This document is an unofficial reference manual for \LaTeX, a document preparation system, version of May 2024.

This manual was originally translated from \texttt{LATEX.HLP} v1.0a in the VMS Help Library. The pre-translation version was written by George D. Greenwade of Sam Houston State University. The \LaTeX 2.09 version was written by Stephen Gilmore. The \LaTeX2e version was adapted from this by Torsten Martinsen. Karl Berry made further updates and additions, and gratefully acknowledges using \textit{Hypertext Help with \LaTeX}, by Sheldon Green, and \textit{\LaTeX Command Summary} (for \LaTeX 2.09) by L. Botway and C. Biemesderfer (published by the \TeX Users Group as \textit{\TeXu} number 10), as reference material. We also gratefully acknowledge additional material appearing in latex2e-reference by Martin Herbert Dietze. (From these references no text was directly copied.)


Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.
# Short Contents

1. About this document .................................................. 1
2. Overview of LaTeX .................................................... 2
3. Document classes ..................................................... 10
4. Fonts ................................................................. 14
5. Layout ................................................................. 26
6. Sectioning ............................................................. 36
7. Cross references ...................................................... 48
8. Environments ......................................................... 52
9. Line breaking ......................................................... 99
10. Page breaking ........................................................ 104
11. Footnotes ............................................................ 107
12. Definitions ........................................................... 112
13. Counters ............................................................. 133
14. Lengths ............................................................... 137
15. Making paragraphs .................................................. 144
16. Math formulas ........................................................ 148
17. Modes ................................................................. 177
18. Page styles ........................................................... 179
19. Spaces ................................................................. 184
20. Boxes ................................................................. 197
21. Graphics .............................................................. 204
22. Color ................................................................. 215
23. Special insertions ..................................................... 219
24. Splitting the input .................................................... 229
25. Front/back matter ..................................................... 233
26. Letters ................................................................. 247
27. Input/output .......................................................... 252
28. Command line interface ............................................ 259
A. Document templates ................................................. 264
Index ................................................................. 267
# Table of Contents

1 About this document ............................................. 1

2 Overview of \LaTeX .............................................. 2
   2.1 Starting and ending ........................................ 2
   2.2 Output files ............................................... 3
   2.3 \TeX engines ............................................... 3
   2.4 Input text .................................................. 5
      2.4.1 Input encodings ....................................... 5
      2.4.2 Ligatures ............................................. 5
      2.4.3 Special characters: \{ \} % $ & _ ^ # ~ .................. 6
   2.5 \LaTeX command syntax ..................................... 7
   2.6 Environment syntax ......................................... 7
   2.7 \documentmetadata: Producing tagged PDF output ............. 8
   2.8 CTAN: The Comprehensive \TeX Archive Network .............. 8

3 Document classes .............................................. 10
   3.1 Document class options .................................... 10
   3.2 \usepackage: Additional packages .......................... 12
   3.3 Class and package creation ................................ 12
      3.3.1 Class and package structure .......................... 12

4 Fonts .................................................................. 14
   4.1 fontenc package ............................................. 14
      4.1.1 \DeclareFontEncoding ................................. 15
      4.1.2 \DeclareTextAccent ................................. 16
      4.1.3 \DeclareTextAccentDefault ......................... 16
      4.1.4 \DeclareTextCommand & \ProvideTextCommand ....... 17
      4.1.5 \DeclareTextCommandDefault & \ProvideTextCommandDefault ............................................. 17
      4.1.6 \DeclareTextComposite ................................ 18
      4.1.7 \DeclareTextCompositeCommand ..................... 18
      4.1.8 \DeclareTextSymbol .................................. 18
      4.1.9 \DeclareTextSymbolDefault ......................... 19
      4.1.10 \LastDeclaredEncoding ............................. 19
      4.1.11 \UseTextSymbol & \UseTextAccent .................. 19
   4.2 Font styles .................................................. 20
   4.3 Font sizes ................................................... 22
   4.4 Low-level font commands ................................... 23

5 Layout .......................................................... 26
   5.1 \onecolumn ................................................ 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.16</td>
<td>list</td>
<td>65</td>
</tr>
<tr>
<td>8.16.1</td>
<td>\item</td>
<td>70</td>
</tr>
<tr>
<td>8.16.2</td>
<td>trivlist</td>
<td>70</td>
</tr>
<tr>
<td>8.17</td>
<td>math</td>
<td>71</td>
</tr>
<tr>
<td>8.18</td>
<td>minipage</td>
<td>71</td>
</tr>
<tr>
<td>8.19</td>
<td>picture</td>
<td>73</td>
</tr>
<tr>
<td>8.19.1</td>
<td>\put</td>
<td>75</td>
</tr>
<tr>
<td>8.19.2</td>
<td>\multiput</td>
<td>76</td>
</tr>
<tr>
<td>8.19.3</td>
<td>\qbezier</td>
<td>76</td>
</tr>
<tr>
<td>8.19.4</td>
<td>\graphpaper</td>
<td>77</td>
</tr>
<tr>
<td>8.19.5</td>
<td>\line</td>
<td>77</td>
</tr>
<tr>
<td>8.19.6</td>
<td>\linethickness</td>
<td>78</td>
</tr>
<tr>
<td>8.19.7</td>
<td>\thinline</td>
<td>78</td>
</tr>
<tr>
<td>8.19.8</td>
<td>\thicklines</td>
<td>78</td>
</tr>
<tr>
<td>8.19.9</td>
<td>\circle</td>
<td>78</td>
</tr>
<tr>
<td>8.19.10</td>
<td>\oval</td>
<td>78</td>
</tr>
<tr>
<td>8.19.11</td>
<td>\shortstack</td>
<td>79</td>
</tr>
<tr>
<td>8.19.12</td>
<td>\vector</td>
<td>80</td>
</tr>
<tr>
<td>8.19.13</td>
<td>\makebox (picture)</td>
<td>80</td>
</tr>
<tr>
<td>8.19.14</td>
<td>\framebox (picture)</td>
<td>80</td>
</tr>
<tr>
<td>8.19.15</td>
<td>\frame</td>
<td>81</td>
</tr>
<tr>
<td>8.19.16</td>
<td>\dashbox</td>
<td>81</td>
</tr>
<tr>
<td>8.20</td>
<td>quotation &amp; quote</td>
<td>82</td>
</tr>
<tr>
<td>8.21</td>
<td>tabbing</td>
<td>82</td>
</tr>
<tr>
<td>8.22</td>
<td>table</td>
<td>84</td>
</tr>
<tr>
<td>8.23</td>
<td>tabular</td>
<td>85</td>
</tr>
<tr>
<td>8.23.1</td>
<td>\multicolumn</td>
<td>88</td>
</tr>
<tr>
<td>8.23.2</td>
<td>\vline</td>
<td>90</td>
</tr>
<tr>
<td>8.23.3</td>
<td>\cline</td>
<td>90</td>
</tr>
<tr>
<td>8.23.4</td>
<td>\hline</td>
<td>90</td>
</tr>
<tr>
<td>8.24</td>
<td>thebibliography</td>
<td>91</td>
</tr>
<tr>
<td>8.24.1</td>
<td>\bibitem</td>
<td>91</td>
</tr>
<tr>
<td>8.24.2</td>
<td>\cite</td>
<td>92</td>
</tr>
<tr>
<td>8.24.3</td>
<td>\nocite</td>
<td>93</td>
</tr>
<tr>
<td>8.24.4</td>
<td>Using Bib\TeX</td>
<td>93</td>
</tr>
<tr>
<td>8.24.4.1</td>
<td>Bib\TeX error messages</td>
<td>94</td>
</tr>
<tr>
<td>8.25</td>
<td>theorem</td>
<td>95</td>
</tr>
<tr>
<td>8.26</td>
<td>titlepage</td>
<td>95</td>
</tr>
<tr>
<td>8.27</td>
<td>verbatim</td>
<td>96</td>
</tr>
<tr>
<td>8.27.1</td>
<td>\verb</td>
<td>97</td>
</tr>
<tr>
<td>8.28</td>
<td>verse</td>
<td>97</td>
</tr>
</tbody>
</table>
9 Line breaking ................................................. 99
  9.1 \ .................................................... 99
  9.2 \
obeycr & \restorecr .................................. 100
  9.3 newline ................................................ 101
  9.4 \- (discretionary hyphen) ............................ 101
  9.5 \slash: breakable `/` .................................. 101
  9.6 \discretionary (generalized hyphenation point) .... 102
  9.7 \fussy & \sloppy ...................................... 102
    9.7.1 sloppypar ........................................ 102
  9.8 \hyphenation ........................................... 103
  9.9 \linebreak & \nolinebreak ............................ 103

10 Page breaking ............................................. 104
  10.1 \clearpage & \cleardoublepage ....................... 104
  10.2 \newpage ............................................... 105
  10.3 \enlargethispage ..................................... 105
  10.4 \pagebreak & \nopagebreak ........................... 106

11 Footnotes ................................................ 107
  11.1 \footnote .............................................. 107
  11.2 \footnotemark ........................................ 108
  11.3 \footnotetext ........................................ 109
  11.4 Footnotes in section headings ........................ 109
  11.5 Footnotes in a table ................................. 110
  11.6 Footnotes of footnotes .............................. 111

12 Definitions ............................................. 112
  12.1 \newcommand & \renewcommand ......................... 112
    12.1.1 Control sequence, control word and control symbol .... 114
  12.2 \providecommand ...................................... 114
  12.3 \makeatletter & \makeatother ......................... 115
  12.4 \@ifstar ............................................... 115
  12.5 \newcounter: Allocating a counter ................... 117
  12.6 \newlength ........................................... 117
  12.7 \newsavebox .......................................... 118
  12.8 \newenvironment & \renewenvironment .................. 118
  12.9 \newtheorem .......................................... 120
  12.10 \newfont ............................................ 122
  12.11 \protect ............................................ 123
  12.12 \ignorespaces & \ignorespacesafterend ............. 123
  12.13 xspace package ..................................... 125
  12.14 Class and package commands ......................... 125
    12.14.1 \AtBeginDvi & \AtEndDvi ......................... 125
    12.14.2 \AtEndOfClass & \AtEndOfPackage ............... 126
    12.14.3 \CheckCommand .................................. 126
12.14.4 \ClassError and \PackageError and others......... 126
12.14.5 \CurrentOption........................................ 127
12.14.6 \DeclareOption.................................... 127
12.14.7 \DeclareRobustCommand.............................. 127
12.14.8 \ExecuteOptions.................................... 128
12.14.9 \IfFileExists & InputIfFileExists.................. 128
12.14.10 \LoadClass & \LoadClassWithOptions.............. 129
12.14.11 \NeedsTeXFormat.................................... 129
12.14.12 \OptionNotUsed........................................ 130
12.14.13 \PassOptionsToClass & \PassOptionsToPackage.... 130
12.14.14 \ProcessOptions....................................... 130
12.14.15 \ProvidesClass & \ProvidesPackage................... 131
12.14.16 \ProvidesFile......................................... 132
12.14.17 \RequirePackage & \RequirePackageWithOptions... 132

13 Counters.................................................. 133
13.1 \alph \Alph \arabic \roman \Roman \fnsymbol: Printing
counters.................................................... 133
13.2 \usecounter................................................. 134
13.3 \value....................................................... 134
13.4 \setcounter............................................... 135
13.5 \addtocounter............................................. 135
13.6 \refstepcounter........................................... 135
13.7 \stepcounter............................................... 136
13.8 \day & \month & \year........................................ 136

14 Lengths................................................... 137
14.1 Units of length............................................ 138
14.2 \setlength................................................ 139
14.3 \addtolength............................................. 139
14.4 \settodepth............................................... 140
14.5 \settoheight............................................... 140
14.6 \settowidth............................................... 141
14.7 \stretch................................................... 141
14.8 Expressions................................................ 142

15 Making paragraphs.......................... 144
15.1 \par......................................................... 145
15.2 \indent & \noindent...................................... 145
15.3 \parindent & \parskip.................................... 146
15.4 Marginal notes............................................ 147
16 Math formulas .................................................. 148
  16.1 Subscripts & superscripts .................................. 149
  16.2 Math symbols .............................................. 149
    16.2.1 Arrows .................................................. 160
    16.2.2 \boldmath & \unboldmath ............................ 161
      16.2.2.1 \textbf: Individual bold math symbols .......... 161
      16.2.2.2 OpenType bold math .............................. 161
    16.2.3 Blackboard bold ...................................... 162
    16.2.4 Calligraphic .......................................... 162
    16.2.5 Delimiters ............................................ 162
      16.2.5.1 \left & \right .................................... 163
      16.2.5.2 \bigl, \bigr, etc .................................. 165
    16.2.6 Dots, horizontal or vertical ........................ 166
    16.2.7 Greek letters ......................................... 167
  16.3 Math functions ............................................. 168
  16.4 Math accents .............................................. 169
  16.5 Over- or under math ....................................... 170
  16.6 Spacing in math mode ..................................... 171
    16.6.1 \smash .................................................. 172
    16.6.2 \phantom & \vphantom & \hphantom ..................... 173
    16.6.3 \mathstrut ............................................ 174
  16.7 Math styles ................................................ 174
  16.8 Math miscellany .......................................... 175
    16.8.1 Colon character : & \colon ................................ 175
    16.8.2 \ast .................................................... 175
    16.8.3 \frac .................................................... 176
    16.8.4 \sqrt .................................................... 176
    16.8.5 \stackrel ............................................. 176

17 Modes ............................................................ 177
  17.1 \ensuremath .................................................. 177

18 Page styles ..................................................... 179
  18.1 \maketitle ................................................... 179
  18.2 \pagenumbering ............................................. 180
  18.3 \pagemode ................................................... 181
  18.4 \thepage .................................................... 182
  18.5 \thechapter ................................................. 183
19 Spaces .................................................. 184
  19.1 \enspace & \quad & \qquad .................................. 184
  19.2 \hspace ............................................. 184
  19.3 \hfill .............................................. 185
  19.4 \hss .................................................. 185
  19.5 \spacefactor ........................................ 186
    19.5.1 \@ ............................................. 187
    19.5.2 \frenchspacing & \nonfrenchspacing .............. 187
    19.5.3 \normalsfcodes ................................... 188
  19.6 Backslash-space, \ ................................ 188
  19.7 ~, \nobreakspace .................................... 189
  19.8 \thinspace & \negthinspace ............................ 190
  19.9 \.................................................. 190
  19.10 \hrulefill & \dotfill ................................ 191
  19.11 \bigskip & \medskip & \smallskip .................... 191
  19.12 \bigbreak & \medbreak & \smallbreak ............... 192
  19.13 \strut ............................................. 193
  19.14 \vspace ............................................ 194
  19.15 \vfill ............................................. 195
  19.16 \addvspace ........................................ 195

20 Boxes .................................................. 197
  20.1 \mbox & \makebox .................................... 197
  20.2 \fbox & \framebox .................................... 198
  20.3 \parbox ............................................. 199
  20.4 \raisebox .......................................... 200
  20.5 \sbox & \savebox .................................... 201
  20.6 \lrbox ............................................... 202
  20.7 \usebox ............................................. 202

21 Graphics .............................................. 204
  21.1 graphics package options .............................. 204
  21.2 graphics package configuration ....................... 205
    21.2.1 \graphicspath .................................. 205
    21.2.2 \DeclareGraphicsExtensions ....................... 206
    21.2.3 \DeclareGraphicsRule ................................ 207
  21.3 Commands for graphics ................................ 208
    21.3.1 \includegraphics ................................ 208
    21.3.2 \rotatebox ..................................... 213
    21.3.3 \scalebox ...................................... 214
    21.3.4 \resizebox ...................................... 214
## 22 Color ................................................. 215
22.1 color package options ......................... 215
22.2 Color models ...................................... 215
22.3 Commands for color ................................
  22.3.1 Define colors ................................ 216
  22.3.2 Colored text .................................. 216
  22.3.3 Colored boxes ................................. 218
  22.3.4 Colored pages ................................. 218

## 23 Special insertions .............................. 219
23.1 Printing special characters ....................... 219
23.2 Upper and lower case ............................. 219
23.3 Symbols by font position ......................... 220
23.4 Text symbols ..................................... 220
23.5 Accents ............................................. 224
  23.5.1 \accent ....................................... 225
23.6 Additional Latin letters ......................... 226
23.7 inputenc package .................................. 227
23.8 \rule ................................................. 228
23.9 \today ................................................ 228

## 24 Splitting the input ............................. 229
24.1 \endinput ........................................... 229
24.2 \include & \includeonly ........................... 230
24.3 \input ............................................... 232

## 25 Front/back matter .............................. 233
25.1 Table of contents, list of figures, list of tables . 233
  25.1.1 \dottedtocline ............................... 235
  25.1.2 \addcontentsline ............................. 235
  25.1.3 \addtocontents ................................ 236
  25.1.4 \contentsline .................................. 237
  25.1.5 \nofiles ....................................... 238
  25.1.6 \numberline .................................... 238
  25.2 Indexes ............................................ 238
    25.2.1 Produce the index manually ............... 239
    25.2.2 \index ......................................... 240
    25.2.3 makeindex .................................... 242
    25.2.4 \printindex .................................. 244
  25.3 Glossaries ....................................... 244
    25.3.1 \newglossaryentry .......................... 245
    25.3.2 \gls ........................................... 246
1 About this document

This is an unofficial reference manual for the \LaTeX2e document preparation system, which is a macro package for the \TeX typesetting program (see Chapter 2 [Overview], page 2).

