "Times New Roman", serif;
1366     color: #800000 ;
1367     text-shadow: none ;
1368     border: none ;
1369 }
1370
1371 .example {
1372     background: #fffcf5;
1373     background-image: none ;
1374     margin: 2em 5% 2em 5%;
1375     padding: 0.5em 1em;
1376     border: none ;
1377     border-top : 1px solid silver;
1378     border-bottom : 1px solid silver;
1379 }
1380
1381 div.exampletitle{
1382     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1383         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1384         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1385         "Times New Roman", serif;
1386     color: #800000 ;
1387     text-shadow: none ;
1388     border: none ;
1389 }
1390
1391 div.fancyvrlabel{
1392     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1393         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1394         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1395         "Times New Roman", serif;
1396     color: #800000 ;
1397     text-shadow: none ;
1398     border: none ;
1399 }
1400
1401
1402
1403 .verse {
1404     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1405         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1406         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1407         "Times New Roman", serif;
1408 }
1409
1410
1411 figure {
```

```
1412     margin: 3ex 5% 3ex 5% ;
1413     padding: 1ex 1em 1ex 1em ;
1414     background-color: #fffcf5 ;
1415     overflow-x: auto ;
1416     border: none ;
1417 /*     border-top: 1px solid silver; */
1418 /*     border-bottom: 1px solid silver; */
1419 }
1420
1421
1422 figcaption , .lstlisting {
1423     border: none ;
1424 /*     border-top: 1px solid silver ; */
1425 /*     border-bottom: 1px solid silver ; */
1426     background-color: #fffcf5 ;
1427 }
1428
1429 .tnotes {
1430     background: #fffcf5 ;
1431 }
1432
1433 .theorem {
1434     background: none ;
1435 }
1436
1437 .minipage {
1438     background-color: #fffcf5 ;
1439     border: none ;
1440     margin: 1em ;
1441 }
1442
1443 div.floatrow figure { border: none ; }
1444
1445 figure figure { border: none ; }
1446
1447
1448 nav.toc, nav.lof, nav.lot, nav.lol {
1449     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1450         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1451         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1452         "Times New Roman", serif;
1453 }
1454
1455 nav.sidetoc {
1456     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1457         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1458         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1459         "Times New Roman", serif;
1460     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
1461     border-radius: 0px 0px 0px 20px ;
```

```

1462 }
1463
1464 div.sidetocitle{
1465     color: #800000 ;
1466 }
1467
1468 header{
1469     background-color: #e0e0e0 ;
1470     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
1471     text-align:center ;
1472 }
1473
1474 footer{
1475     background-color: #e0e0e0 ;
1476     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
1477     padding: 2ex 1em 2ex 1em ;
1478     clear:right ;
1479     text-align:left ;
1480 }
1481
1482 nav.botnavigation {
1483     background: #dedcd5 ;
1484     border-top: 1px solid black ;
1485 }
1486 \end{VerbatimOut}
1487 % \end{Verbatim}% for syntax highlighting

```

70.6 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.

```

1488 \begin{VerbatimOut}{sample_project.css}
1489 /* ( --- Start of project.css --- ) */
1490 /* A sample project-specific CSS file for lwarp --- ) */
1491
1492 /* Load default lwarp settings: */
1493 @import url("lwarp.css") ;
1494 /* or lwarp_formal.css, lwarp_sagebrush.css */
1495
1496 /* Project-specific CSS setting follow here. */
1497 /* . . . */
1498
1499 /* ( --- End of project.css --- ) */
1500 \end{VerbatimOut}
1501 % \end{Verbatim}% for syntax highlighting

```

70.7 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.

```

1502 \begin{VerbatimOut}{lwarp_html.xdy}
1503 (require "tex/inputenc/latin.xdy")
1504 (merge-rule "\\PS *" "Postscript")
1505 (require "texindy.xdy")
1506 (require "page-ranges.xdy")
1507 (require "book-order.xdy")
1508 (markup-locref :open "\hyperindexref{" :close "{")
1509 \end{VerbatimOut}
1510 % \end{Verbatim}% for syntax highlighting

```

70.8 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.

```

1511 \begin{VerbatimOut}{lwarp_mathjax.txt}
1512 <!-- https://groups.google.com/forum/#!topic/
1513                               mathjax-users/jUteWUcE2bY -->
1514 <script type="text/x-mathjax-config">
1515 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
1516     var seteqsectionDefault = {name: "", num: 0};
1517     var seteqsections = {}, seteqsection = seteqsectionDefault;
1518     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
1519     var AMS = MathJax.Extension["TeX/AMSmath"];
1520     TEX.Definitions.Add({
1521     macros: {
1522         seteqsection: "mySection",
1523         seteqnumber: "mySetEqNumber"
1524     }
1525     });
1526
1527     PARSE.Augment({
1528     mySection: function (name) {
1529         seteqsection.num = AMS.number;
1530         var n = this.GetArgument(name);
1531         if (n === "") {
1532             seteqsection = seteqsectionDefault;
1533         } else {

```



```

1534         if (!seteqsections["_"+n])
1535             seteqsections["_"+n] = {name:n, num:0};
1536         seteqsection = seteqsections["_"+n];
1537     }
1538     AMS.number = seteqsection.num;
1539 },
1540 mySetEqNumber: function (name) {
1541     var n = this.GetArgument(name);
1542     if (!n || !n.match(/^ *[0-9]+ *$/))
1543         n = ""; else n = parseInt(n)-1;
1544     <!-- $ syntax highlighting -->
1545     if (n === "" || n < 1)
1546         TEX.Error
1547             ("Argument to "+name+" should be a positive integer");
1548     AMS.number = n;
1549 }
1550 });
1551 MathJax.Hub.Config({
1552   TeX: {
1553     equationNumbers: {
1554       formatTag: function (n)
1555         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
1556       formatID: function (n) {
1557         n = (seteqsection.name+'.'+n).replace
1558             (/[:'"<>&]/g,"").replace(/\./,"");
1559         return 'mjax-eqn-' + n;
1560       }
1561     }
1562   }
1563 });
1564 });
1565 </script>
1566
1567 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
1568 <script type="text/x-mathjax-config">
1569   MathJax.Ajax.config.path["Contrib"] =
1570     "https://cdn.mathjax.org/mathjax/contrib";
1571 </script>
1572
1573 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
1574                                     tree/master/siunitx -->
1575 <script type="text/x-mathjax-config">
1576   MathJax.Hub.Config({
1577     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
1578     jax: ["input/TeX", "output/HTML-CSS"],
1579     tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]},
1580     TeX: {extensions: ["AMSmath.js", "AMSsymbols.js", "sinuitx.js"]}
1581   });
1582 </script>
1583

```

```

1584 <script type="text/x-mathjax-config">
1585 MathJax.Hub.Config({
1586     TeX: {
1587         equationNumbers: {
1588             autoNumber: "AMS"
1589         }
1590     }
1591 });
1592 </script>
1593
1594 <script
1595     src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
1596 </script>
1597 \end{VerbatimOut}
1598 % \end{Verbatim}% for syntax highlighting

```

70.9 Lwarpmk option

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

```

1599 \begin{LWR@createlwarpmk}

```

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

```

1600 \begin{VerbatimOut}{lwarpmk.lua}
1601 #!/usr/bin/env texlua
1602
1603 -- Copyright 2016 Brian Dunn
1604
1605 -- Print the usage of the lwarpmk command:
1606
1607 printversion = "v0.21"
1608
1609 function printhelp ()
1610 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
1611 end
1612
1613 function printusage ()
1614 print ( [[
1615
1616 lwarpmk print [project]: Compile a print version.
1617 lwarpmk printindex [project]: Process the index for the print version.
1618 lwarpmk html [project]: Compile an HTML version.
1619 lwarpmk htmlindex [project]: Process the index for the html version.
1620 lwarpmk again [project]: Touch the source code to trigger recompiles.
1621 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
1622 lwarpmk pdftohtml [project]:

```

```

1623     For use with latexmk or a Makefile:
1624     Convert project_html.pdf to project_html.html and
1625     individual HTML files.
1626 lwarpmk clean [project]: Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
1627 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
1628 lwarpmk -h: Print this help message.
1629 lwarpmk --help: Print this help message.
1630
1631 ]] )
1632 printconf ()
1633 end
1634
1635 -- Print the format of the configuration file lwarpmk.conf:
1636
1637 function printconf ()
1638 print ( [[
1639 An example lwarpmk.conf or <project>.lwarpmkconf project file:
1640 --
1641 opsystem = "Unix"      (or "Windows")
1642 latexname = "pdflatex" (or "lualatex", or "xelatex")
1643 sourcename = "projectname" (the source-code filename w/o .tex)
1644 homehtmlfilename = "index" (or perhaps the project name)
1645 htmlfilename = ""      (or "projectname" - filename prefix)
1646 uselatexmk = "false"   (or "true" to use latexmk to build PDFs)
1647 --
1648 Filenames must contain only letters, numbers, underscore, or dash.
1649 Values must be in "quotes".
1650
1651 ]] ) ;
1652 end
1653
1654
1655 -- Split one large sourcefile into a number of files,
1656 -- starting with destfile.
1657 -- The file is split at each occurrence of <!--|Start file|newfilename|*
1658
1659 function splitfile (destfile,sourcefile)
1660 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
1661 io.input(sourcefile)
1662 io.output(destfile)
1663 for line in io.lines() do
1664 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|") ;
1665 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
1666 io.output(newfilename) ;
1667 else -- not a splitpoint
1668 io.write (line .. "\n") ;
1669 end
1670 end -- do
1671 end -- function
1672

```