This document’s home page is https://latexref.xyz; it has separate web pages for each topic. Alternatively, https://latexref.xyz/dev/latex2e.html has the entire document on a single page. For other output formats, the sources, and plenty more information, see https://latexref.xyz/dev/.

In this document, we will mostly just use ‘\LaTeX’ rather than ‘\LaTeX2e’, since the previous version of \LaTeX (2.09) was frozen decades ago.

\LaTeX is maintained by a group of volunteers (https://latex-project.org). The official documentation written by the \LaTeX project is available from their web site. The present document is completely unofficial and has not been written or reviewed by the \LaTeX maintainers. Do not send bug reports or anything else about this document to them. Instead, please send all comments to latexrefman@tug.org. This is a public list; you can (un)subscribe, view the archives, etc., at https://lists.tug.org/latexrefman.

This document is a reference, not a tutorial. There is a vast array of other information available about \LaTeX, at all levels. Here are a few introductions.

https://ctan.org/pkg/latex-doc-ptr
Two pages of recommended references to \LaTeX documentation.

https://ctan.org/pkg/first-latex-doc
Writing your first document, with a bit of both text and math.

https://ctan.org/pkg/lshort
A longer introduction to \LaTeX, translated to many languages.

https://tug.org/begin.html
Overview of getting started with \TeX and \LaTeX.
2 Overview of \LaTeX

\LaTeX{} is a system for typesetting documents. It was originally created by Leslie Lamport in 1984, but has been maintained by a group of volunteers for many years now (https://latex-project.org). It is widely used, particularly but not exclusively for mathematical and technical documents.

A \LaTeX{} user writes an input file containing text to be typeset along with interspersed commands. The default encoding for the text is UTF-8 (as of 2018). The commands specify, for example, how the text should be formatted.

\LaTeX{} is implemented as a set of so-called “macros” (a \TeX{} format) which use Donald E. Knuth’s \TeX{} typesetting program or one of its derivatives, collectively known as “engines”. Thus, the user produces output, typically PDF, by giving the input file to a \TeX{} engine. The following sections describe all this in more detail.

The term \LaTeX{} is also sometimes used to mean the language in which the input document is marked up, that is, to mean the set of commands available to a \LaTeX{} user.

The name \LaTeX{} is short for “Lamport \TeX{}”. It is pronounced LAH-teck or LAY-teck, or sometimes LAY-tecks. Inside a document, produce the logo with \LaTeX{}. Where use of the logo is not sensible, such as in plain text, write it as ‘\LaTeX’.

2.1 Starting and ending

\LaTeX{} files have a simple global structure, with a standard beginning and ending. Here is a small example:

\begin{verbatim}
\documentclass{article}
\begin{document}
Hello, \LaTeX\ world.
\end{document}
\end{verbatim}

Every \LaTeX{} document has a \begin{document} line and an \end{document} line. Here, the ‘article’ is the document class. It is implemented in a file article.cls. You can use any document class available on your system. A few document classes are defined by \LaTeX{} itself, and a vast array of others are available. See Chapter 3 [Document classes], page 10.

You can include other \LaTeX{} commands between the \documentclass and the \begin{document} commands. This area is called the preamble.

The \begin{document}... \end{document} pair defines an environment; the ‘document’ environment (and no others) is required in all \LaTeX{} documents (see Section 8.6 [document], page 57). \LaTeX{} provides many environments that are documented here (see Chapter 8 [Environments], page 52). Many more are available to you from external packages, most importantly those available at CTAN (see Section 2.8 [CTAN], page 8).

The following sections discuss how to produce PDF or other output from a \LaTeX{} input file.
Chapter 2: Overview of \LaTeX

2.2 Output files

\LaTeX{} produces a main output file and at least two auxiliary files. The main output file’s name ends in either .dvi or .pdf.

.dvi If \LaTeX{} is invoked with the system command latex then it produces a DeVice Independent file, with extension .dvi. You can view this file with a command such as xdvi, or convert it to a PostScript .ps file with dvips or to a Portable Document Format .pdf file with dvipdfmx. The contents of the file can be dumped in human-readable form with dvitype. A vast array of other DVI utility programs are available (https://mirror.ctan.org/dviware).

.pdf If \LaTeX{} is invoked via the system command pdflatex, among other commands (see Section 2.3 [\TeX{} engines], page 3), then the main output is a Portable Document Format (PDF) file. Typically this is a self-contained file, with all fonts and images included.

\LaTeX{} always produces at least two additional files.

.log This transcript file contains summary information such as a list of loaded packages. It also includes diagnostic messages and perhaps additional information for any errors.

.aux Auxiliary information is used by \LaTeX{} for things such as cross references. For example, the first time that \LaTeX{} finds a forward reference—a cross reference to something that has not yet appeared in the source—it will appear in the output as a doubled question mark ???. When the referred-to spot does eventually appear in the source then \LaTeX{} writes its location information to this .aux file. On the next invocation, \LaTeX{} reads the location information from this file and uses it to resolve the reference, replacing the double question mark with the remembered location.

\LaTeX{} may produce yet more files, characterized by the filename ending. These include a .lof file that is used to make a list of figures, a .lot file used to make a list of tables, and a .toc file used to make a table of contents (see Section 25.1 [Table of contents etc.], page 233). A particular class may create others; the list is open-ended.

2.3 \TeX{} engines

\LaTeX{} is a large set of commands (macros) that is executed by a \TeX{} program (see Chapter 2 [Overview], page 2). Such a set of commands is called a format, and is embodied in a binary .fmt file, which can be read much more quickly than the corresponding \TeX{} source.

This section gives a terse overview of the \TeX{} programs that are commonly available (see also Chapter 28 [Command line interface], page 259).

\texttt{latex} \texttt{pdflatex} In \TeX{} Live (https://tug.org/texlive), if \LaTeX{} is invoked via either the system command latex or pdflatex, then the pdf\TeX{} engine is run (https://ctan.org/pkg/pdftex). When invoked as latex, the main output is a .dvi file; as pdflatex, the main output is a .pdf file.

pdflatex incorporates the e-\TeX{} extensions to Knuth’s original program (https://ctan.org/pkg/etex), including additional programming features
and bi-directional typesetting, and has plenty of extensions of its own. e-\TeX is available on its own as the system command \texttt{etex}, but this is plain \TeX (and produces \texttt{.dvi}).

In other \TeX distributions, \texttt{latex} may invoke e-\TeX rather than pdf\TeX. In any case, the e-\TeX extensions can be assumed to be available in \LaTeX, and a few extensions beyond e-\TeX, particularly for file manipulation.

\textbf{\texttt{lualatex}}  If \LaTeX is invoked via the system command \texttt{lualatex}, the Lua\TeX engine is run (https://ctan.org/pkg/luatex). This program allows code written in the scripting language Lua (http://luatex.org) to interact with \TeX’s typesetting. Lua\TeX handles UTF-8 Unicode input natively, can handle OpenType and TrueType fonts, and produces a \texttt{.pdf} file by default. There is also \texttt{dvilualatex} to produce a \texttt{.dvi} file.

\textbf{\texttt{xelatex}}  If \LaTeX is invoked with the system command \texttt{xelatex}, the Xe\TeX engine is run (https://tug.org/xetex). Like Lua\TeX, Xe\TeX natively supports UTF-8 Unicode and TrueType and OpenType fonts, though the implementation is completely different, mainly using external libraries instead of internal code. Xe\TeX produces a \texttt{.pdf} file as output; it does not support DVI output.

Internally, Xe\TeX creates an \texttt{.xdv} file, a variant of DVI, and translates that to PDF using the (x)dvipdfmx program, but this process is automatic. The \texttt{.xdv} file is only useful for debugging.

\textbf{\texttt{hilatex}}  If \LaTeX is invoked via the system command \texttt{hilatex}, the Hi\TeX engine is run (https://ctan.org/pkg/hitex). This program produces its own format, named HINT, designed especially for high-quality typesetting on mobile devices.

\textbf{\texttt{platex \ uplatex}}  These commands provide significant additional support for Japanese and other languages; the \texttt{u} variant supports Unicode. See https://ctan.org/pkg/ptex and https://ctan.org/pkg/uptex.

As of 2019, there is a companion \texttt{-dev} command and format for all of the above, except \texttt{hitex}:

\texttt{dvilualatex-dev}
\texttt{latex-dev}
\texttt{lualatex-dev}
\texttt{pdflatex-dev}
\texttt{platex-dev}
\texttt{uplatex-dev}
\texttt{xelatex-dev}

These are candidates for an upcoming \LaTeX release. The main purpose is to find and address compatibility problems before an official release.

These \texttt{-dev} formats make it easy for anyone to help test documents and code: you can run, say, \texttt{pdflatex-dev} instead of \texttt{pdflatex}, without changing anything else in your environment. Indeed, it is easiest and most helpful to always run the \texttt{-dev} versions instead of bothering to switch back and forth. During quiet times after a release, the commands will be equivalent.
These are not daily snapshots or untested development code. They undergo the same extensive regression testing by the \LaTeX{} team before being released. For more information, see “The \LaTeX{} release workflow and the \LaTeX{} dev formats” by Frank Mittelbach, \textit{TUGboat} 40:2, \url{https://tug.org/TUGboat/tb40-2/tb125mitt-dev.pdf}.

\section*{2.4 Input text}

To a first approximation, most input characters in \LaTeX{} print as themselves. But there are exceptions, as discussed in the following sections.

\subsection*{2.4.1 Input encodings}

The input to \TeX{} (or any computer program) ultimately consists of a sequence of bytes. (Nowadays, a byte is almost universally an eight-bit number, i.e., an integer between 0 and 255, inclusive.) The input encoding defines how to interpret that sequence of bytes, and thus how \LaTeX{} behaves.

Today, by far the most common way to encode text is with UTF-8, a so-called “Unicode Transformation Format” which specifies how to transform a sequence of 8-bit bytes to Unicode code points, which are defined independent of any particular representation. The Unicode encoding defines code points for virtually all characters used today in written text.

When \TeX{} was created, Unicode and UTF-8 did not exist and the 7-bit ASCII encoding was by far the most widely used. So \TeX{} does not require Unicode for text input. UTF-8 is a superset of ASCII, so a pure 7-bit ASCII document is also UTF-8.

Since 2018, the default input encoding for \LaTeX{} is UTF-8. Some methods for handling documents written in some other encoding, such as ISO-8859-1 (Latin 1), are explained in Section 23.7 \texttt{[inputenc package]}, page 227.

You can easily find more about all these topics in any introductory computer text or online. For example, you might start at: \url{https://en.wikipedia.org/wiki/Unicode}.

\subsection*{2.4.2 Ligatures}

A ligature combines two or more letters (more generally, characters) into a single glyph. For example, in Latin-based typography, the two letters ‘f’ and ‘i’ are often combined into the glyph ‘fi’.

\LaTeX{} supports ligatures automatically. To continue the example, if the input has the word ‘fine’, written as four separate ASCII characters, \TeX{} will output the word ‘fine’ (with the default fonts), with three typeset glyphs.

In traditional \TeX{}, the available ligatures, if any, are defined by the current font. \TeX{} also uses the ligature mechanism to produce a few typographical characters which were not available in any computer encoding when \TeX{} was invented. In all, in the original Computer Modern fonts, the following input character sequences are defined to lead to ligatures:

\begin{itemize}
  \item ‘ff’ \quad ff (ff ligature, U+FB00)
  \item ‘fi’ \quad fi (fi ligature, U+FB01)
  \item ‘fl’ \quad fl (fl ligature, U+FB02)
  \item ‘ffi’ \quad ffi (ffi ligature, U+FB03)
\end{itemize}
Chapter 2: Overview of \LaTeX 6

'\textbackslash' \textbackslash (f-ligature, U+FB04)

' '' (left double quotation mark, U+201C)

' " (right double quotation mark, U+201D)

'--' – (en-dash, U+2013)

'---' — (em-dash, U+2014)

'!`' ¡ (inverted exclamation mark, U+00A1)

'?`' ¿ (inverted question mark, U+00BF)

(For the f-ligatures above, the text in parentheses shows the individual characters, so in the typeset output you can easily see the difference between the ligature and the original character sequence.)

Nowadays it’s usually possible to directly input the punctuation characters as Unicode characters, and \LaTeX supports that (see previous section). But even today, it can still often be useful to use the ASCII ligature input form; for example, the difference between an en-dash and em-dash, as a single glyph, can be all but impossible to discern, but the difference between two and three ASCII hyphen characters is clear. Similarly with quotation marks, in some fonts.

Thus, even the engines with native support for UTF-8, namely Lua\TeX and Xe\TeX, also support the ASCII ligature input sequences by default, independent of the font used. They also need to do so for compatibility.

By the way, the f-ligatures are also available in Unicode (the “Alphabetic Presentation Forms” block starting at U+FB00), but it’s almost never desirable to use them as input characters, since in principle it should be up to the typesetter and the current font whether to use ligatures. Also, in practice, using them will typically cause searches to fail, that is, a search for the two characters ‘\textbackslash fi’ will not be matched by the ligature ‘fi’ at U+FB01.

2.4.3 Special characters: \{ \} % $ & _ ^ # ~

Besides ligatures (see previous section), a few individual characters have special meaning to \LaTeX. They are called reserved characters or special characters. Here they are:

'\' Introduces a command name, as seen throughout this manual.

'{' Delimits a required argument to a command or a level of grouping, as seen throughout this manual.

'\%' Starts a comment: the ‘%’ and all remaining characters on the current line are ignored.

'\$' Starts and ends math mode (see Chapter 16 [Math formulas], page 148).

'&' Separates cells in a table (see Section 8.23 [tabular], page 85).

'_' Introduce a subscript or superscript, respectively, in math (see Section 16.1 [Subscripts & superscripts], page 149); they produce an error outside math mode. As a little-used special feature, two superscript characters in a row can introduce special notation for an arbitrary character.
‘#’ Stands for arguments in a macro definition (see Section 12.1 [\newcommand & \renewcommand], page 112).

‘-’ Produces a nonbreakable interword space (see Section 19.7 ["], page 189).

See Section 23.1 [Printing special characters], page 219, for how to typeset these characters when you need them literally.

### 2.5 \LaTeX command syntax

In the \LaTeX input file, a command name starts with a backslash character, \. The name itself then consists of either (a) a string of letters or (b) a single non-letter.

\LaTeX commands names are case sensitive; for example, \texttt{\pagebreak} differs from \texttt{\Pagebreak} (the latter is not a standard command). Most command names are lowercase, but in any event you must enter all commands in the same case as they are defined.

A command may be followed by zero, one, or more arguments. These arguments may be either required or optional. Required arguments are contained in curly braces, \{\ldots\}. Optional arguments are contained in square brackets, \[\ldots\]. Generally, but not universally, if the command accepts an optional argument, it comes first, before any required arguments; optional arguments could come after required arguments, or both before and after.

Inside of an optional argument, to use the character close square bracket (\}) hide it inside curly braces, as in \texttt{\item\text{closing bracket \{\}}}. Similarly, if an optional argument comes last, with no required argument after it, then to make the first character of the following text be an open square bracket, hide it inside curly braces.

\LaTeX has the convention that some commands have a * form that is closely related to the form without a *, such as \texttt{\chapter} and \texttt{\chapter*}. The difference in behavior varies from command to command.

This manual describes all accepted options and *-forms for the commands it covers (barring unintentional omissions, a.k.a. bugs).

As of the 2020-10-01 release of \LaTeX, the expl3 and xparse packages are part of the \LaTeX2e format. They provide a completely different underlying programming language syntax. We won’t try to cover that in this document; see the related package documentation and other \LaTeX manuals.

### 2.6 Environment syntax

Synopsis:

\begin{verbatim}
\begin{environment-name}
...
\end{environment-name}
\end{verbatim}

An environment is an area of \LaTeX source, inside of which there is a distinct behavior. For instance, for poetry in \LaTeX put the lines between \texttt{\begin{verse}} and \texttt{\end{verse}}.

\begin{verbatim}
\begin{verse}
There once was a man from Nantucket \ \ \\
... \ \\
\end{verse}
\end{verbatim}
See Chapter 8 [Environments], page 52, for a list of environments. Particularly notable is that every \LaTeX document must have a \texttt{document} environment, a \texttt{\begin{document} ... \end{document}} pair.

The \texttt{environment-name} at the beginning must exactly match that at the end. This includes the case where \texttt{environment-name} ends in a star (*); both the \texttt{\begin} and \texttt{\end} texts must include the star.

Environments may have arguments, including optional arguments. This example produces a table. The first argument is optional (and causes the table to be aligned on its top row) while the second argument is required (it specifies the formatting of columns).

\begin{tabular}{t|l}
  ... & rows-of-table \\
\end{tabular}

2.7 \texttt{DocumentMetadata}: Producing tagged PDF output

The \texttt{DocumentMetadata} command was added to \LaTeX in 2022. It enables so-called “tagging” of the PDF output, aiding accessibility of the PDF. It is supported best with \texttt{LuaLaTeX}; \texttt{pdflatex} and \texttt{XeLaTeX} are supported as well as possible (see Section 2.3 [\TeX engines], page 3).