```
1673 -- Incorrect value, so print an error and exit.
1674
1675 function cvalueerror ( line, linenum , cvalue )
1676     print ( linenum .. " : " .. line ) ;
1677     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
1678     printconf ( ) ;
1679     os.exit(1) ;
1680 end
1681
1682 -- Load settings from the project's "lwarpmk.conf" file:
1683
1684 function loadconf ( )
1685 -- Default configuration filename:
1686 local conffile = "lwarpmk.conf"
1687 -- Optional configuration filename:
1688 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
1689 -- Verify the file exists:
1690 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
1691 print("lwarpmk: " .. conffile .. " does not exist.")
1692 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
1693 printhelp ( ) ;
1694 os.exit(1) -- exit the entire lwarpmk script
1695 else -- file exists
1696 -- Read the file:
1697 print ("lwarpmk: Reading " .. conffile .. ".")
1698 io.input(conffile) ;
1699 -- Scan each line:
1700 local linenum = 0
1701 for line in io.lines() do -- scan lines
1702 linenum = linenum + 1
1703 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w-_]*)\"") ;
1704 -- Error if incorrect enclosing characters:
1705 if ( i == nil ) then
1706 print ( linenum .. " : " .. line ) ;
1707 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
1708 printconf ( ) ;
1709 os.exit(1) ;
1710 end
1711 if ( cvarname == "opssystem" ) then
1712     -- Verify choice of opssystem:
1713     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
1714         opssystem = cvalue
1715     else
1716         cvalueerror ( line, linenum , cvalue )
1717     end
1718 elseif ( cvarname == "latexname" ) then
1719     -- Verify choice of LaTeX compiler:
1720     if (
1721         (cvalue == "pdflatex") or
1722         (cvalue == "xelatex") or
```

```

1723         (cvalue == "lualatex")
1724     ) then
1725         latexname = cvalue
1726     else
1727         cvalueerror ( line, linenum , cvalue )
1728     end
1729 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
1730 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
1731 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
1732 elseif ( cvarname == "uselatexmk" ) then uselatexmk = cvalue
1733 else
1734 print ( linenum .. " : " .. line ) ;
1735 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
1736 printconf ( ) ;
1737 os.exit(1) ;
1738 end
1739 end -- do scan lines
1740 end -- file exists
1741 -- Select some operating-system commands:
1742 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
1743 rmname = "rm"
1744 mvname = "mv"
1745 touchnamepre = "touch"
1746 touchnamepost = ""
1747 dirslash = "/"
1748 opquote= "\""
1749 elseif opsystem=="Windows" then -- For Windows
1750 rmname = "DEL"
1751 mvname = "MOVE"
1752 touchnamepre = "COPY /b"
1753 touchnamepost = "+,,"
1754 dirslash = "\\"
1755 opquote= "\""
1756 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
1757 end --- for Windows
1758 end -- loadconf
1759
1760
1761 function refreshdate ( )
1762 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
1763 end
1764
1765
1766 -- Scan the LaTeX log file for the phrase "Rerun to get",
1767 -- indicating that the file should be compiled again.
1768 -- Return true if found.
1769
1770 function reruntoget (filesourse)
1771 io.input(filesourse)
1772 for line in io.lines() do

```

```
1773 if ( string.find(line,"Rerun to get") ~= nil ) then return true end
1774 end
1775 return false
1776 end
1777
1778 -- Compile one time, return true if should compile again.
1779 -- fsuffix is "" for print, "_html" for HTML output.
1780
1781 function onetime (fsuffix)
1782 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
1783 err = os.execute(
1784 --      "echo " ..
1785      latexname .. " " .. sourcename..fsuffix )
1786 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1787 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
1788 end
1789
1790 -- Compile up to five times.
1791 -- fsuffix is "" for print, "_html" for HTML output
1792
1793 function manytimes (fsuffix)
1794 if onetime(fsuffix) == true then
1795 if onetime(fsuffix) == true then
1796 if onetime(fsuffix) == true then
1797 if onetime(fsuffix) == true then
1798 if onetime(fsuffix) == true then
1799 end end end end end
1800 end
1801
1802 -- Exit if the given file does not exist.
1803
1804 function verifyfileexists (filename)
1805 if (lfs.attributes ( filename , "modification" ) == nil ) then
1806 print ( "lwarpmk: " .. filename .. " not found." ) ;
1807 os.exit (1) ;
1808 end
1809 end
1810
1811 -- Convert <project>_html.pdf into HTML files:
1812
1813 function pdftohtml ()
1814     -- Convert to text:
1815     print ("lwarpmk: Converting " .. sourcename .. "_html.pdf to " .. sourcename .. "_html.html"
1816     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout " .. sourcename .. "_html.pdf " .. sou
1817     -- Split the result into individual HTML files:
1818     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
1819 end
1820
1821 -- Remove auxiliary files:
1822
```

```

1823 function removeaux ()
1824     os.execute ( rmname .. " " ..
1825         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
1826         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
1827         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
1828         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
1829         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
1830         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
1831         sourcename .. ".log " .. sourcename .. "_html.log "
1832     )
1833 end
1834
1835
1836
1837 -- Create lateximages based on lateximages.txt:
1838 function createlateximages ()
1839 print ("lwarpmk: Creating lateximages.")
1840 io.input("lateximages.txt")
1841 -- Create the lateximages directory, ignore error if already exists
1842 err = os.execute("mkdir lateximages")
1843 -- Scan lateximages.txt
1844 for line in io.lines() do
1845 -- lwimgpage is the page number in the PDF which has the image
1846 -- lwimgnum is the sequential lateximage number to assign for the image
1847 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
1848 -- For each entry:
1849 if ( i~=nil ) then
1850 -- Separate out the image into its own single-page pdf:
1851 err = os.execute(
1852 "pdfseparate -f " .. lwimgpage .. " -l " ..
1853 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
1854 -- Crop the image:
1855 err = os.execute(
1856 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
1857 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1858 -- Convert the image to svg:
1859 err = os.execute(
1860 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
1861 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1862 -- Move the result into lateximages/:
1863 err = os.execute(
1864 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
1865 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1866 -- Remove the temporary files:
1867 err = os.execute(
1868 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
1869 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
1870 end
1871 end -- do
1872 end -- function

```

```
1873
1874
1875
1876
1877 -- lwarpmk --version :
1878
1879 if (arg[1] == "--version") then
1880 print ( "lwarpmk: " .. printversion )
1881
1882 else -- not -- version
1883
1884 -- print intro:
1885
1886 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
1887
1888 -- lwarpmk print:
1889
1890 if arg[1] == "print" then
1891 loadconf ()
1892 if ( uselatexmk == "true" ) then
1893     os.execute ( "latexmk -pdf -dvi- -ps- -pdflatex=\"\" .. latexname .." %0 %S\" " .. sourcenam
1894     print ("lwarpmk: Done.")
1895 else -- not latexmk
1896     verifyfileexists (sourcename .. ".tex") ;
1897     -- See if up to date:
1898     if (
1899         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
1900         (
1901             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
1902             lfs.attributes ( sourcename .. ".pdf" , "modification" )
1903         )
1904     ) then
1905         -- Recompile if not yet up to date:
1906         manytimes("")
1907         print ("lwarpmk: Done.") ;
1908     else
1909         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
1910     end
1911 end -- not latexmk
1912
1913 -- lwarp printindex:
1914 -- Compile the index then touch the source
1915 -- to trigger a recompile of the document:
1916
1917 elseif arg[1] == "printindex" then
1918 loadconf ()
1919 print ("lwarpmk: Processing the index.")
1920 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. ".idx")
1921 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
1922 refreshdate ()
```



```
1923 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1924 print ("lwarpmk: Done.")
1925
1926 -- lwarpmk html:
1927
1928 elseif arg[1] == "html" then
1929 loadconf ()
1930 if ( uselatexmk == "true" ) then
1931     -- The recorder option is required to detect changes in <project>.tex
1932     -- while we are loading <project>_html.tex.
1933     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
1934         .. "-e "
1935         .. opquote .. "$makeindex = q/texindy -M lwarp_html.xdy/" .. opquote
1936         .. " -pdflatex=\"\" .. latexname .. " %0 %S\" "
1937         .. sourcename .. "_html.tex" ) ;
1938     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
1939     pdftohtml ()
1940     print ("lwarpmk: Done.")
1941 else -- not latexmk
1942     verifyfileexists ( sourcename .. ".tex" ) ;
1943     -- See if exists and is up to date:
1944     if (
1945         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
1946         (
1947             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
1948             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
1949         )
1950     ) then
1951         -- Recompile if not yet up to date:
1952         manytimes("_html")
1953         pdftohtml ()
1954         print ("lwarpmk: Done.")
1955     else
1956         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
1957     end
1958 end -- not latexmk
1959
1960 elseif arg[1] == "pdftohtml" then
1961     loadconf ()
1962     pdftohtml ()
1963
1964 -- lwarpmk htmlindex:
1965 -- Compile the index then touch the source
1966 -- to trigger a recompile of the document:
1967
1968 elseif arg[1] == "htmlindex" then
1969 loadconf ()
1970 print ("lwarpmk: Processing the index.")
1971 os.execute("texindy -M lwarp_html.xdy " .. sourcename .. "_html.idx")
1972 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex")
```

```
1973 refreshdate ()
1974 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1975 print ("lwarpmk: Done.")
1976
1977 -- lwarpmk limages:
1978 -- Scan the lateximages.txt file to create lateximages,
1979 -- then touch the source to trigger a recompile.
1980
1981 elseif arg[1] == "limages" then
1982 loadconf ()
1983 print ("lwarpmk: Processing images.")
1984 createlateximages ()
1985 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
1986 refreshdate ()
1987 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1988 print ("lwarpmk: Done.")
1989
1990 -- lwarpmk again:
1991 -- Touch the source to trigger a recompile.
1992
1993 elseif arg[1] == "again" then
1994 loadconf ()
1995 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
1996 refreshdate ()
1997 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
1998 print ("lwarpmk: Done.")
1999
2000 -- lwarpmk clean:
2001 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2002
2003 elseif arg[1] == "clean" then
2004 loadconf ()
2005 removeaux ()
2006 print ("lwarpmk: Done.")
2007
2008 -- lwarpmk cleanall
2009 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log
2010 --      and also project.pdf, *.html
2011
2012 elseif arg[1] == "cleanall" then
2013 loadconf ()
2014 removeaux ()
2015 os.execute ( rmname .. " " ..
2016      sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
2017      "*.html"
2018      )
2019 print ("lwarpmk: Done.")
2020
2021 -- lwarpmk with no argument :
2022
```