It is unlike nearly any other command in \LaTeX in that it must occur before the \texttt{documentclass} command that starts a \LaTeX document proper (see \texttt{\documentclass}, page 10). Therefore it must be called with \texttt{\RequirePackage} rather than \texttt{\usepackage} (see \texttt{\RequirePackage}, page 132).

This support is still in development, so we will not try to list all the possible settings. Please see the \texttt{documentmetadata-support-doc} document, part of the \texttt{latex-lab} package (https://ctan.org/pkg/latex-lab). Here is a simple example which enables most tagging currently implemented:

\begin{verbatim}
\DocumentMetadata{testphase={phase-III,firstaid}}
\documentclass{article}
... \\
\end{verbatim}

As you can see from the key name \texttt{testphase}, this is all still in an experimental phase. The \LaTeX developers strongly encourage users to give it a try and report problems, so it can be improved.

2.8 CTAN: The Comprehensive \TeX Archive Network

The Comprehensive \TeX Archive Network, CTAN, is the \TeX and \LaTeX community’s repository of free material. It is a set of Internet sites around the world that offer material related to \LaTeX for download. Visit CTAN on the web at https://ctan.org.

This material is organized into packages, discrete bundles that typically offer some coherent functionality and are maintained by one person or a small number of people. For instance, many publishers have a package that allows authors to format papers to that publisher’s specifications.

In addition to its massive holdings, the \texttt{ctan.org} web site offers features such as search by name or by functionality.
CTAN is not a single host, but instead is a set of hosts, one of which is the so-called “master”. The master host actively manages the material, for instance, by accepting uploads of new or updated packages. For many years, it has been hosted by the German \TeX\ group, DANTE e.V.

Other sites around the world help out by mirroring, that is, automatically syncing their collections with the master site and then in turn making their copies publicly available. This gives users close to their location better access and relieves the load on the master site. The list of mirrors is at https://ctan.org/mirrors.
3 Document classes

The document’s overall class is defined with the \documentclass command, which is normally the first command in a L\TeX{} source file.

\documentclass[options]{class}

The following document class names are built into \LaTeX{}. Many other document classes are available as separate packages (see Chapter 2 [Overview], page 2).

**article** For a journal article, a presentation, and miscellaneous general use.

**book** Full-length books, including chapters and possibly including front matter, such as a preface, and back matter, such as an appendix (see Chapter 25 [Front/back matter], page 233).

**letter** Mail, optionally including mailing labels (see Chapter 26 [Letters], page 247).

**report** For documents of length between an article and a book, such as technical reports or theses, which may contain several chapters.

**slides** For slide presentations—rarely used nowadays. The beamer package is perhaps the most prevalent replacement (\url{https://ctan.org/pkg/beamer}). See Section A.1 [beamer template], page 264, for a small template for a beamer document.

Standard options are described in the next section.

### 3.1 Document class options

You can specify global options or class options to the \documentclass command by enclosing them in square brackets. To specify more than one option, separate them with a comma.

\documentclass[option1,option2,...]{class}

\LaTeX{} automatically passes options specified for \documentclass on to any other loaded classes that can handle them.

Here is the list of the standard class options.

All of the standard classes except slides accept the following options for selecting the typeface size; the default is 10pt:

10pt 11pt 12pt

All of the standard classes accept these options for selecting the paper size (dimensions are listed height by width):

- **a4paper** 210 by 297 mm (about 8.25 by 11.75 inches)
- **a5paper** 148 by 210 mm (about 5.8 by 8.3 inches)
- **b5paper** 176 by 250 mm (about 6.9 by 9.8 inches)
- **executivepaper**
  7.25 by 10.5 inches
- **legalpaper**
  8.5 by 14 inches
letterpaper

8.5 by 11 inches (the default)

When using one of the engines \texttt{pdf\LaTeX}, \texttt{Lua\LaTeX}, or \texttt{Xe\LaTeX} (see Section 2.3 [\LaTeX engines], page 3), options other than \texttt{letterpaper} set the print area but you must also set the physical paper size. Usually, the \texttt{geometry} package is the best way to do that; it provides flexible ways of setting the print area and physical page size. Otherwise, setting the paper size is engine-dependent. For example, with \texttt{pdf\LaTeX}, you could include \texttt{\pdfpagewidth=\paperwidth} and \texttt{\pdfpageheight=\paperheight} in the preamble.

Miscellaneous other options:

\texttt{draft} \hspace{1cm} Mark (\texttt{draft}) or do not mark (\texttt{final}) overfull boxes with a black box in the margin; default is \texttt{final}.

\texttt{final} \hspace{1cm} Put displayed formulas flush left; default is centered.

\texttt{landscape} \hspace{1cm} Selects landscape format; default is portrait.

\texttt{leqno} \hspace{1cm} Put equation numbers on the left side of equations; default is the right side.

\texttt{openbib} \hspace{1cm} Use “open” bibliography format.

\texttt{titlepage} \hspace{1cm} \texttt{notitlepage} \hspace{1cm} Specifies whether there is a separate page for the title information and for the abstract also, if there is one. The default for the \texttt{report} class is \texttt{titlepage}, for the other classes it is \texttt{notitlepage}.

The following options are not available with the \texttt{slides} class.

\texttt{onecolumn} \hspace{1cm} Typeset in one or two columns; default is \texttt{onecolumn}.

\texttt{twocolumn} \hspace{1cm} Selects one- or two-sided layout; default is \texttt{oneside}, except that in the \texttt{book} class the default is \texttt{twoside}.

For one-sided printing, the text is centered on the page. For two-sided printing, the \texttt{\evensidemargin} (\texttt{\oddsidemargin}) parameter determines the distance on even (odd) numbered pages between the left side of the page and the text’s left margin, with \texttt{\oddsidemargin} being 40\% of the difference between \texttt{\paperwidth} and \texttt{\textwidth}, and \texttt{\evensidemargin} is the remainder.

\texttt{openright} \hspace{1cm} Determines if a chapter should start on a right-hand page; default is \texttt{openright} for \texttt{book}, and \texttt{openany} for \texttt{report}.

The \texttt{slides} class offers the option \texttt{clock} for printing the time at the bottom of each note.
3.2 \usepackage: Additional packages

To load a package \texttt{pkg}, with the package options given in the comma-separated list \texttt{options}:

\[
\texttt{\usepackage[options]{pkg}[mindate]}
\]

To specify more than one package you can separate them with a comma, as in \texttt{\usepackage{pkg1,pkg2,...}}, or use multiple \texttt{\usepackage} commands.

If the \texttt{mindate} optional argument is given, \LaTeX{} gives a warning if the loaded package has an earlier date, i.e., is too old. The \texttt{mindate} argument must be in the form YYYY/MM/DD. More info on this: \url{https://tex.stackexchange.com/questions/47743}.

\texttt{\usepackage} must be used in the document preamble, between the \texttt{\documentclass} declaration and the \texttt{\begin{document}}. Occasionally it is necessary to load packages before the \texttt{\documentclass}; see \texttt{\RequirePackage} for that (see \texttt{\RequirePackage}, page 132).

Any options given in the global \texttt{\documentclass} command that are unknown to the selected document class are passed on to the packages loaded with \texttt{\usepackage}.

3.3 Class and package creation

You can create new document classes and new packages. For instance, if your memos must satisfy some local requirements, such as a standard header for each page, then you could create a new class \texttt{sNc\textbackslash m\textbackslash e\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash .\textbackslash c\textbackslash l\textbackslash s} and begin your documents with \texttt{\documentclass[sNc\textbackslash m\textbackslash e\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash .\textbackslash c\textbackslash l\textbackslash s]}.

What separates a package from a document class is that the commands in a package are useful across classes while those in a document class are specific to that class. Thus, a command to set page headers is for a package while a command to make the page headers be \textit{Memo from the SMC Math Department} is for a class.

Inside of a class or package definition you can use the at-sign @ as a character in command names without having to surround the code containing that command with \texttt{\makeatletter} and \texttt{\makeatother} (see Section 12.3 \texttt{\makeatletter} \texttt{\makeatother}, page 115). This allows you to create commands that users will not accidentally redefine.

It is also highly desirable to prefix class- or package-specific commands with your package name or similar string, to prevent your definitions from clashing with those from other packages. For instance, the class \texttt{sNc\textbackslash m\textbackslash e\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash .\textbackslash c\textbackslash l\textbackslash s} might have commands \texttt{sNc\textbackslash to\textbackslash l\textbackslash i\textbackslash s\textbackslash t\textbackslash}, \texttt{sNc\textbackslash f\textbackslash r\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash l\textbackslash i\textbackslash s\textbackslash t\textbackslash}, etc.

3.3.1 Class and package structure

A class file or package file typically has four parts.

1. In the \textit{identification part}, the file says that it is a \LaTeX{} package or class and describes itself, using the \texttt{\NeedsTeXFormat} and \texttt{\ProvidesClass} or \texttt{\ProvidesPackage} commands.

2. The \textit{preliminary declarations part} declares some commands and can also load other files. Usually these commands will be those needed for the code used in the next part. For example, an \texttt{sNc\textbackslash m\textbackslash e\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash .\textbackslash c\textbackslash l\textbackslash s} class might be called with an option to read in a file with a list of people for the to-head, as \texttt{\documentclass[mathto]{sNc\textbackslash m\textbackslash e\textbackslash m\textbackslash o\textbackslash n\textbackslash o\textbackslash t\textbackslash .\textbackslash c\textbackslash l\textbackslash s}}, and therefore needs to define a command \texttt{\newcommand{\setto}{\texttt{\{1\}{\texttt{\def@tolist{\texttt{\#1}}}}}} used in that file.

3. In the \textit{handle options part} the class or package declares and processes its options. Class options allow a user to start their document as \texttt{\documentclass[option

4. The \textit{handle options part} declares and processes its options. Class options allow a user to start their document as \texttt{\documentclass[option

list}{class name}, to modify the behavior of the class. An example is when you declare \documentclass[11pt]{article} to set the default document font size.

4. Finally, in the more declarations part the class or package usually does most of its work: declaring new variables, commands and fonts, and loading other files.

Here is a starting class file, which should be saved as stub.cls where \LaTeX{} can find it, for example in the same directory as the .tex file.

\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{stub}[2017/07/06 stub to start building classes from]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions\relax
\LoadClass{article}

It identifies itself, handles the class options via the default of passing them all to the article class, and then loads the article class to provide the basis for this class’s code.

For more, see the official guide for class and package writers, the Class Guide, at https://ctan.org/pkg/clsguide (much of the description here derives from this document), or the tutorial at https://tug.org/TUGboat/tb26-3/tb84heff.pdf.

See Section 12.14 [Class and package commands], page 125, for some of the commands specifically intended for class and package writers.
4 Fonts

\LaTeX comes with powerful font capacities. For one thing, its New Font Selection Scheme allows you to work easily with the font families in your document (for instance, see Section 4.2 [Font styles], page 20). And, \LaTeX documents can use most fonts that are available today, including versions of Times Roman, Helvetica, Courier, etc. (Note, though, that many fonts do not have support for mathematics.)

The first typeface in the \TeX world was the Computer Modern family, developed by Donald Knuth. It is the default for \LaTeX documents and is still the most widely used. But changing to another font often only involves a few commands. For instance, putting the following in your preamble gives you a Palatino-like font, which is handsome and more readable online than many other fonts, while still allowing you to typeset mathematics. (This example is from Michael Sharpe, https://math.ucsd.edu/~msharpe/RcntFnts.pdf.)

\begin{verbatim}
\usepackage[osf]{newpxtext} % osf for text, not math
\usepackage{cabin} % sans serif
\usepackage[varqu, varl]{inconsolata} % sans serif typewriter
\usepackage[bigdelims, vvarbb]{newpxmath} % bb from STIX
\usepackage[cal=boondoxo]{mathalfa} % mathcal
\end{verbatim}

In addition, the \texttt{xelatex} or \texttt{lualatex} engines allow you to use any fonts on your system that are in OpenType or TrueType format (see Section 2.3 [\TeX engines], page 3).

The \LaTeX Font Catalogue (https://tug.org/FontCatalogue) shows font sample graphics and copy-and-pasteable source to use many fonts, including many with support for mathematics. It aims to cover all Latin alphabet free fonts available for easy use with \LaTeX.

More information is also available from the \TeX Users Group, at https://www.tug.org/fonts/.

4.1 \texttt{fontenc} package

Synopsis:

\begin{verbatim}
\usepackage[font_encoding]{fontenc}
\end{verbatim}

or

\begin{verbatim}
\usepackage[font_encoding1, font_encoding2, ...]{fontenc}
\end{verbatim}

Specify the font encodings. A font encoding is a mapping of the character codes to the font glyphs that are used to typeset your output.

This package only applies if you use the \texttt{pdflatex} engine (see Section 2.3 [\TeX engines], page 3). If you use the \texttt{xelatex} or \texttt{lualatex} engine then instead use the \texttt{fontspec} package.

\TeX’s original font family, Computer Modern, has a limited character set. For instance, to make common accented characters you must use \texttt{\accent} (see Section 23.5.1 [\accent], page 225) but this disables hyphenation. \TeX users have agreed on a number of standards to access the larger sets of characters provided by modern fonts. If you are using \texttt{pdflatex} then put this in the preamble

\begin{verbatim}
\usepackage[T1]{fontenc}
\end{verbatim}

gives you support for the most widespread European languages, including French, German, Italian, Polish, and others. In particular, if you have words with accented letters then \LaTeX
will hyphenate them and your output can be copied and pasted. (The optional second line allows you to directly enter accented characters into your source file.)

If you are using an encoding such as T1 and the characters appear blurry or do not magnify well then your fonts may be bitmapped, sometimes called raster or Type 3. You want vector fonts. Use a package such as \texttt{lm\-modern} or \texttt{cm\-super} to get a font that extends \LaTeX{}’s default using vector fonts.

For each \texttt{font\_encoding} given as an option but not already declared, this package loads the encoding definition files, named \texttt{font\_encodingenc.def}. It also sets \texttt{\encodingdefault} to be the last encoding in the option list.

These are the common values for \texttt{font\_encoding}:

- \texttt{OT1} The original 7-bit encoding for \TeX{}. Limited to mostly English characters.
- \texttt{OMS}, \texttt{OML} Math symbols and math letters encoding.
- \texttt{T1} \TeX{} text extended. Sometimes called the Cork encoding for the users group meeting where it was developed (1990). Gives access to most European accented characters. The most common option for this package.
- \texttt{TS1} Text Companion encoding.

\LaTeX{}’s default is to load \texttt{OML}, \texttt{T1}, \texttt{OT1}, and then \texttt{OMS}, and set the default to \texttt{OT1}.

Even if you do not use accented letters, you may need to specify a font encoding if your font requires it.

If you use \texttt{T1} encoded fonts other than the default Computer Modern family then you may need to load the package that selects your fonts before loading \texttt{fontenc}, to prevent the system from loading any \texttt{T1} encoded fonts from the default.

The \LaTeX{} team reserves encoding names starting with: ‘T’ for the standard text encodings with 256 characters, ‘TS’ for symbols that extend the corresponding T encodings, ‘X’ for test encodings, ‘M’ for standard math encodings with 256 characters, ‘A’ for special applications, ‘OT’ for standard text encodings with 128 characters, and ‘OM’ for standard math encodings with 128 characters (‘O’ stands for ‘obsolete’).

This package provides a number of commands, detailed below. Many of them are encoding-specific, so if you have defined a command that works for one encoding but the current encoding is different then the command is not in effect.

### 4.1.1 \texttt{\DeclareFontEncoding}

**Synopsis:**

\begin{verbatim}
\DeclareFontEncoding{encoding}{text-settings}{math-settings}
\end{verbatim}

\texttt{\DeclareFontEncoding{encoding}}\{\texttt{text-settings}\}\{\texttt{math-settings}\}

Declare the font encoding \texttt{encoding}. It also saves the value of \texttt{encoding} in \texttt{\LastDeclaredEncoding} (see Section 4.1.10 \texttt{\LastDeclaredEncoding}, page 19).

The file \texttt{tienc.def} contains this line (followed by many others).

\begin{verbatim}
\DeclareFontEncoding{T1}{}{}{}
\end{verbatim}

The \texttt{text-settings} are the commands that \LaTeX{} will run every time it switches from one encoding to another with the \texttt{\selectfont} and \texttt{\fontencoding} commands. The \texttt{math-settings} are the commands that \LaTeX{} will use whenever the font is accessed as a math alphabet.
LaTeX ignores any space characters inside text-settings and math-settings, to prevent unintended spaces in the output.

If you invent an encoding you should pick a two or three letter name starting with ‘L’ for ‘local’, or ‘E’ for ‘experimental’.

Note that output encoding files may be read several times by LaTeX so using, e.g., newcommand may cause an error. In addition, such files should contain \ProvidesFile line (see Section 12.14 [Class and package commands], page 125).

Note also that you should use the \...Default commands only in a package, not in the encoding definition files, since those files should only contain declarations specific to that encoding.

4.1.2 \DeclareTextAccent

Synopsis:

\DeclareTextAccent{cmd}{encoding}{slot}

Define an accent, to be put on top of other glyphs, in the encoding encoding at the location slot.

A slot is the number identifying a glyph within a font.

This line from tienc.def declares that to make a circumflex accent as in ^A, the system will put the accent in slot 2 over the ‘A’ character, which is represented in ASCII as 65. (This holds unless there is a relevant \DeclareTextComposite or \DeclareTextCompositeCommand declaration; see Section 4.1.6 [\DeclareTextComposite], page 18.)