```

2023 elseif (arg[1] == nil) then
2024 printhelp ()
2025
2026 -- lwarpmk -h or lwarpmk --help :
2027
2028 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
2029 printusage ()
2030
2031 else
2032 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
2033 printhelp ()
2034 end
2035
2036 end -- not --version
2037 \end{VerbatimOut}
2038 % \end{Verbatim}% for syntax highlighting
2039 \end{LWR@createlwarpmk}

```

Package 3

lwarp-afterpage.sty

71 Afterpage

Pkg afterpage

for **HTML output**: Discard all options for lwarp-afterpage:

```

1 \LWR@ProvidesPackageDrop{afterpage}

2 \newcommand{\afterpage}[1]{#1}

```

Package 4

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 115.1.

for HTML output: 2 \begin{warpHTML}

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*{\ALG@doentity}{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }

20 \end{warpHTML}
```

Package 5

lwarp-alltt.sty

73 Alltt

Pkg alltt

for **HTML output**: alltt is patched for use by lwarp.

```
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 6

lwarp-bookmark.sty

74 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 7

lwarp-booktabs.sty

75 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 8

lwarp-draftwatermark.sty

76 Draftwatermark

Pkg draftwatermark draftwatermark is emulated during HTML output, and the draftwatermark package is ignored.

1 \LWR@ProvidesPackageDrop{draftwatermark}

for HTML output: 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 9

lwarp-epigraph.sty

77 Epigraph

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

```

1 \LWR@ProvidesPackageDrop{epigraph}

for HTML output:
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 10

lwarp-eso-pic.sty

78 Eso-pic

Pkg eso-pic eso-pic is emulated during HTML output, and the eso-pic package is ignored.

```

1 \LWR@ProvidesPackageDrop{eso-pic}

for HTML output: 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 11

lwarp-everypage.sty

79 Everypage

Pkg `everypage` `everypage` is emulated during HTML output, and the `everypage` package is ignored.

```

1 \LWR@ProvidesPackageDrop{everypage}

for HTML output: 2 \newcommand*{\AddEverypageHook}[1]{}
                  3 \newcommand*{\AddThispageHook}[1]{}

```

Package 12

lwarp-extramarks.sty

80 Extramarks

Pkg `extramarks`

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 13

lwarp-fancyhdr.sty

81 Fancyhdr

Pkg fancyhdr

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 14

lwarp-float.sty

82 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]

Pkg float

The float package is emulated.

See section 52.2 for the \listof command.

\newfloat {<1: type>} {<2: placement>} {<3: ext>} [<4: within>]

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
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 {<type>} {<name>}

Sets the text name of the float, such as “Figure”.

```
11 \NewDocumentCommand{\floatname}{m +m}{%
12 \SetupFloatingEnvironment{#1}{name=#2}%
13 }
```

\floatplacement {<type>} {<placement>}

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 15

lwarp-floatrow.sty

83 Floatrow

Pkg floatrow floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{floatrow}

Pkg floatrow

The floatrow package is emulated.

⚠ \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.

Only parameters for caption and object are used.

LWR@insubfloatrow is true if inside a subfloatrow environment.

```
2 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
3 \ifbool{LWR@insubfloatrow}
4 {
```

Inside a subfloatrow. subfigure and subtable take a width argument. Defaults to \linewidth if none is given.

```
5 \IfValueTF{#3}
6 {\@nameuse{sub#2}{#3}}
7 {\@nameuse{sub#2}{\linewidth}}
8 }
9 {
```

Not inside a subfloatrow. `figure` and `table` do not take a width argument.

```

10 \@nameuse{#2}
11 }
12 #6
13
14 #7
15 \ifbool{LWR@insubfloatrow}
16 {\@nameuse{endsub#2}}
17 {\@nameuse{end#2}}
18 }

```

Not used:

```

19 \newcommand*{\nocapbeside}{}
20 \newcommand*{\capbeside}{}
21 \newcommand*{\captop}{}
22 \newlength{\FBwidth}
23 \setlength{\FBwidth}{.3\linewidth}
24 \newlength{\FBheight}
25 \setlength{\FBheight}{2in}
26 \newcommand*{\useFCwidth}{}
27 \newcommand{\floatsetup}[2][{}]{
28 \newcommand{\thisfloatsetup}[1]{
29 \newcommand{\clearfloatsetup}[1]{
30 \newcommand*{\killfloatstyle}{}

```

Preamble and default width are ignored.

```

31 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
32 \@namedef{#1}{
33 \floatbox{#2}
34 }
35 }

```

Preamble and default width are ignored.

```

36 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
37 \@namedef{#1}{%
38 \floatbox{#2}
39 }
40 }

```

```

41 \newfloatcommand{ffigbox}{figure}[\nocapbeside][

```

```

42 \newfloatcommand{ttabbox}{table}[\captop][\FBwidth]

```

```

43 \newfloatcommand{fcapside}{figure}[\capbeside][

```

The row of floats is placed into a div of class floatrow.

```
44 \newenvironment*{floatrow}[1][2]
45 {
46 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
47 \booltrue{LWR@infloatrow}
48 \setlength{\linewidth}{6in/#1}
49 }
50 {
51 \boolfalse{LWR@infloatrow}
52 \endBlockClass
53 }
```

Keys for \DeclareNewFloatType:

```
54 \newcommand*{\LWR@frowkeyplacement}{}
55 \newcommand*{\LWR@frowkeyname}{}
56 \newcommand*{\LWR@frowkeyfileext}{}
57 \newcommand*{\LWR@frowkeywithin}{}
58 \newcommand*{\LWR@frowkeycapstyle}{}
59
60 \define@key{frowkeys}{placement}{}%
61 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
62 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
63 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
64 \define@key{frowkeys}{relatedcapstyle}{}%
```

Use \listof{type}{Title} to print a list of the floats.

```
65 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
66 \renewcommand*{\LWR@frowkeyplacement}{}
67 \renewcommand*{\LWR@frowkeyname}{}
68 \renewcommand*{\LWR@frowkeyfileext}{}
69 \renewcommand*{\LWR@frowkeywithin}{}
70 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
71 \LWR@traceinfo{about to setkeys frowkeys}%
72 \setkeys{frowkeys}{#2}%
73 \LWR@traceinfo{finished setkeys frowkeys}%
```


Create a new float with optional [within]:

```

74 \ifthenelse{\equal{\LWR@frowkeywithin}{}}
75 {
76 \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ \LWR@frowkeyfileext}%
77 \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
78 }%
79 {%
80 \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ \LWR@frowkeyfileext\ \LWR@frowkeywi
81 \newfloat{#1}{\LWR@frowkeyplacement}%
82 {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
83 \LWR@traceinfo{*** finished newfloat #1}
84 }%

```

Rename the float if a name was given:

```

85 \ifthenelse{\equal{\LWR@frowkeyname}{}}
86 {}
87 {\floatname{#1}{\LWR@frowkeyname}}%
88 }

```

Not used:

```

89 \newcommand{\buildFBBBOX}[2]{}
90 \newcommand*{\CenterFloatBoxes}{}
91 \newcommand*{\TopFloatBoxes}{}
92 \newcommand*{\BottomFloatBoxes}{}
93 \newcommand*{\PlainFloatBoxes}{}
94
95 \newcommand{\capsubrowsettings}{}
96
97 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```

98 \newcommand{\RawCaption}[1]{#1}

```

Places additional text inside a float, inside a CSS div of class floatfoot.

```

99 \NewDocumentCommand{\floatfoot}{s +m}{%
100 \begin{BlockClass}{floatfoot}
101 #2
102 \end{BlockClass}
103 }

```

Used to compute \linewidth.

```

104 \newbool{\LWR@insubfloatrow}
105 \boolfalse{\LWR@insubfloatrow}

```

```
106 \newenvironment*{subfloatrow}[1][2]
107 {
```

The row of floats is placed into a `div` of class `floatrow`:

```
108 \BlockClass{floatrow}
```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
109 \begingroup
110 \booltrue{LWR@insubfloatrow}
111 }
112 {
113 \endgroup
114 \endBlockClass
115 \boolfalse{LWR@insubfloatrow}
116 }
```

Package 16

lwarp-fontenc.sty

84 Fontenc

Pkg **fontenc** Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

```
1 \LWR@ProvidesPackageDrop{fontenc}
```

```
2 \LWR@noloadafter{fontenc}
```

Package 17

lwarp-fontspec.sty

85 Fontspec

Pkg **fontspec** Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

```
1 \LWR@ProvidesPackageDrop{fontspec}
```

```
2 \LWR@noloadafter{fontspec}
```

Package 18

lwarp-geometry.sty

86 Geometry

Pkg geometry

for **HTML output:** Discard all options for lwarp-geometry:

```
1 \LWR@ProvidesPackageDrop{geometry}
```

Package 19

lwarp-graphics.sty

87 Graphics

Pkg graphics graphics is not used. The user is required to use graphicx instead.

for HTML output:

```
1 \PackageError{lwarp-graphics}
2 {The graphics package is not supported by
3 the lwarp package's HTML conversion.
4 Please use graphicx instead. \protect\usepackage{graphicx}}
5 {Graphicx requires a different syntax but is more flexible,
6 and it is emulated by lwarp.}
```

Package 20

lwarp-graphicx.sty

88 Graphicx

Pkg graphicx graphicx is emulated during HTML output, and the graphicx package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{graphicx}
```

Package 21

lwarp-hyperref.sty

89 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 22

lwarp-inputenc.sty

90 Inputenc

Pkg `inputenc` Error if inputenc is loaded after lwarp.

Discard all options for lwarp-inputenc:

```
1 \LWR@ProvidesPackageDrop{inputenc}
```

```
2 \LWR@noloadafter{inputenc}
```

Package 23

lwarp-keyfloat.sty

91 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 24

lwarp-letterspace.sty

92 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 25

lwarp-listings.sty

93 Listings

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}{toclstlisting}{#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 reenale line numbers:

```

26 \let\lsthk@EveryPar\relax%
27
28 \LWR@afterendverbatim%
29 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
30 \else%

```

Inline, so create the closing ``:

```

31 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
32 \fi%

```

Final listings deinit:

```

33 \LWR@origlst@DeInit%
34 }

```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```

35 \def\lst@MakeCaption#1{%
36 \LWR@traceinfo{MAKING CAPTION at #1}%
37 \lst@ifdisplaystyle
38 \LWR@traceinfo{making a listings display caption}%
39 \ifx #1t%
40 \ifx\lst@caption\@empty\expandafter\lst@HRefStepCounter \else
41 \expandafter\refstepcounter
42 \fi {lstlisting}%
43 \LWR@traceinfo{About to assign label: !\lst@label!}%
44 % \ifx\lst@label\@empty\else
45 % \label{\lst@label}\fi
46 \LWR@traceinfo{Finished assigning the label.}%
47 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
48 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
49 \lst@ifnolol\else
50 \ifx\lst@caption\@empty
51 \ifx\lst@caption\@empty
52 \ifx\lst@intname\@empty \else \def\lst@temp{ }%

```

```
53 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
54 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
55 % \addcontentsline{lol}{lstlisting}{\lst@name}
56 \fi\fi
57 \fi
58 \else
```

This would have to be modified for lwarp:

```
59 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
60 \addcontentsline{lol}{lstlisting}%
61 {\protect\numberline{\thelstlisting}{\protect\ignorespaces \lst@@caption \
62 \fi
63 \fi
64 \fi
65 \ifx\lst@caption\@empty\else
66 \LWR@traceinfo{lst@caption not empty-}%
67 \lst@ifsubstring #1\lst@captionpos
68 {\begingroup
69 \LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
70 % \let\@vskip\vskip
71 % \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
72 % \def\lst@vskip{\nobreak\@vskip\@tempskipa\nobreak}%
73 % \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
74 % \ifx #1t\allowbreak \fi
75 \ifx\lst@title\@empty
```

New lwarp code to create a caption:

```
76 \lst@makecaption\fnnum\lstlisting{\ignorespaces \lst@caption}
77 \else
```

New lwarp code to create a title:

```
78 % \lst@maketitle\lst@title % (AS)
79 \LWR@traceinfo{Making title: \lst@title}%
80 \begin{BlockClass}{lstlistingtitle}% lwarp
81 \lst@maketitle\lst@title% lwarp
82 \end{BlockClass}% lwarp
83 \fi
84 \LWR@traceinfo{About to assign label: !\lst@label!}%
85 \ifx\lst@label\@empty\else
```

```

86 \leavevmode% gets rid of bad space factor error
87 \GetTitleStringExpand{\lst@caption}%
88 \edef\LWR@lntemp{\GetTitleStringResult}%
89 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
90 \label{\lst@label}\fi
91 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

92 %           \ifx #1b\allowbreak \fi
93           \endgroup}{}%
94   \fi
95 \LWR@traceinfo{end of making a listings display caption}%
96   \else
97 \LWR@traceinfo{INLINE}%
98   \fi
99 \LWR@traceinfo{DONE WITH CAPTION at #1}%
100 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

101 \lst@Key{numbers}{none}{%
102   \let\lst@PlaceNumber\@empty
103   \lstKV@SwitchCases{#1}%
104   {none&\\%
105     left&\def\lst@PlaceNumber{%
106 % \llap{
107 \LWR@orignormalfont%
108 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
109 % }
110 }
111 \\%
112   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
113     \kern\VerbatimHTMLWidth \kern\lst@numbersep
114     \lst@numberstyle{\thelstnumber}}}%
115   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
116
117 \end{warpHTML}

```

Package 26

lwarp-longtable.sty

94 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` * `[<horizontalalignment>] {<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 27

lwarp-ltcaption.sty

95 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 28


lwarp-mdframed.sty

96 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:

```
4 \mdfsetup{
5   startcode={\LWR@mdframedstart},
6   endcode={\LWR@mdframedend},
7   startinnercode={\LWR@startpars},
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 29

lwarp-microtype.sty

97 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 30

lwarp-multicol.sty

98 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.

for HTML output: 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 31

lwarp-multirow.sty

99 Multirow

Pkg `multirow` `multirow` is emulated during HTML output, and the `multirow` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{multirow}`

Package 32

lwarp-needspace.sty

100 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 33

lwarp-newunicodechar.sty

101 Newunicodechar

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

```
1 \LWR@ProvidesPackageDrop{newunicodechar}
2 \LWR@noloadafter{newunicodechar}
```

Package 34

lwarp-nowidow.sty

102 Nowidow

Pkg nowidow nowidow is not used during HTML conversion.

for HTML output: Discard all options for lwarp-nowidow:

```
1 \LWR@ProvidesPackageDrop{nowidow}
2 \newcommand*{\nowidow}[1] [] {}
3 \newcommand*{\setnowidow}[1] [] {}
4 \newcommand*{\noclub}[1] [] {}
5 \newcommand*{\setnoclub}[1] [] {}
```

Package 35

lwarp-pagenote.sty

103 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.

for HTML output: Discard all options for `lwarp-pagenote`:

```
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 36

lwarp-placeins.sty

104 Placeins

Pkg placeins placeins is not used during HTML conversion.

for HTML output: Discard all options for lwarp-placeins:

```
1 \LWR@ProvidesPackageDrop{placeins}
```

```
2 \newcommand*{\FloatBarrier}{} 
```


Package 37

lwarp-ragged2e.sty

105 Ragged2e

Pkg ragged2e ragged2e is not used during HTML conversion.

for HTML output: Discard all options for lwarp-ragged2e:

```

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 38

lwarp-rotating.sty

106 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 39

lwarp-setspace.sty

107 Setspace

Pkg setspace setspace is not used during HTML conversion.

for HTML output: Discard all options for lwarp-setspace:

```

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}

```

Package 40

lwarp-textpos.sty

108 Textpos

Pkg `textpos` `textpos` is emulated during HTML output, and the `textpos` package is ignored.

```

1 \LWR@ProvidesPackageDrop{textpos}

for HTML output:
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 41

lwarp-threeparttable.sty

109 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 42

lwarp-titles.sty

110 Titles

Pkg titles titles is loaded and used by lwarp during HTML output. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-titles:

```

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 43

lwarp-titlesec.sty

111 Titlesec

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.
 for HTML output: Discard all options for lwarp-titlesec:

```

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 44

lwarp-titletoc.sty

112 Titletoc

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.
 for HTML output: Discard all options for lwarp-titletoc:

```

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 45

lwarp-titling.sty

113 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.

for HTML output: Pass all options to lwarp-titling:

```
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 46

lwarp-tocloft.sty

114 Tocloft

Pkg `tocloft` `tocloft` is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

for HTML output: Discard all options for `lwarp-tocloft`:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*{\cftmarktoc}{}
4 \newcommand*{\cfttoctitlefont}{}
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*{\cftmarklof}{}
9 \newcommand*{\cftloftitlefont}{}
10 \newcommand*{\cftafterloftitle}{}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

13 \newcommand*{\cftmarklot}{}
14 \newcommand*{\cftlottitlefont}{}
15 \newcommand*{\cftafterlottitle}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}{1}
20 \newcommand*{\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*{\cftsetpnumwidth}[1]{}
24 \newcommand*{\cftsetrmarg}[1]{}

```

```
25 \newcommand*{\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpartincent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpfont}{}
31 \newcommand*{\cftpapresnum}{}
32 \newcommand*{\cftpataftersnum}{}
33 \newcommand*{\cftpataftersnumb}{}
34 \newcommand*{\cftpleader}{}
35 \newcommand*{\cftpdotsep}{1}
36 \newcommand*{\cftpfont}{}
37 \newcommand*{\cftpartafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}{}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchapfont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}{}
53 \newcommand*{\cftsecpresnum}{}
54 \newcommand*{\cftsecaftersnum}{}
55 \newcommand*{\cftsecaftersnumb}{}
56 \newcommand*{\cftsecleader}{}
57 \newcommand*{\cftsecdotsep}{1}
58 \newcommand*{\cftsecfont}{}
59 \newcommand*{\cftsecafterpnum}{}

60 \newlength{\cftbeforesubsecskip}
61 \newlength{\cftsubsecindent}
62 \newlength{\cftsubsecnumwidth}
63 \newcommand*{\cftsubsecfont}{}
64 \newcommand*{\cftsubsecpresnum}{}
65 \newcommand*{\cftsubsecaftersnum}{}
66 \newcommand*{\cftsubsecaftersnumb}{}
67 \newcommand*{\cftsubsecleader}{}
68 \newcommand*{\cftsubsecdotsep}{1}
```

```
69 \newcommand*{\cftsubsecpagefont}{}
70 \newcommand*{\cftsubsecafterpnum}{}

71 \newlength{\cftbeforesubsubsecskip}
72 \newlength{\cftsubsubsecindent}
73 \newlength{\cftsubsubsecnumwidth}
74 \newcommand*{\cftsubsubsecfont}{}
75 \newcommand*{\cftsubsubsecpresnum}{}
76 \newcommand*{\cftsubsubsecaftersnum}{}
77 \newcommand*{\cftsubsubsecaftersnumb}{}
78 \newcommand*{\cftsubsubsecleader}{}
79 \newcommand*{\cftsubsubsecdotsep}{1}
80 \newcommand*{\cftsubsubsecpagefont}{}
81 \newcommand*{\cftsubsubsecafterpnum}{}

82 \newlength{\cftbeforeparaskip}
83 \newlength{\cftparaindent}
84 \newlength{\cftparanumwidth}
85 \newcommand*{\cftparafont}{}
86 \newcommand*{\cftparapresnum}{}
87 \newcommand*{\cftparaftersnum}{}
88 \newcommand*{\cftparaftersnumb}{}
89 \newcommand*{\cftparaleader}{}
90 \newcommand*{\cftparadotsep}{1}
91 \newcommand*{\cftparapagefont}{}
92 \newcommand*{\cftparaafterpnum}{}

93 \newlength{\cftbeforesubparaskip}
94 \newlength{\cftsubparaindent}
95 \newlength{\cftsubparanumwidth}
96 \newcommand*{\cftsubparafont}{}
97 \newcommand*{\cftsubparapresnum}{}
98 \newcommand*{\cftsubparaftersnum}{}
99 \newcommand*{\cftsubparaftersnumb}{}
100 \newcommand*{\cftsubparaleader}{}
101 \newcommand*{\cftsubparadotsep}{1}
102 \newcommand*{\cftsubparapagefont}{}
103 \newcommand*{\cftsubparaafterpnum}{}