\DeclareTextAccent{\^}{T1}{2}

If cmd has already been defined then \DeclareTextAccent does not give an error but it does log the redefinition in the transcript file.

4.1.3 \DeclareTextAccentDefault

Synopsis:

\DeclareTextAccentDefault{\cmd}{encoding}

If there is an encoding-specific accent command \cmd but there is no associated \DeclareTextAccent for that encoding then this command will pick up the slack, by saying to use it as described for encoding.

For example, to make the encoding OT1 be the default encoding for the accent \", declare this.

\DeclareTextAccentDefault{\"}{OT1}

If you issue a \" when the current encoding does not have a definition for that accent then LaTeX will use the definition from OT1

That is, this command is equivalent to this call (see Section 4.1.11 [\UseTextSymbol \& \UseTextAccent], page 19).

\DeclareTextCommandDefault[1]{\cmd}
{\UseTextAccent{encoding}{\cmd}{#1}}

Note that \DeclareTextAccentDefault works for any one-argument fontenc command, not just the accent command.
4.1.4 \DeclareTextCommand & \ProvideTextCommand

Synopsis, one of:
\DeclareTextCommand{cmd}{encoding}{defn}
\DeclareTextCommand{cmd}{encoding}[nargs]{defn}
\DeclareTextCommand{cmd}{encoding}[nargs][optargdefault]{defn}

or one of:
\ProvideTextCommand{cmd}{encoding}{defn}
\ProvideTextCommand{cmd}{encoding}[nargs]{defn}
\ProvideTextCommand{cmd}{encoding}[nargs][optargdefault]{defn}

Define the command \texttt{cmd}, which will be specific to one encoding. The command name \texttt{cmd} must be preceded by a backslash, \. These commands can only appear in the preamble. Redefining \texttt{cmd} does not cause an error. The defined command will be robust even if the code in \texttt{defn} is fragile (see Section 12.11 \protect, page 123).

For example, the file \texttt{t1enc.def} contains this line.
\DeclareTextCommand{\textperthousand}{T1}{\%\char 24 }

With that, you can express parts per thousand.
\usepackage[T1]{fontenc} \% in preamble
...
Legal limit is \( 0.8 \) \textperthousand.

If you change the font encoding to OT1 then you get an error like ‘LaTeX Error: Command \textperthousand unavailable in encoding OT1’.

The \texttt{ProvideTextCommand} variant does the same, except that it does nothing if \texttt{cmd} is already defined. The \texttt{DeclareTextSymbol} command is faster than this one for simple slot-to-glyph association (see Section 4.1.8 \texttt{DeclareTextSymbol}, page 18).

The optional \texttt{nargs} and \texttt{optargdefault} arguments play the same role here as in \texttt{newcommand} (see Section 12.1 \texttt{newcommand & \renewcommand}, page 112). Briefly, \texttt{nargs} is an integer from 0 to 9 specifying the number of arguments that the defined command \texttt{cmd} takes. This number includes any optional argument. Omitting this argument is the same as specifying 0, meaning that \texttt{cmd} will have no arguments. And, if \texttt{optargdefault} is present then the first argument of \texttt{cmd} is optional, with default value \texttt{optargdefault} (which may be the empty string). If \texttt{optargdefault} is not present then \texttt{cmd} does not take an optional argument.

4.1.5 \DeclareTextCommandDefault & \ProvideTextCommandDefault

Synopsis:
\DeclareTextCommandDefault{cmd}{defn}

or:
\ProvideTextCommandDefault{cmd}{defn}

Give a default definition for \texttt{cmd}, for when that command is not defined in the encoding currently in force. This default should only use encodings known to be available.

This makes \texttt{copyright} available.
\DeclareTextCommandDefault{\copyright}{\textcircled{c}}
It uses only an encoding (OMS) that is always available.

The \texttt{\DeclareTextCommandDefault} should not occur in the encoding definition files since those files should declare only commands for use when you select that encoding. It should instead be in a package.

As with the related non-default commands, the \texttt{\ProvideTextCommandDefault} has exactly the same behavior as \texttt{\DeclareTextCommandDefault} except that it does nothing if \texttt{\cmd} is already defined (see Section 4.1.4 \texttt{\DeclareTextCommand & \ProvideTextCommand}, page 17). So, packages can use it to provide fallbacks that other packages can improve upon.

### 4.1.6 \texttt{\DeclareTextComposite}

**Synopsis:**

\begin{verbatim}
\DeclareTextComposite{\cmd}{\encoding}{\simpleobject}{\slot}
\end{verbatim}

Access an accented glyph directly, that is, without having to put an accent over a separate character.

This line from \texttt{t1enc.def} means that \texttt{^o} will cause \LaTeX{} to typeset lowercase ‘o’ by taking the character directly from slot 224 in the font.

\begin{verbatim}
\DeclareTextComposite{^o}{T1}{o}{244}
\end{verbatim}

See Section 4.1 [fontenc package], page 14, for a list of common encodings. The \texttt{\simpleobject} should be a single character or a single command. The \texttt{\slot} argument is usually a positive integer represented in decimal (although octal or hexadecimal are possible). Normally \texttt{\cmd} has already been declared for this encoding, either with \texttt{\DeclareTextAccent} or with a one-argument \texttt{\DeclareTextCommand}. In \texttt{t1enc.def}, the above line follows the \texttt{\DeclareTextAccent{^o}{T1}{2}} command.

### 4.1.7 \texttt{\DeclareTextCompositeCommand}

**Synopsis:**

\begin{verbatim}
\DeclareTextCompositeCommand{\cmd}{\encoding}{\arg}{\code}
\end{verbatim}

A more general version of \texttt{\DeclareTextComposite} that runs arbitrary code with \texttt{\cmd}.

This allows accents on ‘i’ to act like accents on dotless i, \texttt{\i}.

\begin{verbatim}
\DeclareTextCompositeCommand{^i}{OT1}{i}{^i}
\end{verbatim}

See Section 4.1 [fontenc package], page 14, for a list of common encodings. Normally \texttt{\cmd} will have already been declared with \texttt{\DeclareTextAccent} or as a one argument \texttt{\DeclareTextCommand}.

### 4.1.8 \texttt{\DeclareTextSymbol}

**Synopsis:**

\begin{verbatim}
\DeclareTextSymbol{\cmd}{\encoding}{\slot}
\end{verbatim}

Define a symbol in the encoding \texttt{\encoding} at the location \texttt{\slot}. Symbols defined in this way are for use in text, not mathematics.

For example, this line from \texttt{tienc.def} declares the number of the glyph to use for «, the left guillemet.

\begin{verbatim}
\DeclareTextSymbol{guillemetleft}{T1}{19}
\end{verbatim}
The command `\DeclareTextCommand{\guillemetleft}{T1}{\char 19}` has the same effect but is slower (see Section 4.1.4 [\DeclareTextCommand & \ProvideTextCommand], page 17).

See Section 4.1 [fontenc package], page 14, for a list of common encodings. The slot can be specified in decimal, or octal (as in '023), or hexadecimal (as in "13), although decimal has the advantage that single quote or double quote could be redefined by another package.

If `\cmd` has already been defined then \DeclareTextSymbol does not give an error but it does log the redefinition in the transcript file.

### 4.1.9 \DeclareTextSymbolDefault

Synopsis:

`\DeclareTextSymbolDefault{\cmd}{\encoding}`

If there is an encoding-specific symbol command `\cmd` but there is no associated \DeclareTextSymbol for that encoding, then this command will pick up the slack, by saying to get the symbol as described for `\encoding`.

For example, to declare that if the current encoding has no meaning for `\textdollar` then use the one from OT1, declare this.

`\DeclareTextSymbolDefault{\textdollar}{OT1}`

That is, this command is equivalent to this call (see Section 4.1.11 [\UseTextSymbol & \UseTextAccent], page 19).

`\DeclareTextCommandDefault{\cmd}{\UseTextSymbol{\encoding}{\cmd}}`

Note that \DeclareTextSymbolDefault can be used to define a default for any zero-argument fontenc command.

### 4.1.10 \LastDeclaredEncoding

Synopsis:

`\LastDeclaredEncoding`

Get the name of the most recently declared encoding. The \DeclareFontEncoding command stores the name so that it can be retrieved with this command (see Section 4.1.1 [\DeclareFontEncoding], page 15).

This relies on \LastDeclaredEncoding rather than give the name of the encoding explicitly.

`\DeclareFontEncoding{JH1}{}{}`

`\DeclareTextAccent{\'}{\LastDeclaredEncoding}{0}`

### 4.1.11 \UseTextSymbol & \UseTextAccent

Synopsis:

`\UseTextSymbol{\encoding}{\cmd}`

or:

`\UseTextAccent{\encoding}{\cmd}{\text}`

Use a symbol or accent not from the current encoding.
Chapter 4: Fonts

In general, to use a fontenc command in an encoding where it is not defined, and if the command has no arguments, then you can use it like this:

\UseTextSymbol{OT1}{\ss}

which is equivalent to this (note the outer braces form a group, so LATEX reverts back to the prior encoding after the \ss):

{\fontencoding{OT1}\selectfont\ss}

Similarly, to use a fontenc command in an encoding where it is not defined, and if the command has one argument, you can use it like this:

\UseTextAccent{OT1}{\'}{a}

which is equivalent to this (again note the outer braces forming a group):

{\fontencoding{OT1} \selectfont \'{\fontencoding{enc_in_use} \selectfont a}}

Here, enc_in_use is the encoding in force before this sequence of commands, so that ‘a’ is typeset using the current encoding and only the accent is taken from OT1.

4.2 Font styles

The following type style commands are supported by LATEX.

In the table below the listed commands, the \text... commands, are used with an argument as in \textit{text}. This is the preferred form. But shown after it in parenthesis is the corresponding declaration form, which is often useful. This form takes no arguments, as in {\itshape text}. The scope of the declaration form lasts until the next type style command or the end of the current group. In addition, each has an environment form such as \begin{itshape}...\end{itshape}, which we’ll describe further at the end of the section.

These commands, in any of the three forms, are cumulative; for instance you can get bold sans serif by saying either of \sffamily\bfseries or \bfseries\sffamily.

One advantage of these commands is that they automatically insert italic corrections if needed (see Section 19.9 [/], page 190). Specifically, they insert the italic correction unless the following character is in the list \nocorrlist, which by default consists of period and comma. To suppress the automatic insertion of italic correction, use \nocorr at the start or end of the command argument, such as \textit{\nocorr text} or \textsc{\text\nocorr}.

\textrm (\rmfamily)
Roman.
\textit (\itshape)
Italics.
\textmd (\mdseries)
Medium weight (default).
\textbf (\bfseries)
Boldface.
\textup (\upshape)
Upright (default).
\textsl (\slshape)
Slanted.
Chapter 4: Fonts

\textsf{\textttfamily} (\sffamily)
Sans serif.

\textsc{\scshape} (\scshape)
Small caps.

\texttt{\ttfamily} (\ttfamily)
Typewriter.

\textnormal{\normalfont} (\normalfont)
Main document font.

Although it also changes fonts, the \texttt{\textit{\textttfamily text}} command is semantic, for text to be emphasized, and should not be used as a substitute for \texttt{\textit{\textttfamily text}}. For example, \texttt{\textbf{\textit{\textttfamily start text end text middle text}}} will result in the \texttt{\textbf{\textit{\textttfamily start text end text}}} in italics, but \texttt{\textbf{\textit{\textttfamily middle text}}} will be in roman.

\LaTeX also provides the following commands, which unconditionally switch to the given style, that is, are \texttt{not} cumulative. They are used as declarations: \texttt{\textbf{\textit{\textttfamily cmd\ldots}}} instead of \texttt{\textbf{\textit{\textttfamily cmd\ldots}}}.

(The unconditional commands below are an older version of font switching. The earlier commands are an improvement in most circumstances. But sometimes an unconditional font switch is what is needed.)

\texttt{\textbf{bf}}
Switch to bold face.

\texttt{\textit{cal}}
Switch to calligraphic letters for math.

\texttt{\textit{it}}
Italics.

\texttt{\textit{rm}}
Roman.

\texttt{\textit{sc}}
Small caps.

\texttt{\textit{sf}}
Sans serif.

\texttt{\textit{sl}}
Slanted (oblique).

\texttt{\textit{tt}}
Typewriter (monospace, fixed-width).

The \texttt{\textbf{em}} command is the unconditional version of \texttt{\textit{emph}}.

The following commands are for use in math mode. They are not cumulative, so \texttt{\mathit{\textbf{mathbf{\mathit{\textttfamily symbol}}}}} does not create a boldface and italic \texttt{\textttfamily symbol}; instead, it will just be in italics. This is because typically math symbols need consistent typographic treatment, regardless of the surrounding environment.

\texttt{\textit{mathrm}}
Roman, for use in math mode.

\texttt{\textit{mathbf}}
Boldface, for use in math mode.

\texttt{\textit{mathsf}}
Sans serif, for use in math mode.

\texttt{\textit{mathit}}
Typewriter, for use in math mode.

\texttt{\textit{mathit}} (\texttt{\textit{mit}})
Italics, for use in math mode.
\texttt{\textbackslash mathnormal}

For use in math mode, e.g., inside another type style declaration.

\texttt{\textbackslash mathcal}

Calligraphic letters, for use in math mode.

In addition, the command \texttt{\textbackslash mathversion{bold}} can be used for switching to bold letters and symbols in formulas. \texttt{\textbackslash mathversion{normal}} restores the default.

Finally, the command \texttt{\textbackslash oldstylenums{numerals}} will typeset so-called “old-style” numerals, which have differing heights and depths (and sometimes widths) from the standard “lining” numerals, which all have the same height as uppercase letters. \LaTeX{}’s default fonts support this, and will respect \texttt{\textbackslash textbf} (but not other styles; there are no italic old-style numerals in Computer Modern). Many other fonts have old-style numerals also; sometimes package options are provided to make them the default. FAQ entry: \url{https://www.texfaq.org/FAQ-osf}.

### 4.3 Font sizes

The following standard type size commands are supported by \LaTeX{}. The table shows the command name and the corresponding actual font size used (in points) with the ‘10pt’, ‘11pt’, and ‘12pt’ document size options, respectively (see Section 3.1 [Document class options], page 10).

<table>
<thead>
<tr>
<th>Command</th>
<th>10pt</th>
<th>11pt</th>
<th>12pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>\tiny</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>\scriptsize</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>\footnotesize</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>\small</td>
<td>9</td>
<td>10</td>
<td>10.95</td>
</tr>
<tr>
<td>\normalsize (default)</td>
<td>10</td>
<td>10.95</td>
<td>12</td>
</tr>
<tr>
<td>\large</td>
<td>12</td>
<td>12</td>
<td>14.4</td>
</tr>
<tr>
<td>\Large</td>
<td>14.4</td>
<td>14.4</td>
<td>17.28</td>
</tr>
<tr>
<td>\LARGE</td>
<td>17.28</td>
<td>17.28</td>
<td>20.74</td>
</tr>
<tr>
<td>\huge</td>
<td>20.74</td>
<td>20.74</td>
<td>24.88</td>
</tr>
<tr>
<td>\Huge</td>
<td>24.88</td>
<td>24.88</td>
<td>24.88</td>
</tr>
</tbody>
</table>

The commands are listed here in declaration (not environment) form, since that is how they are typically used. For example.

\begin{quotation} \small
The Tao that can be named is not the eternal Tao.
\end{quotation}

Here, the scope of the \texttt{\small} lasts until the end of the \texttt{quotation} environment. It would also end at the next type style command or the end of the current group, so you could enclose it in curly braces \{\texttt{\small This text is typeset in the small font.}\}.

Trying to use these commands in math, as with $\texttt{\small mv^-2/2}$, results in ‘\LaTeX{} Font Warning: Command \texttt{\small invalid in math mode}’, and the font size doesn’t change. To work with a too-large formula, often the best option is to use the \texttt{displaymath} environment (see Chapter 16 [Math formulas], page 148), or one of the environments from the \texttt{amsmath} package. For inline mathematics, such as in a table of formulas, an alternative is something like \{\texttt{\small $mv^-2/2$}\}. (Sometimes \texttt{\scriptsize} and \texttt{\scriptstyle} are confused. Both
change the font size, but the latter also changes a number of other aspects of how mathematics is typeset. See Section 16.7 [Math styles], page 174.)

An environment form of each of these commands is also defined; for instance, \begin{tiny}...\end{tiny}. However, in practice this form can easily lead to unwanted spaces at the beginning and/or end of the environment without careful consideration, so it’s generally less error-prone to stick to the declaration form.

(Aside: Technically, due to the way \LaTeX defines \begin and \end, nearly every command that does not take an argument technically has an environment form. But in almost all cases, it would only cause confusion to use it. The reason for mentioning the environment form of the font size declarations specifically is that this particular use is not rare.)

### 4.4 Low-level font commands

These commands are primarily intended for writers of macros and packages. The commands listed here are only a subset of the available ones.

- \fontencoding{encoding}
  Select the font encoding, the encoding of the output font. There are a large number of valid encodings. The most common are OT1, Knuth’s original encoding for Computer Modern (the default), and T1, also known as the Cork encoding, which has support for the accented characters used by the most widespread European languages (German, French, Italian, Polish and others), which allows \TeX{} to hyphenate words containing accented letters. For more, see https://ctan.org/pkg/encguide.

- \fontfamily{family}
  Select the font family. The web page https://tug.org/FontCatalogue/ provides one way to browse through many of the fonts easily used with \LaTeX{}. Here are examples of some common families.

```
pag  Avant Garde
fvs   Bitstream Vera Sans
pbk   Bookman
bch   Charter
ccr   Computer Concrete
cmr   Computer Modern
cmss  Computer Modern Sans Serif
cmtt  Computer Modern Typewriter
pctt  Courier
phv   Helvetica
fi4   Inconsolata
lmr   Latin Modern
lmss  Latin Modern Sans
lmtt  Latin Modern Typewriter
pnc   New Century Schoolbook
ppl   Palatino
ptm   Times
uncl  Uncial
put   Utopia
```
Chapter 4: Fonts

\fontseries{series}
Select the font series. A series combines a weight and a width. Typically, a font supports only a few of the possible combinations. Some common combined series values include:

- m Medium (normal)
- b Bold
- c Condensed
- bc Bold condensed
- bx Bold extended

The possible values for weight, individually, are:

- ul Ultra light
- el Extra light
- l Light
- sl Semi light
- m Medium (normal)
- sb Semi bold
- b Bold
- eb Extra bold
- ub Ultra bold

The possible values for width, individually, are (the meaning and relationship of these terms varies with individual typefaces):

- uc Ultra condensed
- ec Extra condensed
- c Condensed
- sc Semi condensed
- m Medium
- sx Semi expanded
- x Expanded
- ex Extra expanded
- ux Ultra expanded

When forming the series string from the weight and width, drop the m that stands for medium weight or medium width, unless both weight and width are m, in which case use just one (‘m’).

\fontshape{shape}
Select font shape. Valid shapes are:

- n Upright (normal)
- it Italic
- sl Slanted (oblique)
- sc Small caps
- ui Upright italics
- ol Outline

The two last shapes are not available for most font families, and small caps are often missing as well.
\fontsize{\textit{size}}{\textit{skip}}
Set the font size and the line spacing. The unit of both parameters defaults to points (pt). The line spacing is the nominal vertical space between lines, baseline to baseline. It is stored in the parameter \baselineskip. The default \baselineskip for the Computer Modern typeface is 1.2 times the \fontsize. Changing \baselineskip directly is inadvisable since its value is reset every time a size change happens; instead use \baselinestretch. (see Section 5.6 [\baselineskip & \baselinestretch], page 30).

\linespread{factor}
Equivalent to \renewcommand{\baselinestretch}{factor}, and therefore must be followed by \selectfont to have any effect. Best specified in the preamble. See Section 5.6 [\baselineskip & \baselinestretch], page 30, for using \setspace package instead.

\selectfont
The effects of the font commands described above do not happen until \selectfont is called, as in \textit{\fontfamily{familyname}}\selectfont. It is often useful to put this in a macro:
\texttt{\newcommand*[\texttt{myfont}]{\texttt{\fontfamily{familyname}}\texttt{\selectfont}}}
(see Section 12.1 [\texttt{newcommand & \renewcommand}], page 112).

\usefont{enc}{family}{series}{shape}
The same as invoking \fontencoding, \fontfamily, \fontseries and \fontshape with the given parameters, followed by \selectfont. For example:
\usefont{ot1}{cmr}{m}{n}
5 Layout

Commands for controlling the general page layout.

5.1 \onecolumn

Synopsis:

\onecolumn

Start a new page and produce single-column output. If the document is given the class option onecolumn then this is the default behavior (see Section 3.1 [Document class options], page 10). This command is fragile (see Section 12.11 [\protect], page 123).

5.2 \twocolumn

Synopsis:

\twocolumn
\twocolumn[prelim one column text]

Start a new page and produce two-column output. If the document is given the class option twocolumn then this is the default (see Section 3.1 [Document class options], page 10). This command is fragile (see Section 12.11 [\protect], page 123).

If the optional prelim one column text argument is present, it is typeset in one-column mode before the two-column typesetting starts.

These parameters control typesetting in two-column output:

\columnsep

The distance between columns. The default is 35pt. Change it with a command such as \setlength{\columnsep}{40pt}. You must change it before the two column mode starts; in the preamble is a good place.

\columnseprule

The width of the rule between columns. The default is 0pt, meaning that there is no rule. Otherwise, the rule appears halfway between the two columns. Change it with a command such as \setlength{\columnseprule}{0.4pt}; before the two-column mode starts.

\columnwidth

The width of a single column. In one-column mode this is equal to \textwidth. In two-column mode by default \LaTeX sets the width of each of the two columns, \columnwidth, to be half of \textwidth minus \columnsep.

In a two-column document, the starred environments table* and figure* are two columns wide, whereas the unstarred environments table and figure take up only one column (see Section 8.10 [figure], page 60, and see Section 8.22 [table], page 84). \LaTeX places starred floats at the top of a page. The following parameters control float behavior of two-column output.

\dbltopfraction

The maximum fraction at the top of a two-column page that may be occupied by two-column wide floats. The default is 0.7, meaning that the height of a table*
or `figure*` environment must not exceed 0.7\textwidth. If the height of your starred float environment exceeds this then you can take one of the following actions to prevent it from floating all the way to the back of the document:

- Use the [tp] location specifier to tell \LaTeX{} to try to put the bulky float on a page by itself, as well as at the top of a page.
- Use the [t!] location specifier to override the effect of `dbltopfraction` for this particular float.
- Increase the value of `dbltopfraction` to a suitably large number, to avoid going to float pages so soon.

You can redefine it, as with `\renewcommand{dbltopfraction}{0.9}`.

`dblfloatpagefraction`

For a float page of two-column wide floats, this is the minimum fraction that must be occupied by floats, limiting the amount of blank space. \LaTeX{}’s default is 0.5. Change it with `\renewcommand`.

`dblfloatsep`

On a float page of two-column wide floats, this length is the distance between floats, at both the top and bottom of the page. The default is 12pt plus2pt minus2pt for a document set at 10pt or 11pt, and 14pt plus2pt minus4pt for a document set at 12pt.

`dbltextfloatsep`

This length is the distance between a multi-column float at the top or bottom of a page and the main text. The default is 20pt plus2pt minus4pt.

`dbltopnumber`

On a float page of two-column wide floats, this counter gives the maximum number of floats allowed at the top of the page. The \LaTeX{} default is 2.

This example uses `\twocolumn`’s optional argument of to create a title that spans the two-column article:

```latex
\documentclass[twocolumn]{article}
\newcommand{\authormark}[1]{\textsuperscript{#1}}
\begin{document}
\twocolumn[{
\centering
LARGE The Title \[1.5em]
\large Author One\authormark{1},
Author Two\authormark{2},
Author Three\authormark{1} \[1em]
\normalsize
\begin{tabular}{p{.2\textwidth}@{\hspace{2em}}p{.2\textwidth}}
\authormark{1}Department one &\authormark{2}Department two \\
School one &School two
\end{tabular}\[3em] % space below title part
}]

Two column text here.
```

5.3 \flushbottom

Make all pages in the document after this declaration have the same height, by stretching
the vertical space where necessary to fill out the page. This is most often used when making
two-sided documents since the differences in facing pages can be glaring.

If \TeX{} cannot satisfactorily stretch the vertical space in a page then you get a message like
'Underfull \vbox (badness 10000) has occurred while output is active'. If you get
that, one option is to change to \raggedbottom (see Section 5.4 \raggedbottom, page 28).
Alternatively, you can adjust the \textwidth to make compatible pages, or you can add some
vertical stretch glue between lines or between paragraphs, as in \setlength{\parskip}{0ex
plus0.1ex}. Your last option is to, in a final editing stage, adjust the height of individual
pages (see Section 10.3 \enlargethispage, page 105).

The \flushbottom state is the default only if you select the \twocolumn document class
option (see Section 3.1 \twocolumn, page 10), and for indexes made using makeidx.

5.4 \raggedbottom

Make all later pages the natural height of the material on that page; no rubber vertical lengths
will be stretched. Thus, in a two-sided document the facing pages may be different heights.
This command can go at any point in the document body. See Section 5.3 \flushbottom, page 28.

This is the default unless you select the \twocolumn document class option (see Section 3.1
\twocolumn, page 10).

5.5 Page layout parameters

\columnsep
\columnseprule
\columnwidth

The distance between the two columns, the width of a rule between the columns,
and the width of the columns, when the document class option \twocolumn is
in effect (see Section 3.1 \twocolumn, page 10). See Section 5.2
\twocolumn, page 26.

\headheight

Height of the box that contains the running head. The default in the \article, \report, and \book classes is ‘12pt’, at all type sizes.

\headsep
Vertical distance between the bottom of the header line and the top of the main
text. The default in the \article and \report classes is ‘25pt’. In the \book class
the default is: if the document is set at 10pt then it is ‘0.25in’, and at 11pt or
12pt it is ‘0.275in’.

\footskip
Distance from the baseline of the last line of text to the baseline of the page
footer. The default in the \article and \report classes is ‘30pt’. In the \book class
the default is: when the type size is 10pt the default is ‘0.35in’, while at
11pt it is ‘0.38in’, and at 12pt it is ‘30pt’.
\linewidth
Width of the current line, decreased for each nested list (see Section 8.16 [list], page 65). That is, the nominal value for \linewidth is to equal \textwidth but for each nested list the \linewidth is decreased by the sum of that list’s \leftmargin and \rightmargin (see Section 8.14 [itemize], page 63).

\marginparpush
\marginsep
\marginparwidth
The minimum vertical space between two marginal notes, the horizontal space between the text body and the marginal notes, and the horizontal width of the notes.

Normally marginal notes appear on the outside of the page, but the declaration \reversemarginpar changes that (and \normalmarginpar changes it back).

The defaults for \marginparpush in both book and article classes are: ‘7pt’ if the document is set at 12pt, and ‘5pt’ if the document is set at 11pt or 10pt. For \marginsep, in article class the default is ‘10pt’ except if the document is set at 10pt and in two-column mode where the default is ‘11pt’.

For \marginsep in book class the default is ‘10pt’ in two-column mode and ‘7pt’ in one-column mode.

For \marginparwidth in both book and article classes, in two-column mode the default is 60\% of \paperwidth – \textwidth, while in one-column mode it is 50\% of that distance.

\oddsidemargin
\evensidemargin
The \oddsidemargin length is the extra distance between the left side of the page and the text’s left margin, on odd-numbered pages when the document class option twoside is chosen and on all pages when oneside is in effect. When twoside is in effect, on even-numbered pages the extra distance on the left is \evensidemargin.

\LaTeX’s default is that \oddsidemargin is 40\% of the difference between \paperwidth and \textwidth, and \evensidemargin is the remainder.

\paperheight
The height of the paper, as distinct from the height of the print area. Normally set with a document class option, as in \documentclass[a4paper]{article} (see Section 3.1 [Document class options], page 10).

\paperwidth
The width of the paper, as distinct from the width of the print area. Normally set with a document class option, as in \documentclass[a4paper]{article} (see Section 3.1 [Document class options], page 10).

\textheight
The normal vertical height of the page body. If the document is set at a nominal type size of 10pt then for an article or report the default is ‘43\baselineskip’, while for a book it is ‘41\baselineskip’. At a type size of 11pt the default is
‘38\baselineskip’ for all document classes. At 12pt it is ‘36\baselineskip’ for all classes.

\textwidth

The full horizontal width of the entire page body. For an article or report document, the default is ‘345pt’ when the chosen type size is 10pt, the default is ‘360pt’ at 11pt, and it is ‘390pt’ at 12pt. For a book document, the default is ‘4.5in’ at a type size of 10pt, and ‘5in’ at 11pt or 12pt.

In multi-column output, \textwidth remains the width of the entire page body, while \columnwidth is the width of one column (see Section 5.2 [\twocolumn], page 26).

In lists (see Section 8.16 [list], page 65), \textwidth remains the width of the entire page body (and \columnwidth the width of the entire column), while \linewidth may decrease for nested lists.

Inside a minipage (see Section 8.18 [minipage], page 71) or \parbox (see Section 20.3 [\parbox], page 199), all the width-related parameters are set to the specified width, and revert to their normal values at the end of the minipage or \parbox.

\hsize

This entry is included for completeness: \hsize is the \TeX primitive parameter used when text is broken into lines. It should not be used in normal \LaTeX documents.

\topmargin

Space between the top of the \TeX page (one inch from the top of the paper, by default) and the top of the header. The value is computed based on many other parameters: \paperheight − 2in − \headheight − \headsep − \textheight − \footskip, and then divided by two.

\topskip

Minimum distance between the top of the page body and the baseline of the first line of text. For the standard classes, the default is the same as the font size, e.g., ‘10pt’ at a type size of 10pt.

5.6 \baselineskip & \baselinestretch

The \baselineskip is a rubber length (see Chapter 14 [Lengths], page 137). It gives the leading, the normal distance between lines in a paragraph, from baseline to baseline.

Ordinarily document authors do not directly change \baselineskip while writing. Instead, it is set by the low level font selection command \fontsize (see [low level font commands fontsize], page 24). The \baselineskip’s value is reset every time a font change happens and so any direct change to \baselineskip would vanish the next time there was a font switch. For how to influence line spacing, see the discussion of \baselinestretch below.

Usually, a font’s size and baseline skip is assigned by the font designer. These numbers are nominal in the sense that if, for instance, a font’s style file has the command \fontsize{10pt}{12pt} then that does not mean that the characters in the font are 10 pt tall; for instance, parentheses and accented capitals may be taller. Nor does it mean that if the lines are spaced less than 12 pt apart then they risk touching. Rather these numbers are
typographic judgements. (Often, the \baselineskip is about twenty percent larger than the font size.)

The \baselineskip is not a property of each line but of the entire paragraph. As a result, large text in the middle of a paragraph, such as a single \Huge Q, will be squashed into its line. \TeX will make sure it doesn’t scrape up against the line above but won’t change the \baselineskip for that one line to make extra room above. For the fix, use a \strut (see Section 19.13 [strut], page 193).

The value of \baselineskip that \TeX uses for the paragraph is the value in effect at the blank line or command that ends the paragraph unit. So if a document contains this paragraph then its lines will be scrunched together, compared to lines in surrounding paragraphs.

Many people see a page break between text and a displayed equation as bad style, so in effect the display is part of the paragraph. Because this display is in footnotesize, the entire paragraph has the baseline spacing matching that size.

\{\footnotesize $$a+b = c$$\}

The process for making paragraphs is that when a new line is added, if the depth of the previous line plus the height of the new line is less than \baselineskip then \TeX inserts vertical glue to make up the difference. There are two fine points. The first is that if the lines would be too close together, closer than \lineskiplimit, then \TeX instead uses \lineskip as the interline glue. The second is that \TeX doesn’t actually use the depth of the previous line. Instead it uses \prevdepth, which usually contains that depth. But at the beginning of the paragraph (or any vertical list) or just after a rule, \prevdepth has the value -1000 pt and this special value tells \TeX not to insert any interline glue at the paragraph start.

In the standard classes \lineskiplimit is 0 pt and \lineskip is 1 pt. By the prior paragraph then, the distance between lines can approach zero but if it becomes zero (or less than zero) then the lines jump to 1 pt apart.

Sometimes authors must, for editing purposes, put the document in double space or one-and-a-half space. The right way to influence the interline distance is via \baselinestretch. It scales \baselineskip, and has a default value of 1.0. It is a command, not a length, and does not take effect until a font change happens, so set the scale factor like this:

\renewcommand{\baselinestretch}{1.5}\selectfont.

The most straightforward way to change the line spacing for an entire document is to put \linespread{factor} in the preamble. For double spacing, take factor to be 1.6 and for one-and-a-half spacing use 1.3. These numbers are rough: for instance, since the \baselineskip is about 1.2 times the font size, multiplying by 1.6 gives a baseline skip to font size ratio of about 2. (The \linespread command is defined as \renewcommand{\baselinestretch}{factor} so it also won’t take effect until a font setting happens. But that always takes place at the start of a document, so there you don’t need to follow it with \selectfont.)

A simpler approach is the \setspace package. The basic example:

\usepackage{setspace}
\doublespacing % or \onehalfspacing for 1.5

In the preamble these will start the document off with that sizing. But you can also use these declarations in the document body to change the spacing from that point.
forward, and consequently there is \singlespacing to return the spacing to normal. In the document body, a better practice than using the declarations is to use environments, such as \begin{doublespace} ... \end{doublespace}. The package also has commands to do arbitrary spacing: \setstretch{factor} and \begin{spacing}{factor} ... \end{spacing}. This package also keeps the line spacing single-spaced in places where that is typically desirable, such as footnotes and figure captions. See the package documentation.

5.7 Floats

Some typographic elements, such as figures and tables, cannot be broken across pages. They must be typeset outside of the normal flow of text, for instance floating to the top of a later page.

\LaTeX can have a number of different classes of floating material. The default is the two classes, \texttt{figure} (see Section 8.10 \texttt{[figure]}, page 60) and \texttt{table} (see Section 8.22 \texttt{[table]}, page 84), but you can create a new class with the package \texttt{float}.

Within any one float class \LaTeX always respects the order, so that the first figure in a document source must be typeset before the second figure. However, \LaTeX may mix the classes, so it can happen that while the first table appears in the source before the first figure, it appears in the output after it.