104 \newlength{\cftbeforefigskip}
105 \newlength{\cftfigindent}
106 \newlength{\cftfignumwidth}
107 \newcommand*{\cftfigfont}{}
108 \newcommand*{\cftfigpresnum}{}
109 \newcommand*{\cftfigaftersnum}{}
110 \newcommand*{\cftfigaftersnumb}{}
111 \newcommand*{\cftfigleader}{}
112 \newcommand*{\cftfigdotsep}{1}
113 \newcommand*{\cftfigpagefont}{}

```

```

114 \newcommand*{\cftfigafterpnum}{}

115 \newlength{\cftbeforesubfigskip}
116 \newlength{\cftsubfigindent}
117 \newlength{\cftsubfignumwidth}
118 \newcommand*{\cftsubfigfont}{}
119 \newcommand*{\cftsubfigpresnum}{}
120 \newcommand*{\cftsubfigaftersnum}{}
121 \newcommand*{\cftsubfigaftersnumb}{}
122 \newcommand*{\cftsubfigleader}{}
123 \newcommand*{\cftsubfigdotsep}{1}
124 \newcommand*{\cftsubfigpagefont}{}
125 \newcommand*{\cftsubfigafterpnum}{}

126 \newlength{\cftbeforetabskip}
127 \newlength{\cfttabindent}
128 \newlength{\cfttabnumwidth}
129 \newcommand*{\cfttabfont}{}
130 \newcommand*{\cfttabpresnum}{}
131 \newcommand*{\cfttabaftersnum}{}
132 \newcommand*{\cfttabaftersnumb}{}
133 \newcommand*{\cfttableader}{}
134 \newcommand*{\cfttabdotsep}{1}
135 \newcommand*{\cfttabpagefont}{}
136 \newcommand*{\cfttabafterpnum}{}

137 \newlength{\cftbeforesubtabskip}
138 \newlength{\cftsubtabindent}
139 \newlength{\cftsubtabnumwidth}
140 \newcommand*{\cftsubtabfont}{}
141 \newcommand*{\cftsubtabpresnum}{}
142 \newcommand*{\cftsubtabaftersnum}{}
143 \newcommand*{\cftsubtabaftersnumb}{}
144 \newcommand*{\cftsubtableader}{}
145 \newcommand*{\cftsubtabdotsep}{1}
146 \newcommand*{\cftsubtabpagefont}{}
147 \newcommand*{\cftsubtabafterpnum}{}

148 \newcommand{\cftsetindents}[3]{}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberon}[1]{}

```

Emulated through the \newfloat mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}

```

```

154 {\newfloat{#2}{tbp}{#3}[#1]}
155 {\newfloat{#2}{tbp}{#3}}
156 \@namedef{listof#2}{\listof{#2}{#4}}
157 \@namedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \@namedef{cft#2font}{}
162 \@namedef{cft#2presnum}{}
163 \@namedef{cft#2aftersnum}{}
164 \@namedef{cft#2aftersnumb}{}
165 \@namedef{cft#2leader}{}
166 \@namedef{cft#2dotsep}{1}
167 \@namedef{cft#2pagefont}{}
168 \@namedef{cft#2afterpnum}{}
169 }

```

\cftchapterprecis from tocloft:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{
176   \addtocontents{toc}{%
177     {
178       \protect\begin{quote}#1\protect\end{quote}}
179   }
180 }

```

Package 47

lwarp-trivfloat.sty

115 Trivfloat

Pkg `trivfloat` `trivfloat` is forced to use the built-in `lwarp` emulation for floats.

for HTML output: Discard all options for `lwarp-trivfloat`:

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

for HTML & PRINT: `3 \begin{warpall}`



Package `trivfloat` is supported via the `lwarp` emulation of float. For HTML usage, use only `\usepackage{trivfloat}`, without any options, which tells `trivfloat` to use the emulated float functions instead of using `floatrow` or `memoir`.

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
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}
```

115.1 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 48

lwarp-wallpaper.sty

116 Wallpaper

Pkg wallpaper wallpaper is emulated during HTML output, and the wallpaper package is ignored.

```
1 \LWR@ProvidesPackageDrop{wallpaper}
```

for HTML output:

```
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 49

lwarp-wrapfig.sty

117 Wrapfig

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

```
1 \LWR@ProvidesPackageDrop{wrapfig}
```

for HTML output: 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 50

lwarp-xcolor.sty

118 Xcolor

Pkg `xcolor`

`support` Color definitions, models, and mixing are fully supported without any changes required.

`tables` Colored tables are ignored. 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.

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 }
```

Ignored. Use `\NewCSS` instead.

```
20 \newcommand*{\LWR@pagecolor}[2][named]{}
```

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 }
```

```
[<framemodel>] {<framecolor>} [<boxmodel>] {<boxcolor>} {<text>}
```

is converted into a framed HTML hex background color span.

```
26 \NewDocumentCommand{\LWR@fcolorbox}{0{named} m 0{named} m m}{%
27 \begingroup%
28 \LWR@colorstyle{border:1px solid }{#1}{#2}%
29 {\LWR@colorstyle{background:}{#3}{#4}{#5}}%
```