The placement of floats is subject to parameters, given below, that limit the number of floats that can appear at the top of a page, and the bottom, etc. If so many floats are queued that the limits prevent them all from fitting on a page then \LaTeX places what it can and defers the rest to the next page. In this way, floats may end up being typeset far from their place in the source. In particular, a float that is big may migrate to the end of the document. In which event, because all floats in a class must appear in sequential order, every following float in that class also appears at the end.

In addition to changing the parameters, for each float you can tweak where the float placement algorithm tries to place it by using its \texttt{placement} argument. The possible values are a sequence of the letters below. The default for both \texttt{figure} and \texttt{table}, in both \texttt{article} and \texttt{book} classes, is \texttt{tbp}.

- \texttt{t} (Top)—at the top of a text page.
- \texttt{b} (Bottom)—at the bottom of a text page. (However, \texttt{b} is not allowed for full-width floats (\texttt{figure*}) with double-column output. To ameliorate this, use the \texttt{stfloats} or \texttt{dblfloatfix} package, but see the discussion at caveats in the FAQ: \url{https://www.texfaq.org/FAQ-2colfloat}.
- \texttt{h} (Here)—at the position in the text where the \texttt{figure} environment appears. However, \texttt{h} is not allowed by itself; \texttt{t} is automatically added. To absolutely force a float to appear “here”, you can \texttt{\usepackage{float}} and use the \texttt{H} specifier which it defines. For further discussion, see the FAQ entry at \url{https://www.texfaq.org/FAQ-figurehere}.
- \texttt{p} (Page of floats)—on a separate float page, which is a page containing no text, only floats.
- \texttt{!} Used in addition to one of the above; for this float only, \LaTeX ignores the restrictions on both the number of floats that can appear and the relative
amounts of float and non-float text on the page. The ! specifier does not mean “put the float here”; see above.

Note: the order in which letters appear in the placement argument does not change the order in which \LaTeX{} tries to place the float; for instance, \texttt{btp} has the same effect as \texttt{tbp}. All that placement does is that if a letter is not present then the algorithm does not try that location. Thus, \LaTeX{}'s default of \texttt{tbp} is to try every location except placing the float where it occurs in the source.

To prevent \LaTeX{} from moving floats to the end of the document or a chapter you can use a \texttt{\clearpage} command to start a new page and insert all pending floats. If a pagebreak is undesirable then you can use the afterpage package and issue \texttt{\afterpage{\clearpage}}. This will wait until the current page is finished and then flush all outstanding floats.

\LaTeX{} can typeset a float before where it appears in the source (although on the same output page) if there is a \texttt{t} specifier in the placement parameter. If this is not desired, and deleting the \texttt{t} is not acceptable as it keeps the float from being placed at the top of the next page, then you can prevent it by either using the \texttt{flafter} package or using the command \texttt{\suppressfloats[t]}, which causes floats for the top position on this page to moved to the next page.

Parameters relating to fractions of pages occupied by float and non-floating text (change them with \texttt{\renewcommand{\textit{parameter}}{\textit{decimal between 0 and 1}}}):

\texttt{\bottomfraction}

The maximum fraction of the page allowed to be occupied by floats at the bottom; default ‘.3’.

\texttt{\floatpagefraction}

The minimum fraction of a float page that must be occupied by floats; default ‘.5’.

\texttt{\textfraction}

Minimum fraction of a page that must be text; if floats take up too much space to preserve this much text, floats will be moved to a different page. The default is ‘.2’.

\texttt{\topfraction}

Maximum fraction at the top of a page that may be occupied before floats; default ‘.7’.

Parameters relating to vertical space around floats (change them with a command of the form \texttt{\setlength{\textit{parameter}}{\textit{length expression}}}):

\texttt{\floatsep}

Space between floats at the top or bottom of a page; default ‘12pt plus2pt minus2pt’.

\texttt{\intextsep}

Space above and below a float in the middle of the main text; default ‘12pt plus2pt minus2pt’ for 10 point and 11 point documents, and ‘14pt plus4pt minus4pt’ for 12 point documents.
\textfloatsep
Space between the last (first) float at the top (bottom) of a page; default ‘20pt plus2pt minus4pt’.

Counters relating to the number of floats on a page (change them with a command of the form \setcounter{ctrname}{natural number}):

bottomnumber
Maximum number of floats that can appear at the bottom of a text page; default 1.

dbltopnumber
Maximum number of full-sized floats that can appear at the top of a two-column page; default 2.

topnumber
Maximum number of floats that can appear at the top of a text page; default 2.

totalnumber
Maximum number of floats that can appear on a text page; default 3.

The principal \TeX FAQ entry relating to floats \url{https://www.texfaq.org/FAQ-floats} contains suggestions for relaxing \LaTeX’s default parameters to reduce the problem of floats being pushed to the end. A full explanation of the float placement algorithm is in Frank Mittelbach’s article “How to influence the position of float environments like figure and table in \LaTeX?” (\url{https://www.latex-project.org/publications/2014-FMi-TUB-tb111mitt-float-placement.pdf}).

5.7.1 \caption
Synopsis:

\caption{caption-text}

or

\caption[short-caption-text]{caption-text}

Make a caption for a floating environment, such as a figure or table environment (see Section 8.10 [figure], page 60, or Section 8.22 [table], page 84).

In this example, \LaTeX places a caption below the vertical blank space that is left by the author for the later inclusion of a picture.

\begin{figure}
  \vspace*{1cm}
  \caption{Alonzo Cushing, Battery A, 4th US Artillery.}
  \label{fig:CushingPic}
\end{figure}

The \caption command will label the caption-text with something like ‘Figure 1:’ for an article or ‘Figure 1.1:’ for a book. The text is centered if it is shorter than the text width, or set as an unindented paragraph if it takes more than one line.

In addition to placing the caption-text in the output, the \caption command also saves that information for use in a list of figures or list of tables (see Section 25.1 [Table of contents etc.], page 233).
Here the \caption command uses the optional \textit{short-caption-text}, so that the shorter text appears in the list of tables, rather than the longer \textit{caption-text}.

\begin{table}
\centering
\begin{tabular}{|*{3}{c}|}
\hline
4 & 9 & 2 \\ 
3 & 5 & 7 \\ 
8 & 1 & 6 \\ 
\hline
\end{tabular}
\caption[\textit{Lo Shu} magic square]{The \textit{Lo Shu} magic square, which is unique among squares of order three up to rotation and reflection.}
\label{tab:LoShu}
\end{table}

\LaTeX will label the \textit{caption-text} with something like ‘Table 1:’ for an article or ‘Table 1.1:’ for a book.

The caption can appear at the top of the \textit{figure} or \textit{table}. For instance, that would happen in the prior example by putting the \caption between the \centering and the \begin{tabular}.

Different floating environments are numbered separately, by default. It is \caption that updates the counter, and so any \label must come after the \caption. The counter for the \textit{figure} environment is named \texttt{figure}, and similarly the counter for the \textit{table} environment is \texttt{table}.

The text that will be put in the list of figures or list of tables is moving argument. If you get the \LaTeX error ‘! \texttt{Argument of @caption has an extra }’ then you must put \texttt{protect} in front of any fragile commands. See Section 12.11 [\protect], page 123.

The \caption package has many options to adjust how the caption appears, for example changing the font size, making the caption be hanging text rather than set as a paragraph, or making the caption always set as a paragraph rather than centered when it is short.
6 Sectioning

Structure your text into divisions: parts, chapters, sections, etc. All sectioning commands have the same form, one of:

\textit{sectioning-command}\{\textit{title}\}
\textit{sectioning-command\*}\{\textit{title}\}
\textit{sectioning-command}\{toc-title\}\{\textit{title}\}

For instance, declare the start of a subsection as with \subsection{Motivation}.

The table has each sectioning-command in \LaTeX. All are available in all of \LaTeX’s standard document classes \texttt{book}, \texttt{report}, and \texttt{article}, except that \texttt{chapter} is not available in \texttt{article}.

<table>
<thead>
<tr>
<th>Sectioning unit</th>
<th>Command</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>\part</td>
<td>-1 (book, report), 0 (article)</td>
</tr>
<tr>
<td>Chapter</td>
<td>\chapter</td>
<td>0</td>
</tr>
<tr>
<td>Section</td>
<td>\section</td>
<td>1</td>
</tr>
<tr>
<td>Subsection</td>
<td>\subsection</td>
<td>2</td>
</tr>
<tr>
<td>Subsubsection</td>
<td>\subsubsection</td>
<td>3</td>
</tr>
<tr>
<td>Paragraph</td>
<td>\paragraph</td>
<td>4</td>
</tr>
<tr>
<td>Subparagraph</td>
<td>\subparagraph</td>
<td>5</td>
</tr>
</tbody>
</table>

All these commands have a *-form that prints \textit{title} as usual but does not number it and does not make an entry in the table of contents. An example of using this is for an appendix in an \texttt{article}. The input \texttt{\appendix\section{Appendix}} gives the output ‘A Appendix’ (see Section 6.6 [appendix], page 42). You can lose the numbering ‘A’ by instead entering \texttt{\section*{Appendix}} (articles often omit a table of contents and have simple page headers so the other differences from the \texttt{\section} command may not matter).

The section title \textit{title} provides the heading in the main text, but it may also appear in the table of contents and in the running head or foot (see Chapter 18 [Page styles], page 179). You may not want the same text in these places as in the main text. All of these commands have an optional argument \textit{toc-title} for these other places.

The level number in the table above determines which sectional units are numbered, and which appear in the table of contents. If the sectioning command’s level is less than or equal to the value of the counter \texttt{secnumdepth} then the titles for this sectioning command will be numbered (see [Sectioning/secnumdepth], page 37). And, if level is less than or equal to the value of the counter \texttt{tocdepth} then the table of contents will have an entry for this sectioning unit (see [Sectioning/tocdepth], page 37).

\LaTeX expects that before you have a \texttt{\subsection} you will have a \texttt{\section} and, in a \texttt{book} class document, that before a \texttt{\section} you will have a \texttt{\chapter}. Otherwise you can get something like a subsection numbered ‘3.0.1’.

\LaTeX lets you change the appearance of the sectional units. As a simple example, you can change the section numbering to uppercase letters with this (in the preamble):
\texttt{\renewcommand\thesection{\Alph{section}}} . (See Section 13.1 [alph \Alph \arabic \roman \Roman \fnsymbol] , page 133.) CTAN has many packages that make this adjustment easier, notably \texttt{titlesec}.
Two counters relate to the appearance of headings made by sectioning commands.

**secnumdepth**

Controls which sectioning unit are numbered. Setting the counter with \setcounter{secnumdepth}{level} will suppress numbering of sectioning at any depth greater than level (see Section 13.4 [\setcounter], page 135). See the above table for the level numbers. For instance, if the secnumdepth is 1 in an article then a section{Introduction} command will produce output like ‘1 Introduction’ while subsection{Discussion} will produce output like ‘Discussion’, without the number. \LaTeX’s default secnumdepth is 3 in article class and 2 in the book and report classes.

**tocdepth**

Controls which sectioning units are listed in the table of contents. The setting \setcounter{tocdepth}{level} makes the sectioning units at level be the smallest ones listed (see Section 13.4 [\setcounter], page 135). See the above table for the level numbers. For instance, if tocdepth is 1 then the table of contents will list sections but not subsections. \LaTeX’s default tocdepth is 3 in article class and 2 in the book and report classes.

### 6.1 \part

Synopsis, one of:

\part{title}  
\part*[title]  
\part[toc-title]{title}

Start a document part. The standard \LaTeX classes book, report, and article, all have this command.

This produces a document part, in a book.

\part{VOLUME I \  
PERSONAL MEMOIRS OF U. S. GRANT}  
\chapter{ANCESTRY--BIRTH--BOYHOOD.}  
My family is American, and has been for generations,  
in all its branches, direct and collateral.

In each standard class the \part command outputs a part number such as ‘Part I’, alone on its line, in boldface, and in large type. Then \LaTeX outputs title, also alone on its line, in bold and in even larger type. In class book, the \LaTeX default puts each part alone on its own page. If the book is two-sided then \LaTeX will skip a page if needed to have the new part on an odd-numbered page. In report it is again alone on a page, but \LaTeX won’t force it onto an odd-numbered page. In an article \LaTeX does not put it on a fresh page, but instead outputs the part number and part title onto the main document page.

The * form shows title but it does not show the part number, does not increment the part counter, and produces no table of contents entry.

The optional argument toc-title will appear as the part title in the table of contents (see Section 25.1 [Table of contents etc.], page 233) and in running headers (see Chapter 18 [Page styles], page 179). If it is not present then title will be there. This example puts a line break in title but omits the break in the table of contents.

\part[Up from the bottom; my life]{Up from the bottom\ my life}
For determining which sectional units are numbered and which appear in the table of contents, the level number of a part is -1 (see [Sectioning/secnumdepth], page 37, and [Sectioning/tocdepth], page 37).

In the class article, if a paragraph immediately follows the part title then it is not indented. To get an indent you can use the package \texttt{indentfirst}.

One package to change the behavior of \texttt{\part} is \texttt{titlesec}. See its documentation on CTAN.

\section*{6.2 \texttt{\chapter}}

Synopsis, one of:
\begin{itemize}
  \item \texttt{\chapter\{title\}}
  \item \texttt{\chapter*\{title\}}
  \item \texttt{\chapter[toc-title]\{title\}}
\end{itemize}

Start a chapter. The standard \LaTeX\ classes \texttt{book} and \texttt{report} have this command but \texttt{article} does not.

This produces a chapter.
\begin{itemize}
  \item \texttt{\chapter\{Loomings\}}
  \item Call me Ishmael.
  \item Some years ago---never mind how long precisely---having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world.
\end{itemize}

The \LaTeX\ default starts each chapter on a fresh page, an odd-numbered page if the document is two-sided. It produces a chapter number such as ‘\texttt{Chapter 1}’ in large boldface type (the size is \texttt{\Huge}). It then puts \texttt{title} on a fresh line, in boldface type that is still larger (size \texttt{\Huge}). It also increments the \texttt{chapter} counter, adds an entry to the table of contents (see Section 25.1 \texttt{[Table of contents etc.], page 233}), and sets the running header information (see Chapter 18 \texttt{[Page styles], page 179}).

The \texttt{*} form shows \texttt{title} on a fresh line, in boldface. But it does not show the chapter number, does not increment the \texttt{chapter} counter, produces no table of contents entry, and does not affect the running header. (If you use the page style \texttt{headings} in a two-sided document then the header will be from the prior chapter.) This example illustrates.
\begin{itemize}
  \item \texttt{\chapter\{Preamble\}}
\end{itemize}

The optional argument \texttt{toc-title} will appear as the chapter title in the table of contents (see Section 25.1 \texttt{[Table of contents etc.], page 233}) and in running headers (see Chapter 18 \texttt{[Page styles], page 179}). If it is not present then \texttt{title} will be there. This shows the full name in the chapter title,
\begin{itemize}
  \item \texttt{\chapter\{Hermann Klaus Hugo (Peter) Weyl (1885--1955)\}}
\end{itemize}
but only ‘\texttt{Weyl}’ on the contents page. This puts a line break in the title but that doesn’t work well with running headers so it omits the break in the contents
\begin{itemize}
  \item \texttt{\chapter\{Given it all; my story\}}\texttt{\{Given it all\ \my\story\}}
\end{itemize}

For determining which sectional units are numbered and which appear in the table of contents, the level number of a chapter is 0 (see [Sectioning/secnumdepth], page 37, and see [Sectioning/tocdepth], page 37).
The paragraph that follows the chapter title is not indented, as is a standard typographical practice. To get an indent use the package `indentfirst`.

You can change what is shown for the chapter number. To change it to something like ‘Lecture 1’, put in the preamble either `\renewcommand{\chaptername}{Lecture}` or this (see Section 12.3 \[\texttt{\textbackslash makeatletter \textbackslash makeatother}\], page 115).

\makeatletter
\renewcommand{\@chapapp}{Lecture}
\makeatother

To make this change because of the primary language for the document, see the package `babel`.

In a two-sided document \LaTeX puts a chapter on odd-numbered page, if necessary leaving an even-numbered page that is blank except for any running headers. To make that page completely blank, see Section 10.1 \[\texttt{\textbackslash clearpage \& \textbackslash cleardoublepage}\], page 104.

To change the behavior of the `\chapter` command, you can copy its definition from the \LaTeX format file and make adjustments. But there are also many packages on CTAN that address this. One is `titlesec`. See its documentation, but the example below gives a sense of what it can do.

\usepackage{titlesec} % in preamble
\titleformat{\chapter}{\Huge\bfseries} % format of title
{} % label, such as 1.2 for a subsection
{} % length of separation between label and title
{} % before-code hook

This omits the chapter number ‘Chapter 1’ from the page but unlike `\chapter*` it keeps the chapter in the table of contents and the running headers.

6.3 \section

Synopsis, one of:

\section{title}
\section*{title}
\section[\texttt{toc-title}]{title}

Start a section. The standard \LaTeX classes `article`, `book`, and `report` all have this command.

This produces a section.

In this Part we tend to be more interested in the function, in the input-output behavior, than in the details of implementing that behavior.

\section{Turing machines}

Despite this desire to downplay implementation, we follow the approach of A”Turing that the first step toward defining the set of computable functions is to reflect on the details of what mechanisms can do.
For the standard \LaTeX classes \texttt{book} and \texttt{report} the default output is like ‘1.2 \textit{title}’ (for chapter 1, section 2), alone on its line and flush left, in boldface and a larger type (the type size is \texttt{\Large}). The same holds in \texttt{article} except that there are no chapters in that class so it looks like ‘2 \textit{title}’.

The * form shows title. But it does not show the section number, does not increment the \texttt{section} counter, produces no table of contents entry, and does not affect the running header. (If you use the page style \texttt{headings} in a two-sided document then the header will be from the prior section.)

The optional argument \texttt{toc-title} will appear as the section title in the table of contents (see Section 25.1 [Table of contents etc.], page 233) and in running headers (see Chapter 18 [Page styles], page 179). If it is not present then \textit{title} will be there. This shows the full name in the title of the section:

\begin{verbatim}
\section[Elizabeth~II]{Elizabeth the Second, by the Grace of God of the United Kingdom, Canada and Her other Realms and Territories Queen, Head of the Commonwealth, Defender of the Faith.}
\end{verbatim}

but only ‘Elizabeth II’ on the contents page and in the headers. This has a line break in title but that does not work with headers so it is omitted from the contents and headers.

\begin{verbatim}
\section[Truth is, I cheated; my life story]{Truth is, I cheated\my life story}
\end{verbatim}

For determining which sectional units are numbered and which appear in the table of contents, the level number of a section is 1 (see [Sectioning/secnumdepth], page 37, and see [Sectioning/tocdepth], page 37).

The paragraph that follows the section title is not indented, as is a standard typographical practice. One way to get an indent is to use the package \texttt{indentfirst}.

In general, to change the behavior of the \texttt{section} command, there are a number of options. One is the \texttt{\@startsection} command (see Section 6.8 \texttt{\@startsection}, page 43). There are also many packages on CTAN that address this, including \texttt{titlesec}. See the documentation but the example below gives a sense of what they can do.

\begin{verbatim}
\usepackage{titlesec} % in preamble
\titleformat{\section}{\normalfont\Large\bfseries}{\makebox[1pc][r]{\thesection\hspace{1pc}}}{0pt}{} % label
\titlespacing*{\section}{-1pc}{18pt}{10pt}[10pc]
\end{verbatim}

That puts the section number in the margin.

## 6.4 \texttt{subsection}

Synopsis, one of:

\begin{verbatim}
\subsection{title}
\subsection*{title}
\subsection[\texttt{toc-title}]{title}
\end{verbatim}
Start a subsection. The standard \LaTeX\ classes \texttt{article}, \texttt{book}, and \texttt{report} all have this command.

This produces a subsection.

\texttt{We will show that there are more functions than Turing machines and that therefore some functions have no associated machine.}

\texttt{\subsection{Cardinality}} \texttt{We will begin with two paradoxes that dramatize the challenge to our intuition posed by comparing the sizes of infinite sets.}

For the standard \LaTeX\ classes \texttt{book} and \texttt{report} the default output is like ‘1.2.3 title’ (for chapter 1, section 2, subsection 3), alone on its line and flush left, in boldface and a larger type (the type size is \texttt{large}). The same holds in \texttt{article} except that there are no chapters in that class so it looks like ‘2.3 title’.

The \texttt{*} form shows \texttt{title}. But it does not show the subsection number, does not increment the subsection counter, and produces no table of contents entry.

The optional argument \texttt{toc-title} will appear as the subsection title in the table of contents (see Section 25.1 \cite[Table of contents etc.]{Sectioning}, page 233). If it is not present then \texttt{title} will be there. This shows the full text in the title of the subsection:

\texttt{\subsection\[$\alpha,\beta,\gamma$ paper\]{\textit{The Origin of Chemical Elements} by R.A."Alpher, H."Bethe, and G."Gamow}}

But only ‘$\alpha,\beta,\gamma$ paper’ on the contents page.

For determining which sectional units are numbered and which appear in the table of contents, the level number of a subsection is 2 (see \cite[Sectioning/secnumdepth]{Sectioning}, page 37, and see \cite[Sectioning/tocdepth]{Sectioning}, page 37).

The paragraph that follows the subsection title is not indented, as is a standard typographical practice. One way to get an indent is to use the package \texttt{indentfirst}.

There are a number of ways to change the behavior of the \texttt{\subsection} command. One is the \texttt{@startsection} command (see Section 6.8 \cite[@startsection]{Sectioning}, page 43). There are also many packages on CTAN that address this, including \texttt{titlesec}. See the documentation but the example below gives a sense of what they can do.

\texttt{\usepackage{titlesec} % in preamble}
\texttt{\titleformat{\subsection}[runin]}
\texttt{\normalfont\normalsize\bfseries} % format of the title
\texttt{\thesubsection} % label
\texttt{0.6em} % space between label and title
\texttt{\{} % before-code hook

That puts the subsection number and \texttt{title} in the first line of text.

\section*{6.5 \texttt{\subsubsection}, \texttt{\paragraph}, \texttt{\subparagraph}}

Synopsis, one of:

\texttt{\subsubsection\{title\}}
\texttt{\subsubsection*\{title\}}
\texttt{\subsubsection\{toc-title\}\{title\}}
or one of:
\paragraph{title}
\paragraph*{title}
\paragraph[&title]{title}

or one of:
\subparagraph{title}
\subparagraph*{title}
\subparagraph[&title]{title}

Start a subsubsection, paragraph, or subparagraph. The standard \LaTeX classes article, book, and report all have these commands, although they are not commonly used.

This produces a subsubsection.
\subsubsection{Piston ring compressors: structural performance}

Provide exterior/interior wall cladding assemblies capable of withstanding the effects of load and stresses from consumer-grade gasoline engine piston rings.

The default output of each of the three does not change over the standard \LaTeX classes article, book, and report. For \subsubsection the title is alone on its line, in boldface and normal size type. For \paragraph the title is inline with the text, not indented, in boldface and normal size type. For \subparagraph the title is inline with the text, with a paragraph indent, in boldface and normal size type (Because an article has no chapters its subsubsections are numbered and so it looks like ‘1.2.3 title’, for section 1, subsection 2, and subsubsection 3. The other two divisions are not numbered.)

The * form shows title. But it does not increment the associated counter and produces no table of contents entry (and does not show the number for \subsubsection).

The optional argument toc-title will appear as the division title in the table of contents (see Section 25.1 [Table of contents etc.], page 233). If it is not present then title will be there.

For determining which sectional units are numbered and which appear in the table of contents, the level number of a subsubsection is 3, of a paragraph is 4, and of a subparagraph is 5 (see [Sectioning/secnumdepth], page 37, and see [Sectioning/tocdepth], page 37).

The paragraph that follows the subsubsection title is not indented, as is a standard typographical practice. One way to get an indent is to use the package indentfirst.

There are a number of ways to change the behavior of the these commands. One is the \@startsection command (see Section 6.8 [\@startsection], page 43). There are also many packages on CTAN that address this, including titlesec. See the documentation on CTAN.

6.6 \appendix

Synopsis:
\appendix

This does not directly produce any output. But in a book or report document it declares that subsequent \chapter commands start an appendix. In an article it does the same, for \section commands. It also resets the chapter and section counters to 0 in a book or report, and in an article resets the section and subsection counters.
In this book

\chapter{One} ...
\chapter{Two} ...
...
\appendix
\chapter{Three} ...
\chapter{Four} ...

the first two will generate output numbered ‘Chapter 1’ and ‘Chapter 2’. After the \appendix the numbering will be ‘Appendix A’ and ‘Appendix B’. See Section A.4 [Larger book template], page 265, for another example.

The appendix package adds the command \appendixpage to put a separate ‘Appendices’ in the document body before the first appendix, and the command \addappheadtotoc to do the same in the table of contents. You can reset the name ‘Appendices’ with a command like \renewcommand{\appendixname}{Specification}, as well as a number of other features. See the documentation on CTAN.

### 6.7 \frontmatter, \mainmatter, \backmatter

Synopsis, one or more of:

\frontmatter
...
\mainmatter
...
\backmatter
...

Format a book class document differently according to which part of the document is being produced. All three commands are optional.

Traditionally, a book’s front matter contains such things as the title page, an abstract, a table of contents, a preface, a list of notations, a list of figures, and a list of tables. (Some of these front matter pages, such as the title page, are traditionally not numbered.) The back matter may contain such things as a glossary, notes, a bibliography, and an index.

The \frontmatter command makes the pages numbered in lowercase roman, and makes chapters not numbered, although each chapter’s title appears in the table of contents; if you use other sectioning commands here, use the \* -version (see Chapter 6 [Sectioning], page 36).

The \mainmatter command changes the behavior back to the expected version, and resets the page number.

The \backmatter command leaves the page numbering alone but switches the chapters back to being not numbered.

See Section A.4 [Larger book template], page 265, for an example using these three commands.

### 6.8 \@startsection: Typesetting sectional unit headings

Synopsis:

\@startsection{name}{level}{indent}{beforeskip}{afterskip}{style}
Used to help redefine the behavior of commands that start sectioning divisions such as `\section` or `\subsection`.

The `titlesec` package makes manipulation of sectioning easier. Further, while most requirements for sectioning commands can be satisfied with `@startsection`, some cannot. For instance, in the standard \LaTeX book and report classes the commands `\chapter` and `\report` are not constructed using this. To make such a command you may want to use the `secdef` command.

The `@startsection` macro is used like this:

```latex
@startsection{name}{level}{indent}{beforeskip}{afterskip}{style}*\[toctitle\] {title}
```

so that issuing

```latex
\renewcommand{\section}{@startsection{name}{level}{indent}{beforeskip}{afterskip}{style}}
```

redefines `\section` while keeping its standard calling form `\section*{title}` (in which, as a reminder, the star * is optional). See Chapter 6 [Sectioning], page 36. This implies that when you write a command like `\renewcommand{\section}{...}`, the `@startsection{...}` must come last in the definition. See the examples below.

**name**  
Name of the counter used to number the sectioning header. This counter must be defined separately. Most commonly this is either `section`, `subsection`, or `paragraph`. Although in those cases the counter name is the same as the sectioning command itself, you don’t have to use the same name.

Then `\thename` displays the title number and `\namemarker` is for the page headers. See the third example below.

**level**  
An integer giving the depth of the sectioning command. See Chapter 6 [Sectioning], page 36, for the list of standard level numbers.

If `level` is less than or equal to the value of the counter `secnumdepth` then titles for this sectioning command will be numbered (see [Sectioning/secnumdepth], page 37). For instance, if `secnumdepth` is 1 in an `article` then the command `\section{Introduction}` will produce output like “1 Introduction” while `\subsection{Discussion}` will produce output like “Discussion”, without the number prefix.

If `level` is less than or equal to the value of the counter `tocdepth` then the table of contents will have an entry for this sectioning unit (see [Sectioning/tocdepth], page 37). For instance, in an `article`, if `tocdepth` is 1 then the table of contents will list sections but not subsections.
**indent**  
A length giving the indentation of all of the title lines with respect to the left margin. To have the title flush with the margin use 0pt. A negative indentation such as \parindent will move the title into the left margin.

**beforeskip**  
The absolute value of this length is the amount of vertical space that is inserted before this sectioning unit’s title. This space will be discarded if the sectioning unit happens to start at the beginning of a page. If this number is negative then the first paragraph following the header is not indented; if it is non-negative then the first paragraph is indented. (Example: the negative of 1pt plus 2pt minus 3pt is -1pt plus -2pt minus -3pt.)

For example, if beforeskip is $-3.5ex \text{ plus } -1ex \text{ minus } -0.2ex$ then to start the new sectioning unit, \LaTeX will add about 3.5 times the height of a letter x in vertical space, and the first paragraph in the section will not be indented. Using a rubber length, with plus and minus, is good practice here since it gives \LaTeX more flexibility in making up the page (see Chapter 14 \[Lengths\], page 137).

The full accounting of the vertical space between the baseline of the line prior to this sectioning unit’s header and the baseline of the header is that it is the sum of the \parskip of the text font, the \baselineskip of the title font, and the absolute value of the beforeskip. This space is typically rubber so it may stretch or shrink. (If the sectioning unit starts on a fresh page so that the vertical space is discarded then the baseline of the header text will be where \LaTeX would put the baseline of the first text line on that page.)

**afterskip**  
This is a length. If afterskip is non-negative then this is the vertical space inserted after the sectioning unit’s title header. If it is negative then the title header becomes a run-in header, so that it becomes part of the next paragraph. In this case the absolute value of the length gives the horizontal space between the end of the title and the beginning of the following paragraph. (Note that the negative of 1pt plus 2pt minus 3pt is -1pt plus -2pt minus -3pt.)

As with beforeskip, using a rubber length, with plus and minus components, is good practice here since it gives \LaTeX more flexibility in putting together the page.

If afterskip is non-negative then the full accounting of the vertical space between the baseline of the sectioning unit’s header and the baseline of the first line of the following paragraph is that it is the sum of the \parskip of the title font, the \baselineskip of the text font, and the value of after. That space is typically rubber so it may stretch or shrink. (Note that because the sign of afterskip changes the sectioning unit header’s from standalone to run-in, you cannot use a negative afterskip to cancel part of the \parskip.)

**style**  
Controls the styling of the title. See the examples below. Typical commands to use here are \centering, \raggedright, \normalfont, \hrule, or \newpage. The last command in style may be one that takes one argument, such as \MakeUppercase or \fbox that takes one argument. The section title will be supplied as the argument to this command. For instance, setting style to \textbf{\MakeUppercase} would produce titles that are bold and uppercase.

These are \LaTeX’s defaults for the first three sectioning units that are defined with \@startsection, for the article, book, and report classes.
Chapter 6: Sectioning

• For section: level is 1, indent is 0 pt, beforeskip is \-3.5ex plus \-1ex minus \-0.2ex, afterskip is 2.3ex plus 0.2ex, and style is \normalfont\Large\bfseries.

• For subsection: level is 2, indent is 0 pt, beforeskip is \-3.25ex plus \-1ex minus \-0.2ex, afterskip is 1.5ex plus 0.2ex, and style is \normalfont\large\bfseries.

• For subsubsection: level is 3, indent is 0 pt, beforeskip is \-3.25ex plus \-1ex minus \-0.2ex, afterskip is 1.5ex plus 0.2ex, and style is \normalfont\normalsize\bfseries.

Some examples follow. These go either in a package or class file or in the preamble of a \LaTeX document. If you put them in the preamble they must go between a \makeatletter command and a \makeatother. (Probably the error message You can't use \textbackslash spacefactor' in vertical mode. means that you forgot this.) See Section 12.3 [\makeatletter & \makeatother], page 115.

This will put section titles in large boldface type, centered. It says \renewcommand because \LaTeX's standard classes have already defined a \section. For the same reason it does not define a section counter, or the commands \thesection and \l@section.

\renewcommand\section{%
 \@startsection{section}{1}{\centering

 This will put subsection titles in small caps type, inline with the paragraph.

\renewcommand\subsection{%
 \@startsection{subsection}{2}{\scshape

 The prior examples redefined existing sectional unit title commands. This defines a new one, illustrating the needed counter and macros to display that counter.

\setcounter{secnumdepth}{6} show counters this far down
\newcounter{subsubparagraph} counter for numbering
\renewcommand{\thesubsubparagraph} how to display
\newcommand{\thesubparagraph} numbering
\newcommand{\subsubparagraph} {\@startsection
 {subsubparagraph}{6} \{0em\}
 {\baselineskip} \{0.5\baselineskip\}
 {\normalfont\normalsize}}
\newcommand*\l@subsubparagraph{\@dottedtocline{6}{10em}{5em}}% for toc
\newcommand{\subsubparagraphmark}[1]{% for page headers
7 Cross references

We often want something like ‘See Theorem~31’. But by-hand typing the 31 is poor practice. Instead you should write a label such as \texttt{\label{eq:GreensThm}} and then reference it, as with See equation~\texttt{\ref{eq:GreensThm}}. \LaTeX{} will automatically work out the number, put it into the output, and will change that number later if needed.

We will see this with Theorem~\texttt{\ref{th:GreensThm}}. \% forward reference

\begin{theorem} \label{th:GreensThm}
...
\end{theorem}

...  
See Theorem~\texttt{\ref{th:GreensThm}} on page~\texttt{\pageref{th:GreensThm}}.

\LaTeX{} tracks cross reference information in a file having the extension \texttt{.aux} and with the same base name as the file containing the \texttt{\label}. So if \texttt{\label} is in \texttt{calculus.tex} then the information is in \texttt{calculus.aux}. \LaTeX{} puts the information in that file every time it runs across a \texttt{\label}.

The behavior described in the prior paragraph results in a quirk that happens when your document has a forward reference, a \texttt{\ref} that appears before the associated \texttt{\label}. If this is the first time that you are compiling the document then you will get ‘\LaTeX~Warning: Label(s) may have changed. Rerun to get cross references right’ and in the output the forward reference will appear as two question marks ‘??’, in boldface. A similar thing happens if you change some things so the references changes; you get the same warning and the output contains the old reference information. In both cases, resolve this by compiling the document a second time.

The \texttt{cleveref} package enhances \LaTeX{}’s cross referencing features. You can arrange that if you enter \texttt{\begin{thm}\label{th:Nerode}...\end{thm}} then \texttt{\cref{th:Nerode}} will output ‘Theorem 3.21’, without you having to enter the “Theorem.”