```
30 \endgroup%  
31 }
```


Package 51

lwarp-xfrac.sty

119 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		Docs: Commands for a successful HTML conversion.	53
General: 2016/03/08 Initial version	1		
v0.11		Docs: Commands into a warpprint environment.	52
General: 2016/03/11	1	Docs: Newclude limitations. . .	63
Added section:		Docs: Table: Cross-referencing data structures.	184
Operating-System portability.	84	Docs: Table: Float data structures.	193
Added section: Selecting the operating system.	51	Docs: Trademarks section. . .	262
Test Suite: limages and index in README.txt	1	Docs: Troubleshooting cross-references.	65
Test Suite: MS-Windows in README.txt	1	floatrow: Added package	318
v0.12		Pagenote: Assigned cleveref name.	238
\LWR@newhtmlfile: Bugfix: TOC with numbered files.	116	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		
Test Suite: Sub-figures	1	v0.15	
v0.13		\printpublished: No div if empty.	137
\CaptionSeparator: Fix for newer babel package.	195	\printsubtitle: No div if empty.	137
\LWR@LwarpStart: \up and \fup	128	General: 2016/04/06	1
\published: Default to empty published.	134	Added	322
\subtitle: Default to empty subtitle.	134	Ampersand (&): Fixed handling when passed as an argument.	162
General: 2016/03/24	1	Docs: Added warning icons for items needing special attention.	69
Removed package: subfig	1	Docs: Clarify print/HTML output.	51
Test Suite: Ordinals, Subcaption	1	Docs: Moved the supported functions table to the introduction.	19
tikzpicture: Fix dollar-redefined bug for newer package.	240	Files: lwarp_formal.css added. . .	1
v0.14		Fix: steps counter	321
\LWR@htmlsectionfilename: Fix: Links to home page.	91	Fixed & handling.	319
\noteentry: Fix for label inside a footnote.	111	Test Suite: test_suite_formal.css file added.	1
General: 2016/03/31	1		

v0.16	
General: 2016/04/11	1
\lwr@titlingpage: Improved	
print-output spacing.	138
Added XeLaTeX, LuaLaTeX	
support.	75
Docs: Font and UTF-8 support.	50
Docs: Moved location of	
\usepackage{lwarp}.	51
Docs: Text not converting.	65
Fix: amsmath options clash	79
Fix: newtxmath compatibility.	79
Lwarp no longer selects fonts.	50, 75
Removed package: suffix	1
Test Suite: Improved	
titlingpage.	138
Test Suite: Lwarp no longer	
selects fonts.	1
Test Suite: Supports XeLaTeX,	
LuaLaTeX.	1
xfrac: Adjusted for the use of	
any font:	376
v0.17	
\lwr@htmlsectionfilename: Fix:	
Links when entire doc is one	
HTML page.	91
General: 2016/04/14	1
mdframed: Added.	339
Test Suite: Fix: Print-version	
front-matter page numbers.	1
Test Suite: Mdfamed	1
v0.18	
\lwr@hspace: \hspace supported	253
\lwr@includegraphicsb: Add:	
svgz file extension.	232
em, ex, %, px dimensions	
preserved.	232
Fix: \linewidth, \textwidth,	
\textheight inside a	
minipage.	232
Improved HTML output	
linebreaks.	232
\lwr@myshorttoc: Reorganize	
\HomeHTMLFilename logic.	199
\lwr@newhtmlfile: Sidetoc after	
title, improving responsive	
design.	115
\lwr@requesttoc: Reorganize	
\HomeHTMLFilename logic.	129
\lwr@subhyperref: Improved	
HTML output linebreaks.	191
\lwr@subhyperrefclass: Improved	
HTML output linebreaks.	191
\lwr@subinlineimage: Surpress	
extra space.	192
General: 2016/05/19	1
File: lwarp.css: Improved TOC	
outline display.	1
Files: lwarp.css and	
lwarp_formal.css: Improved	
responsive design.	1
Microtype disabled during	
HTML generation	76
PDF Unicode input characters.	70
Test Suite: Verse package	1
\lwr@image: pdfcrop: --hires	
added.	223
Reorganize \HomeHTMLFilename	
logic.	223
Surpress extra space.	223
verse: Supports verse, memoir	
packages.	147
minipage: Fix: \linewidth,	
\textwidth, \textheight	
inside a minipage.	244
v0.19	
\HTMLFileName: Docs: Escape	
filename underscores.	91
\HomeHTMLFileName: Docs: Escape	
filename underscores.	91
\lwr@LwarpStart: Enabled \\	
equal to \newline.	127
\lwr@doubledollar: MathJax	
support.	209
\lwr@filestart:	
lwarp_mathjax.txt loaded.	124
\lwr@hspace: Fix: \hspace length	
computations	253
\lwr@minipagestartpars:	
Surpresses paragraph tags	
between minipages.	252
\lwr@singledollar: MathJax	
support.	209
\lwr@tabledatamultirowtag:	
Added optional args.	179
\lwr@lateximageFontSizeName: Add:	
User-adjustable	
math/lateximage font size.	222

<code>\minipagefullwidth</code> : Added: No width tag for the next minipage in HTML.	244	<code>\LWR@hspace</code> : Add: Supports HTML thin breakable space	253
<code>\rule</code> : Added	257	<code>\LWR@htmldivclass</code> : Added optional style.	98
<code>\warpHTMLonly</code> : Added.	71	<code>\LWR@htmlelementclass</code> : Added optional style.	98
<code>\warpprintonly</code> : Replaces <code>\rowprintedonly</code>	71	<code>\LWR@htmlsectionfilename</code> : HTMLFileName: removed additional trailing '-', and may be empty.	91
General: 2016/06/08	1	Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	91
Added.	375	<code>\LWR@includegraphicsb</code> : Fix: <code>\linewidth</code> in a floatrow.	232
Avoids MathJax.	208	Fix: Expands filename.	232
cleveref: Loaded <code>\AtEndPreamble</code>	238	<code>\LWR@longtabledatacaptiontag</code> : Fix: Pars in captions.	176
CSS for table note item.	357	<code>\LWR@section</code> : Combined higher-level sections together into files.	119
Docs: multirow browser bug workaround.	60, 159	<code>\LWR@subhtml elementclass</code> : Factored code.	98
Docs: Math options.	51	<code>\SetHTMLFileNumber</code> : Add: Control file numbers.	91
Docs: Table: Cross-referencing data structures, updated.	184	<code>\cpagerefFor</code> : User-redefinable word for page references.	239
File: <code>lwarp_mathjax.txt</code> added.	1	<code>\dotfill</code> : Inserts an ellipsis.	250
File: <code>lwarp.css</code> : <code>tnoteitemheader</code> added	1	<code>\hfill</code> : Inserts a <code>\qqquad</code>	250
Introduction: MathJax support mentioned.	18	<code>\hrulefill</code> : Inserts a short rule.	250
MathJax support added.	210, 215, 216	<code>\hyperindexref</code> : Print mode provided in case hyperref not used.	206
Options: <code>mathsvg</code> and <code>mathjax</code>	72	<code>\pageref</code> : Added.	190
Supports colored <code>\rule</code>	372	<code>\tracinglwarp</code> : Added.	83
titles: null <code>\pagestyle</code> and <code>\thispagestyle</code> for HTML.	79	<code>\warpOSwindows</code> : Auto-detects operating system.	85
<code>tikzpicture</code> : Adapts to <code>tikz</code> version.	240	General: 2017/02/09	1
<code>equation</code> : MathJax support.	213	<code>afterpage</code> : Added.	307
v0.20 <code>\BlockClassSingle</code> : Renamed from "LWR@htmldivclassline".	100	<code>alltt</code> : Added.	309
<code>\HTMLFileName</code> : No longer escape underscores.	91	<code>bookmark</code> : Added.	310
<code>\HomeHTMLFileName</code> : No longer escape underscores.	91	caption and subcaption supported.	1
<code>\InlineClass</code> : Renamed from "inlineclass".	100	cleveref and referencing patches: Applied <code>\AfterEndPreamble</code>	238
<code>\LWR@LwarpStart</code> : Fix: math cross references.	128	<code>draftwatermark</code> : Added.	311
<code>\LWR@closeparagraph</code> : <code>\unskip</code> extra spaces.	103	<code>eso-pic</code> : Added.	313
No break tags in the start/end of a tabular.	103	<code>everypage</code> : Added.	314
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	252	<code>extramarks</code> : Added.	314
<code>\LWR@filestart</code> : Adds meta description.	124	<code>fancyhdr</code> : Added.	315

hyperref: Additional user macros.	326	Page geometry set to 6in wide with large margins.	76
keyfloat: Added.	330	Parallel versions of aux files for print/HTML.	1
letterspace: User-interface emulated.	331	Removed reliance on make, grep, gawk.	1
listings: Added.	332	Tabular: \unskip extra spaces.	159
lrcaption: Added.	338	Test Suite: HTML meta descriptions.	1
lwarp-newproject: Added.	263	Verbatim: Added.	150
microtype: User-interface emulated.	346	verbatim: Added.	148
needspace: Added.	349	BlockClass: Added optional style.	99
nowidow: Added.	350	Renamed from "blockclass". ..	99
pagenote: Added.	351	LWR@nestspan: Fix: Minipages inside a span.	95
placeins: Added.	352		
ragged2e: Added.	353	v0.21	
setspace: Improved support. ..	354	\LWR@LwarpStart: Changed lateximages to a .txt file. .	127
textpos: Added.	355	\LWR@filestart: Skip title if not given.	124
titles: Added.	358	\LWR@newhtmlfile: Skip title if not given.	115
titlesec: Added.	360	\marginpar: Fixed source listing.	112
titletoc: Added.	361	General: 2017/02/23	1
titling: Improved compatibility.	362	fontenc: Added.	323
tocloft: Added.	363	fontspec: Added.	323
wallpaper: Added.	369	inputenc: Added.	329
wrapfig: Added.	370	newunicodechar: Added.	350
Added @, <, > columns.	159	Add: Errors for misplaced packages.	73
Added \NewHTMLdescription. .	109	Docs: Added internet class. .	24
Added single-expansion data arrays.	89	Docs: Added TeX2page, GladTeX.	24
Code factored into independent lwarp_html files.	307	Docs: Installing on Windows. .	30
Docs: Examples for generating HTML file names.	44	File lwarp_tutorial.txt added.	34
Docs: Improved index.	1	lwarpmk: Fix: lwarpmk again for Windows.	298
Enhanced titling support.	135	lwarpmk: Fix: lwarpmk limages for Windows.	298
File: lwarp.css: Minor fixes for validation.	1	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	298
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1		
Fix: \linewidth in a floatrow.	320		
Improved float caption type handling.	316		
Moved sidebar and example code to test suite.	1		

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		\AmS
\\$	208	amsmath (package) 79
\&	90, 2105	\and 132
\(3022	array (package) 183
\)	3022	AsciiDoc (program) 25
\@@setcpageref	3725	AsciiDoctor (program) 25
\@@setcref	3721	Asciidoctor-LaTeX (program) 25
\@@setcrefrange	3722	\attrib 56, 146
@author	132	\attribution 1720
\@begintheorem	1962	\author 54, 132, 1386
\@capytype	2766	autosec 118
\@currentlabel	111	
\@currentlabelname	2556	B
@date	132	babel (package) 63
\@dlbfloat	2749	\backmatter 1067
\@endtheorem	1972	BaseJobName (option) [lwrap-
\@float	2749	newproject] 263
\@fnssymbol	1670	\BaseJobname 466
\@makecaption	2768	BaseJobname (option) [lwrap] 72
\@maketitle	1681	baseline
\@nameedef	69	tabular 167
\@opargbegintheorem	1967	\bfseries 3922
@published	132	\BibTeX 4102
@subtitle	132	BlockClass (environment) 681
@title	132	\BlockClassSingle 693
\\	252	bookmark (package) 310
\$	209	booktabs (package) 311
\$\$	209	booleans:
A		CombineHigherDepths 44, 112
abstract (environment)	1711	FileSectionNames 44, 91
\abstractname	45, 1710	LWR@amsmultline 215
accents	117	LWR@doingapar 102
\addcontentsline	2817	LWR@doinghline 160
Adobe (program)	26	LWR@doingstartpars 102
\affiliation	1379	LWR@doingtbrule 160
afterpage (package)	307	LWR@freezethisfloat 195
algorithmic		LWR@infloatrow 231
with newfloat, trivfloat	368	LWR@intabularmetadata 161
algorithmicx (package)	308	LWR@minipagefullwidth 244
align (environment)	3234	LWR@minipagethispar 244
align* (environment)	3256	LWR@skippingmrowcell 161
alltt (package)	309	LWR@starredlongtable 176
		LWR@startedrow 160

LWR@tableparcell	160	LWR@tabletotalcolsnext	164
LWR@tikzbabel	240	LWR@thisfloat	195
LWR@tracinglwarp	83	SideTOCDepth	43, 202
LWR@verbtags	148	tocdepth	43
mathjax	71	\cpagerefFor	3724
usingOSwindows	85	\cref@currentlabel	111
warpingHTML	71	cross-references	
warpingprint	71	missing or incorrect	65
bugs	64	CSS	
BVerbatim (environment)	1948	file selection	56
		lwarp.css	55
		per HTML page	56
		project-specific changes	55
C			
calc (package)	78	D	
caption (package)	78	danger icon	69
\caption@begin	2798	\date	55
\caption@end	2798	debugging	64
\captionlistentry	2800	\DeclareGraphicsExtensions	3473
\captionof	2829	\DeclareGraphicsRule	3473
\CaptionSeparator	2767	Deja Vu	50
center (environment)	3452	description	
\centering	127	HTML meta tag	55, 109
\chapter	1185	description (environment)	2035
\citetitle	1727	displaymath (environment)	3031
classes:		\dotfill	3938
internet	24	draftwatermark (package)	311
cleveref (package)	238	E	
cmap (package)	51	\emph	3911
\color	373	\end@dlbfloat	2758
CombineHigherDepths (boolean)	44, 112	\end@float	2758
comment (package)	71	\enlargethispage	4037
Computer Modern	50	\enskip	252, 4003
\ConTeXt	4099	enumerate (environment)	2004
counters:		enumitem (package)	63, 78
FileDepth	43, 112	environ (package)	79
LWR@externalfilecnt	208	environments:	
LWR@htmlfilenumber	91	abstract	1711
LWR@latestautopage	196	align	3234
LWR@lateximagedepth	221	align*	3256
LWR@lateximagenumber	221	BlockClass	681
LWR@midrulecounter	172	BVerbatim	1948
LWR@minipagedepth	241	center	3452
LWR@nextautofloat	196	description	2035
LWR@nextautopage	196	displaymath	3031
LWR@nextequation	210	enumerate	2004
LWR@nextpage	221	equation	3098
LWR@prevFileDepth	118	flalign	3278
LWR@spandepth	101	flalign*	3300
LWR@tablecolspan	164		
LWR@tablecolwidth	164		
LWR@tabletotalcols	164		

gather* (environment)	3212	\includegraphics	3653
GELLMU (program)	24	InDesign (program)	26
generator		\InlineClass	701
HTML meta tag	125	inputenc (package)	50, 329
geometry (package)	76, 324	internet (class)	24
getttitlestring (package)	77	item, list, empty	53
GladTeX (program)	25	itemize (environment)	1986
glyphtounicode (file)	50	\itshape	3927
graphics (package)	227, 325		
\graphicspath	3471	J	
graphicx (package)	227, 325	JavaScript	
		MathJax	57, 206
H			
Hevea (program)	24	K	
\hfill	3936	keyfloat (package)	330
\HomeHTMLFileName	468	kvoptions (package)	71
\HomeHTMLFilename	43		
horizontal space		L	
between minipages	252	label	
\href	2701	in HTML	128
\hrulefill	3937	math environment	215
\hspace	252, 4030	language HTML metadata	124
HTML		\LaTeX	4085
commands for a successful conver-		LaTeX2HTML (program)	24
sion	53	\LaTeXe	4085
conversion settings	43	lateximage (environment)	3374
filename generation	44	\LateximageFontSizeName	3338
headings	68	lateximages	
meta tag		font size	57, 206, 221
description	55, 109	LaTeXML (program)	24
generator	125	lengths:	
viewport	125	HTMLleftmargini	56, 146, 147
tabular column conversion	168	HTMLvleftskip	56, 146, 147
\HTMLentity	456	LWR@minipageheight	242
\HTMLFileName	43, 467	LWR@minipagewidth	242
HTMLleftmargini (length)	56, 146, 147	LWR@tempheight	253
\HTMLunicode	457	LWR@tempraise	253
HTMLvleftskip (length)	56, 146, 147	LWR@tempwidth	253
\hyperindexref	2974	\VerbatimHTMLWidth	148
hyperref (package)	191, 326	vleftmargini	56, 146
\hypertoc	2916	vleftskip	56, 146, 147
\hypertocfloat	2925	letterspace (package)	331
		LibreOffice (program)	25
I		ligatures	50, 51, 76
icon		line numbers	69
warning	69	\linebreak	4031
ifplatform (package)	71	\LinkHome	516
images		list item, empty	53
graphicx package	227	listings (package)	332
in strange places	66	\listof	2883

lmodern (package)	50	LWR@colbeforespec	164
longtable (environment)	2	LWR@commentfooterstyle	221
longtable (package)	336	\LWR@copyfile	283
ltcaption (package)	338	\LWR@createautosec	1073
LuaLaTeX		\LWR@currentcss	869
detection	75	\LWR@currentHTMLdescription	884
file & section names	117	\LWR@descitem	2023
\LuaLaTeX	4093	\LWR@docmidrule	2303
LuaLaTeX (program) [requirement] ..	28	LWR@doingapar (boolean)	102
\LuaTeX	4093	LWR@doinghline (boolean)	160
LVerbatim (environment)	1959	LWR@doingstartpars (boolean)	102
lwrap		LWR@doingtbrule (boolean)	160
loading	51	\LWR@domulticolumn	2366
options	51	\LWR@doubledollar	3000
[lwrap]:		\LWR@endhideamsmath	3136
BaseJobname (option)	72	\LWR@endminipage	3793
mathjax (option)	51, 72	\LWR@endoffline	3981
mathsvg (option)	51, 72	\LWR@endsubminipage	3816
warpHTML (option)	51, 72	\LWR@ensuredoingapar	742
warpprint (option)	51, 71	LWR@externalfilecnt (counter) ..	208
lwrap (package)	51	\LWR@filenamenoblanks	935
[lwrap-newproject]:		\LWR@filestart	1237
BaseJobName (option)	263	\LWR@findword	219
lwrapmk (option)	263	\LWR@floatbegin	2738
warpHTML (option)	263	\LWR@floatend	2751
warpprint (option)	263	LWR@freezethisfloat (boolean) ..	195
lwrap-newproject (package)	263	\LWR@futurenonspacelot	2063
lwrap.css (file)	55, 266	\LWR@getexparray	452
lwrap_formal.css (file)	291	\LWR@getmynexttoken	2070
lwrap_html.xdy (file)	296	\LWR@hidelatexequation	3073
lwrap_mathjax.txt (file)	296	\LWR@hspace	4011
lwrap_sagebrush.css (file)	286	\LWR@htmlblockcomment	620
lwrap_tutorial.txt (file)	34	\LWR@htmlblocktag	628
lwrapmk (file)	298	\LWR@htmlblocktagcomment	633
lwrapmk (option) [lwrap-newproject]	263	\LWR@htmlclosecomment	607
lwrapmk.conf (file)	261	\LWR@htmlcomment	616
\LWR@addmathjax	3087	\LWR@htmldivclass	658
\LWR@afterendverbatim	1800	\LWR@htmldivclassend	661
LWR@amsmultline (boolean)	215	\LWR@htmldivclassend	671
\LWR@atbeginverbatim	1792	\LWR@htmldivclassend	647
\LWR@beginhideamsmath	3128	\LWR@htmldivclassend	652
\LWR@botnavigation	524	\LWR@htmldivclassend	664
\LWR@caption@begin	2780	\LWR@htmldivclassend	674
\LWR@caption@end	2789	LWR@htmlfilenumber (counter)	91
\LWR@clearmidrules	2284	\LWR@htmlfileref	2582
\LWR@closeparagraph	768	\LWR@htmlmathlabel	3118
\LWR@closeprevious	544	\LWR@htmlopencomment	607
\LWR@closetabledatcell	2084	\LWR@htmlrefsectionfilename	509
LWR@colafterspec	164	\LWR@HTMLsanitize	3341
LWR@colatspec	164	\LWR@HTMLsanitizeexpand	3359

\LWR@htmlsectionfilename	478	\LWR@origcolspec	2136
\LWR@htmlspan	582	\LWR@originname	3665
\LWR@htmlspanclass	590	\LWR@originnames	3672
\LWR@htmltag	598	\LWR@parseaftercolumn	2170
\LWR@htmltagc	566	\LWR@parseatcolumn	2141
\LWR@includegraphicsb	3564	\LWR@parsebeforecolumn	2162
\LWR@indexitem	2962, 2966, 2970	\LWR@parsenormalcolumn	2181
\LWR@indexsection	2946	\LWR@parsepcolumn	2189
LWR@infloatrow (boolean)	231	\LWR@parseskipcolumn	2179
LWR@intabularmetadata (boolean)	161	\LWR@parsetablecols	2193
\LWR@itemizeitem	1979	LWR@prevFileDepth (counter)	118
LWR@latestautopage (counter)	196	\LWR@printmccoldata	2332
LWR@lateximagedepth (counter)	221	\LWR@printmccoltype	2307
\LWR@lateximagedepthref	2587	\LWR@printpendingnotes	909
LWR@lateximagenumber (counter)	221	\LWR@printthetitle	1512
\LWR@lateximagenumberref	2590	\LWR@ProvidesPackageDrop	273
\LWR@longtabledatacaptiontag	2384	\LWR@ProvidesPackagePass	261
\LWR@lookforpackagename	222	\LWR@pushoneclose	1076
\LWR@LwarpEnd	1360, 4129	\LWR@requesttoc	1353
\LWR@LwarpStart	1291, 4129	\LWR@requirepackagenames	218
\LWR@makefnmark	3792	\LWR@restoremathlatexformatting	3055
\LWR@makefntext	3785	\LWR@rotboxorigin	3664
\LWR@maketitlesetup	1661	\LWR@rotstyle	3554
\LWR@maybeinthisfloat	2763	\LWR@rule	4038
\LWR@maybenewtablerow	2237	\LWR@scalestyle	3557
LWR@midrulecounter (counter)	172	\LWR@section	1083
LWR@midrules	172	\LWR@sectionnumber	1070
LWR@minipagedepth (counter)	241	\LWR@setexparray	449
LWR@minipagefullwidth (boolean)	244	\LWR@setlatestname	2558
LWR@minipageheight (length)	242	\LWR@sidetoc	2896
\LWR@minipagestartpars	3985	\LWR@singledollar	3012
\LWR@minipagestoppars	3991	LWR@skippingmrowcell (boolean)	161
LWR@minipagethispar (boolean)	244	LWR@spandepth (counter)	101
LWR@minipagewidth (length)	242	\LWR@splabel	2593
\LWR@multicolother	2327	LWR@starredlongtable (boolean)	176
\LWR@multicolpartext	2322	LWR@startedrow (boolean)	160
\LWR@mynexttoken	2062	\LWR@startnewdepth	1077
\LWR@mynexttoc	2841	\LWR@startpars	806
\LWR@nameref	2579	\LWR@startref	2635
LWR@nestspan (environment)	576	\LWR@stoppars	818
\LWR@newhtmlfile	1013	\LWR@stripperiod	2557
\LWR@newlabel	2628	\LWR@strresult	2135
LWR@nextautofloat (counter)	196	\LWR@subcmidrule	2295
LWR@nextautopage (counter)	196	\LWR@subhtml elementclass	638
LWR@nextequation (counter)	210	\LWR@subhyperref	2692
LWR@nextpage (counter)	221	\LWR@subhyperrefclass	2696
\LWR@noloadafter	81	\LWR@subinlineimage	2731
\LWR@noloadbefore	70	\LWR@sublabel	2617
\LWR@nullrule	3784	\LWR@subminipage	3807
\LWR@openparagraph	747	\LWR@subsublabel	2596

<code>\LWR@subtableofcontents</code>	2856	<code>math</code> (environment)	3030
<code>\LWR@syncmathjax</code>	3034	MathJax	
<code>\LWR@tablecolspec</code>	2134	mathjax option	72
<code>LWR@tablecolspos</code> (counter)	164	subequations	58, 207
<code>LWR@tablecolswidth</code> (counter)	164	summary	58, 207
<code>\LWR@tabledatacolumntag</code>	2468	tagged equations	58, 207
<code>\LWR@tabledata multicolumntag</code>	2448	<code>MathJax</code> (program)	58, 207
<code>\LWR@tabledata multirowtag</code>	2454	<code>MathJax</code> (program) [requirement]	28
<code>\LWR@tabledatasinglecolumntag</code>	2253	<code>mathjax</code> (boolean)	71
<code>LWR@tableparcell</code> (boolean)	160	<code>mathjax</code> (option) [lwrap]	51, 72
<code>LWR@tabletotalcols</code> (counter)	164	<code>mathsvg</code> (option) [lwrap]	51, 72
<code>LWR@tabletotalcolsnxt</code> (counter)	164	<code>mdframed</code> (package)	339
<code>LWR@tabular</code> (environment)	2514	<code>\mdseries</code>	3921
<code>\LWR@tabularendoffline</code>	2118	memoir	
<code>LWR@tempheight</code> (length)	253	verse	147
<code>LWR@tempraise</code> (length)	253	<code>memoir</code> (package)	147
<code>LWR@tempwidth</code> (length)	253	meta tag, HTML	
<code>\LWR@thisfilename</code>	933	description	55, 109
<code>LWR@thisfloat</code> (counter)	195	generator	125
<code>\LWR@thisnewfilename</code>	934	viewport	125
<code>LWR@tikzbabel</code> (boolean)	240	<code>\MetaLanguage</code>	45, 1236
<code>\LWR@titlingmaketitle</code>	1698	<code>microtype</code> (package)	51, 76, 346
<code>\LWR@topnavigation</code>	521	<code>\MiKTeX</code>	4109
<code>\LWR@traceinfo</code>	302	minipage	
<code>LWR@tracinglwrap</code> (boolean)	83	horizontal space between	252
<code>\LWR@Verbatimclass</code>	1807	in an <code>\fbox</code>	53
<code>\LWR@verbtags</code> (boolean)	148	<code>minipage</code> (environment)	3831
<code>\LWRsetnextfloat</code>	2771	<code>\minipagefullwidth</code>	3823
<code>\LyX</code>	4110	missing sections	43, 44
M			
<code>Madcap</code> (program)	26	<code>\mrowcell</code>	2510
<code>\mainmatter</code>	1060	<code>MS-Windows</code> (program)	51, 85
<code>\makebox</code>	3895	<code>multicol</code> (package)	347
<code>makeidx</code> (package)	77	<code>multirow</code> (package)	179, 349
<code>\MakeIndex</code>	4102	<code>multline</code> (environment)	3143
<code>\maketitle</code>	54, 139, 1673	<code>multline*</code> (environment)	3166
<code>\maketitlehookaa</code>	131	N	
<code>\maketitlehookaaa</code>	131	<code>\nameref</code>	2681
margin		<code>needspace</code> (package)	349
numbers	69	<code>newclude</code> (package)	63
tags	69	<code>\NewCSS</code>	42, 45, 56, 870
<code>\marginpar</code>	922	<code>newfloat</code>	
markup languages	26	with <code>trivfloat</code> , algorithmic	368
math		<code>\newfloat</code>	2
font size — SVG	57, 206, 221	<code>\newfloat</code> (package)	78
mathjax option	72	<code>\NewHTMLdescription</code>	46, 55, 885
MathJax summary	58, 207	<code>\newline</code>	3979
mathsvg option	72	<code>\newlist</code>	2053
SVG summary	57, 206	<code>\newtheorem</code>	154
		<code>newtxmath</code> (package)	63

newunicodechar (package)	50, 350	etoolbox	70
\nolinebreak	4032	everyhook	77
\nolinkurl	2713	everypage	314
\nopagebreak	4036	expl3	77
\notedivision	895	extramarks	314
\noteentry	901	fancyhdr	315
\notenuminnotes	899	fancyvrb	77
nowidow (package)	350	float	316
\numberline	2913	floatrow	318
numbers		fontenc	50, 323
left margin	69	fontspec	50, 323
O			
OpenOffice (program)	25	geometry	76, 324
options:		getttitlestring	77
[lwarp-newproject]:		graphics	227, 325
BaseJobName	263	graphicx	227, 325
lwarpmk	263	hyperref	191, 326
warpHTML	263	ifplatform	71
warpprint	263	inputenc	50, 329
[lwarp]:		keyfloat	330
BaseJobname	72	kvoptions	71
mathjax	51, 72	letterspace	331
mathsvg	51, 72	listings	332
warpHTML	51, 72	lmodern	50
warpprint	51, 71	longtable	336
\OSPathSymbol	349	ltcaption	338
P			
packages		lwarp	51
required	75	lwarp-newproject	263
packages:		makeidx	77
afterpage	307	mdframed	339
algorithmicx	308	memoir	147
alltt	309	microtype	51, 76, 346
amsmath	79	multicol	347
array	183	multirow	179, 349
babel	63	needspace	349
bookmark	310	newclade	63
booktabs	311	newfloat	78
calc	78	newtxmath	63
caption	78	newunicodechar	50, 350
cleveref	238	nowidow	350
cmap	51	pagenote	78, 351
comment	71	placeins	352
draftwatermark	311	ragged2e	353
enumitem	63, 78	rotating	354
environ	79	setspace	354
epigraph	312	siunitx	62, 227
eso-pic	313	subcaption	78
		textcomp	51
		textpos	355
		threeparttable	357
		tikz	240

titles	79, 358	\printindex	2949
titlesec	360	\printpublished	132, 1495, 1541
titletoc	361	\PrintStack	529
titling	78, 362	\printssubtitle	132, 1520, 1543
tocloft	363	\printthanks	131, 135
trivfloat	368	\printtitle	132, 1504, 1542
verse	147	problems	64
wallpaper	369	programs	
wrapfig	370	utility	27
xcolor	372	programs:	
xfrac	80, 375	[requirement]:	
xifthen	77	LuaLaTeX	28
xparse	76	MathJax	28
xstring	77	pdfcrop	28
zref	79	pdfLaTeX	28
\pagebreak	4033	pdfseparate	28, 32
pagenote (package)	78, 351	pdftocairo	28, 32
\pagenotesubhead	897	pdftotext	28, 32
\pagenumbering	211	perl	33
\pageref	2676	XeLaTeX	28
\pagerefPageFor	2675	Adobe	26
pages		AsciiDoc	25
inaccessible	43, 44	AsciiDoctor	25
\pagestyle	207	Asciidoctor-LaTeX	25
Pandoc (program)	25	Flare	26
\paragraph	1220	FrameMaker	26
\parbox	3889	GELLMU	24
\parsemulticolumnalignment	2354	GladTeX	25
\part	1174	Hevea	24
pdfcrop (program) [requirement]	28	InDesign	26
pdfLaTeX (program) [requirement]	28	LaTeX2HTML	24
pdfseparate (program) [requirement]	28, 32	LaTeXML	24
pdftocairo (program) [requirement]	28, 32	LibreOffice	25
pdftotext (program) [requirement]	28, 32	Madcap	26
Perl	33	MathJax	58, 207
perl (program) [requirement]	33	MS-Windows	51, 85
\phantomsection	4079	OpenOffice	25
picture (environment)	239, 3742	Pandoc	25
placeins (package)	352	Plastex	24
Plastex (program)	24	TeX2page	24
\popclose	422	TeX4ht	24
Poppler	28, 32	TtH	24
\postpublished	131	Windows	51, 85
\postsubtitle	131	Word	25
\prepublished	131	project.css (file)	55
\presubtitle	131, 141	project.lwarpmkconf (file)	265
\printauthor	132, 1529, 1544	project_html.tex (file)	265
\printdate	132, 1534, 1546	\published	55, 1396
		\pushclose	395

Q	
<code>\qqquad</code>	252 , 4002
<code>\quad</code>	252 , 3997
<code>quote</code> (environment)	1735
R	
<code>ragged2e</code> (package)	353
<code>\raggedleft</code>	127
<code>\raggedright</code>	127
<code>\raisebox</code>	3906
<code>\ref</code>	2664
<code>\reflectbox</code>	3710
<code>\renewlist</code>	2053
[requirement]:	
LuaLaTeX (program)	28
MathJax (program)	28
pdfcrop (program)	28
pdfLaTeX (program)	28
pdfseparate (program)	28 , 32
pdftocairo (program)	28 , 32
pdftotext (program)	28 , 32
perl (program)	33
XeLaTeX (program)	28
<code>\RequirePackage</code>	237
<code>\resizebox</code>	3713
<code>\restylefloat</code>	19
<code>\rmfamily</code>	3923
<code>\rotatebox</code>	3684
<code>rotating</code> (package)	354
<code>\rule</code>	4076
S	
<code>sample_project.css</code> (file)	55 , 295
<code>\sb</code>	3931
<code>\scalebox</code>	3699
<code>\scshape</code>	3928 , 3929
section	
depths	68
missing	43 , 44
<code>\section</code>	1198
<code>\SetFirstPageTop</code>	45 , 847
<code>\SetHTMLFileNumber</code>	469
<code>\SetPageBottom</code>	45 , 132 , 863
<code>\SetPageTop</code>	45 , 855
<code>setspace</code> (package)	354
settings	
css project-specific	55
CSS selection	56
HTML conversion	43
lwrap package options	51
selecting output	51
title page	54
<code>\sffamily</code>	3924
<code>\sfrac</code>	375
<code>SideTOCDepth</code> (counter)	43 , 202
<code>\sidetocname</code>	2893
<code>siunitx</code> (package)	62 , 227
<code>\sp</code>	3930
space	
horizontal between minipages ...	252
stack depths	68
<code>subcaption</code> (package)	78
subequations	
MathJax	58 , 207
<code>\subparagraph</code>	1227
<code>\subsection</code>	1205
<code>\subsubsection</code>	1212
<code>\subtitle</code>	55 , 1407
SVG	
math summary	57 , 206
mathsvg option	72
T	
<code>\tableofcontents</code>	45
tabular	
baseline	167
HTML columnn conversion ...	168
tagged equations	
Mathjax	58 , 207
<code>\TeX</code>	4082
<code>TeX2page</code> (program)	24
<code>TeX4ht</code> (program)	24
<code>\textbf</code>	3913
<code>textcomp</code> (package)	51
<code>\textgreater</code>	459
<code>\textit</code>	3918
<code>\textless</code>	459
<code>\textmd</code>	3912
<code>\textnormal</code>	3920
<code>textpos</code> (package)	355
<code>\textrm</code>	3914
<code>\textsc</code>	3919
<code>\textsf</code>	3915
<code>\textsubscript</code>	3933
<code>\textsuperscript</code>	3932
<code>\texttt</code>	3916
<code>\textup</code>	3917
<code>\tfl@chapter@fix</code>	368
<code>\thanks</code>	55
<code>\thanksmarkseries</code>	1702