7.1 \texttt{\label}

Synopsis:

\texttt{\label{key}}

Assign a reference number to \texttt{key}. In ordinary text \texttt{\label{key}} assigns to \texttt{key} the number of the current sectional unit. Inside an environment with numbering, such as a \texttt{table} or \texttt{theorem} environment, \texttt{\label{key}} assigns to \texttt{key} the number of that environment. Retrieve the assigned number with the \texttt{\ref{key}} command (see Section 7.3 [\ref], page 49).

A key name can consist of any sequence of letters, digits, or common punctuation characters. Upper and lowercase letters are distinguished, as usual.

A common convention is to use labels consisting of a prefix and a suffix separated by a colon or period. Thus, \texttt{\label{fig:Post}} is a label for a figure with a portrait of Emil Post. This helps to avoid accidentally creating two labels with the same name, and makes your source more readable. Some commonly-used prefixes:

\begin{itemize}
\item \texttt{ch} for chapters
\end{itemize}
sec
subsec for lower-level sectioning commands
fig for figures
tab for tables
eq for equations

In the auxiliary file the reference information is kept as the text of a command of the form `\newlabel{label}{currentlabel}{pagenumber}`. Here `currentlabel` is the current value of the macro `@currentlabel` that is usually updated whenever you call `\refstepcounter{counter}`.

Below, the key `sec:test` will get the number of the current section and the key `fig:test` will get the number of the figure. (Incidentally, put labels after captions in figures and tables.)

\section{section name}
\label{sec:test}
This is Section~\ref{sec:test}.
\begin{figure}
  ...
  \caption{caption text}
  \label{fig:test}
\end{figure}
See Figure~\ref{fig:test}.

7.2 \pageref

Synopsis:

\pageref{key}

Produce the page number of the place in the text where the corresponding `\label{key}` command appears.

If there is no `\label{key}` then you get something like `LaTeX Warning: Reference `th:GrensThm' on page 1 undefined on input line 11.'

Below, the `\label{eq:main}` is used both for the formula number and for the page number. (Note that the two references are forward references so this document would need to be compiled twice to resolve those.)

The main result is formula~\ref{eq:main} on page~\pageref{eq:main}.

\begin{equation}
\mathbf{P} = \mathbf{NP}
\end{equation}

7.3 \ref

Synopsis:

\ref{key}
Chapter 7: Cross references

Produces the number of the sectional unit, equation, footnote, figure, . . ., of the corresponding \label command (see Section 7.1 [\label], page 48). It does not produce any text, such as the word ‘Section’ or ‘Figure’, just the bare number itself.

If there is no \label{key} then you get something like ‘LaTeX Warning: Reference `th:GrensThm' on page 1 undefined on input line 11.’

In this example the \ref{popular} produces ‘2’. Note that it is a forward reference since it comes before \label{popular} so this document would have to be compiled twice.

The most widely-used format is item number~\ref{popular}.
\begin{enumerate}
\item Plain \TeX
\item \label{popular} \LaTeX
\item Con\TeX t
\end{enumerate}

The cleveref package includes text such as ‘Theorem’ in the reference. See the documentation on CTAN.

7.4 xr package

Synopsis:
\begin{verbatim}
usepackage{xr}
\externaldocument{document-basename}
\end{verbatim}
or
\begin{verbatim}
usepackage{xr}
\externaldocument[reference-prefix]{document-basename}
\end{verbatim}

Make cross references to the external document document-basename.tex.

Here is an example. If lectures.tex has this in the preamble
\begin{verbatim}
usepackage{xr}
\externaldocument{exercises}
\externaldocument[H-]{hints}
\externaldocument{answers}
\end{verbatim}
then it can use cross reference labels from the other three documents. Suppose that exercises.tex has an enumerated list containing this,
\begin{verbatim}
\item \label{exer:EulersThm} What if every vertex has odd degree?
\end{verbatim}
and hints.tex has an enumerated list with this,
\begin{verbatim}
\item \label{exer:EulersThm} Distinguish the case of two vertices.
\end{verbatim}
and answers.tex has an enumerated list with this,
\begin{verbatim}
\item \label{ans:EulersThm} There is no Euler path, except if there are exactly two vertices.
\end{verbatim}

After compiling the exercises, hints, and answers documents, entering this in the body of lectures.tex will result in the lectures getting the reference numbers used in the other documents.

See Exercise~\ref{exer:EulersThm}, with Hint~\ref{H-exer:EulersThm}. The solution is Answer~\ref{ans:EulersThm}. 
The prefix $H-$ for the reference from the hints file is needed because the label in the hints file is the same as the label in the exercises file. Without that prefix, both references would get the number from the later file.

Note: if the document uses the \texttt{hyperref} package then in place of \texttt{xr}, put \texttt{\usepackage{xr-hyper}} before the \texttt{\usepackage{hyperref}}. Also, if any of the multiple documents uses \texttt{hyperref} then they all must use it.
8 Environments

\LaTeX provides many environments for delimiting certain behavior. An environment begins with \texttt{\begin} and ends with \texttt{\end}, like this:
\begin{verbatim}
\begin{environment-name}
...
\end{environment-name}
\end{verbatim}

The \texttt{environment-name} at the beginning must exactly match that at the end. For instance, the input \texttt{\begin{table*}...\end{table}} will cause an error like: ‘\texttt{! LaTeX Error: \begin{table*} on input line 5 ended by \end{table}.}’

Environments are executed within a group.

8.1 abstract

Synopsis:
\begin{verbatim}
\begin{abstract}
...
\end{abstract}
\end{verbatim}

Produce an abstract, possibly of multiple paragraphs. This environment is only defined in the \texttt{article} and \texttt{report} document classes (see Chapter 3 \texttt{[Document classes]}, page 10).

Using the example below in the \texttt{article} class produces a displayed paragraph. Document class option \texttt{titlepage} causes the abstract to be on a separate page (see Section 3.1 \texttt{[Document class options]}, page 10); this is the default only in the \texttt{report} class.
\begin{verbatim}
\begin{abstract}
We compare all known accounts of the proposal made by Porter Alexander to Robert E Lee at the Appomattox Court House that the army continue in a guerrilla war, which Lee refused.
\end{abstract}
\end{verbatim}

The next example produces a one column abstract in a two column document (for a more flexible solution, use the package \texttt{abstract}).
\begin{verbatim}
\documentclass[twocolumn]{article}
...
\begin{document}
\maketitle
\begin{abstract}
Ruth was not just the Sultan of Swat, he was the entire swat team.
\end{abstract}
\end{document}
\end{verbatim}

{ % by-hand insert a footnote at page bottom}
8.2 array

Synopsis:
\begin{array}{cols}
  \text{column 1 entry} & \text{column 2 entry} \ldots & \text{column n entry} \\
  \ldots
\end{array}

or:
\begin{array}{pos cols}
  \text{column 1 entry} & \text{column 2 entry} \ldots & \text{column n entry} \\
  \ldots
\end{array}

Produce a mathematical array. This environment can only be used in math mode (see Chapter 17 [Modes], page 177), and normally appears within a displayed mathematics environment such as equation (see Section 8.9 [equation], page 59). Inside of each row the column entries are separated by an ampersand, (&). Rows are terminated with double-backslashes (see Section 9.1 [\], page 99).

This example shows a three by three array.
\begin{equation*}
\chi(x) = \left| \begin{array}{ccc}
  x-a & -b & -c \\
  -d & x-e & -f \\
  -g & -h & x-i
\end{array} \right|
\end{equation*}

The required argument cols describes the number of columns, their alignment, and the formatting of the intercolumn regions. For instance, \begin{array}{rcl} \ldots \end{array} gives three columns: the first flush right, the second centered, and the third flush left. See Section 8.23 [tabular], page 85, for the complete description of cols and of the other common features of the two environments, including the optional pos argument.

There are two ways that array diverges from tabular. The first is that array entries are typeset in math mode, in textstyle (see Section 16.7 [Math styles], page 174) except if the cols definition specifies the column with p{...}, which causes the entry to be typeset in text mode. The second is that, instead of tabular’s parameter \tabcolsep, L\TeX{}’s intercolumn space in an array is governed by \arraycolsep, which gives half the width between columns. The default for this is ‘5pt’ so that between two columns comes 10 pt of space.

To obtain arrays with braces the standard is to use the amsmath package. It comes with environments pmatrix for an array surrounded by parentheses (\ldots), bmatrix for
an array surrounded by square brackets \([\ldots]\), \texttt{Bmatrix} for an array surrounded by curly braces \(\{\ldots\}\), \texttt{vmatrix} for an array surrounded by vertical bars \(\mid\ldots\mid\), and \texttt{Vmatrix} for an array surrounded by double vertical bars \(\|\ldots\|\), along with a number of other array constructs.

The next example uses the \texttt{amsmath} package.

\begin{verbatim}
\usepackage{amsmath} % in preamble
\begin{equation}
\begin{vmatrix} % array with vert lines 
a & b \\
c & d
\end{vmatrix}=ad-bc\end{equation}
\end{verbatim}

There are many packages concerning arrays. The \texttt{array} package has many useful extensions, including more column types. The \texttt{dcolumn} package adds a column type to center on a decimal point. For both see the documentation on CTAN.

8.3 center

Synopsis:

\begin{verbatim}
\begin{center}
 line1 \\
 line2 \\
\ldots
\end{center}
\end{verbatim}

Create a new paragraph consisting of a sequence of lines that are centered within the left and right margins. Use double-backslash, \texttt{\\}, to get a line break (see Section 9.1 [\\], page 99). If some text is too long to fit on a line then \LaTeX{} will insert line breaks that avoid hyphenation and avoid stretching or shrinking any interword space.

This environment inserts space above and below the text body. See Section 8.3.1 [\texttt{centering}], page 55, to avoid such space, for example inside a \texttt{figure} environment.

This example produces three centered lines. There is extra vertical space between the last two lines.

\begin{verbatim}
\begin{center}
 A Thesis Submitted in Partial Fulfillment \\
of the Requirements of \\
[0.5ex]
 the School of Environmental Engineering
\end{center}
\end{verbatim}

In this example, depending on the page’s line width, \LaTeX{} may choose a line break for the part before the double backslash. If so, it will center each of the two lines and if not it will center the single line. Then \LaTeX{} will break at the double backslash, and will center the ending.

\begin{verbatim}
\begin{center}
 My father considered that anyone who went to chapel and didn't drink 
alcohol was not to be tolerated.\\
\end{center}
\end{verbatim}
I grew up in that belief. ---Richard Burton
\end{center}

A double backslash after the final line is optional. If present it doesn’t add any vertical space.

In a two-column document the text is centered in a column, not in the entire page.

8.3.1 \centering

Synopsis:

{\centering ... }

or

\begin{group}
  \centering ...
\end{group}

Center the material in its scope. It is most often used inside an environment such as figure, or in a parbox.

This example’s \centering declaration causes the graphic to be horizontally centered.

\begin{figure}
  \centering
  \includegraphics[width=0.6\textwidth]{ctan_lion.png}
  \caption{CTAN Lion} \label{fig:CTANLion}
\end{figure}

The scope of this \centering ends with the \end{figure}.

Unlike the center environment, the \centering command does not add vertical space above and below the text. That’s its advantage in the above example; there is not an excess of space.

It also does not start a new paragraph; it simply changes how \LaTeX{} formats paragraph units. If \texttt{ww \{\centering xx \ \ yy\} zz} is surrounded by blank lines then \LaTeX{} will create a paragraph whose first line ‘\texttt{ww xx}’ is centered and whose second line, not centered, contains ‘\texttt{yy zz}’. Usually what is desired is for the scope of the declaration to contain a blank line or the \end command of an environment such as figure or table that ends the paragraph unit. Thus, if \texttt{\{\centering xx \ \ yy\par\} zz} is surrounded by blank lines then it makes a new paragraph with two centered lines ‘xx’ and ‘yy’, followed by a new paragraph with ‘zz’ that is formatted as usual.

8.4 description

Synopsis:

\begin{description}
  \item[\texttt{label of first item}] \texttt{text of first item}
  \item[\texttt{label of second item}] \texttt{text of second item}
  ...
\end{description}

Environment to make a list of labeled items. Each item’s \texttt{label} is typeset in bold and is flush left, so that long labels continue into the first line of the item text. There must
be at least one item; having none causes the \LaTeX error ‘Something’s wrong--perhaps a missing \item’.

This example shows the environment used for a sequence of definitions.

\begin{description}
\item[lama] A priest.
\item[llama] A beast.
\end{description}

The labels ‘lama’ and ‘llama’ are output in boldface, with the left edge on the left margin.

Start list items with the \item command (see Section 8.16.1 \[item\], page 70). Use the optional labels, as in \item[Main point], because there is no sensible default. Following the \item is optional text, which may contain multiple paragraphs.

Since the labels are in bold style, if the label text calls for a font change given in argument style (see Section 4.2 [Font styles], page 20) then it will come out bold. For instance, if the label text calls for typewriter with \item[\tt{label text}] then it will appear in bold typewriter, if that is available. The simplest way around this, in this example to get non-bold typewriter, is to use declarative style: \item[\tt{label text}]. Similarly, get the standard roman font with \item[\rm{label text}].

For other major \LaTeX labelled list environments, see Section 8.14 [itemize], page 63, and Section 8.7 [enumerate], page 57. Unlike those environments, nesting description environments does not change the default label; it is boldface and flush left at all levels.

For information about list layout parameters, including the default values, and for information about customizing list layout, see Section 8.16 [list], page 65. The package enumitem is useful for customizing lists.

This example changes the description labels to small caps.

\renewcommand{\descriptionlabel}[1]{% 
  \hspace{\labelsep}\textsc{#1}}

8.5 displaymath

Synopsis:

\begin{displaymath}
\begin{array}{c}
mathematical text\\
\end{array}
\end{displaymath}

Environment to typeset the mathematical text on its own line, in display style and centered. To make the text be flush-left use the global option fleqn; see Section 3.1 [Document class options], page 10.

In the displaymath environment no equation number is added to the math text. One way to get an equation number is to use the equation environment (see Section 8.9 [equation], page 59).

\LaTeX will not break the math text across lines.

Note that the amsmath package has significantly more extensive displayed equation facilities. For example, there are a number of ways in that package for having math text broken across lines.
The construct \textbackslash [ \texttt{math} \] is a synonym for the environment \texttt{begin\{displaymath\} math \end{displaymath}} but the latter is easier to work with in the source; for instance, searching for a square bracket may get false positives but the word \texttt{displaymath} will likely be unique.

The construct $$math$$ from Plain \TeX is sometimes used as a synonym for \LaTeX\’s \texttt{displaymath}. It is not a synonym, and is not officially supported in \LaTeX at all; $$ doesn’t support the \texttt{fleqn} option (see Section 3.1 [Document class options], page 10), has different vertical spacing, and doesn’t perform consistency checks.

The output from this example is centered and alone on its line.

\begin{displaymath}
\int_1^2 x^2 \, dx = \frac{7}{3}
\end{displaymath}

Also, the integral sign is larger than the inline version \(( \int_1^2 x^2 \, dx = \frac{7}{3} \) produces.

8.6 document

The \texttt{document} environment encloses the entire body of a document. It is required in every \LaTeX document. See Section 2.1 [Starting and ending], page 2.

8.6.1 \texttt{\textbackslash AtBeginDocument}

Synopsis:

\texttt{AtBeginDocument\{code\}}

Save \textit{code} and execute it when \texttt{begin\{document\}} is executed, at the very end of the preamble. The code is executed after the font selection tables have been set up, so the normal font for the document is the current font. However, the code is executed as part of the preamble so you cannot do any typesetting with it.

You can issue this command more than once; the successive code lines will be executed in the order that you gave them.

8.6.2 \texttt{\textbackslash AtEndDocument}

Synopsis:

\texttt{AtEndDocument\{code\}}

Save \textit{code} and execute it near the end of the document. Specifically, it is executed when \texttt{end\{document\}} is executed, before the final page is finished and before any leftover floating environments are processed. If you want some of the code to be executed after these two processes then include a \texttt{clearpage} at the appropriate point in \textit{code}.

You can issue this command more than once; the successive code lines will be executed in the order that you gave them.

8.7 \texttt{enumerate}

Synopsis:

\texttt{begin\{enumerate\}}

\item\texttt{optional label of first item} \textit{text of first item}
\item\texttt{optional label of second item} \textit{text of second item}
... 
\end{enumerate}

Environment to produce a numbered list of items. The format of the label numbering depends on the nesting level of this environment; see below. The default top-level numbering is ‘1.’, ‘2.’, etc. Each enumerate list environment must have at least one item; having none causes the \LaTeX error ‘Something's wrong--perhaps a missing \item’.

This example gives the first two finishers in the 1908 Olympic marathon. As a top-level list the labels would come out as ‘1.’ and ‘2.’.

\begin{enumerate}
\item Johnny Hayes (USA)
\item Charles Hefferon (RSA)
\end{enumerate}

Start list items with the \item command (see Section 8.16.1 [item], page 70). If you give \item an optional argument by following it with square brackets, as in \item[Interstitial label], then the next item will continue the interrupted sequence (see Section 8.16.1 [item], page 70). That is, you will get labels like ‘1.’, then ‘Interstitial label’, then ‘2.’. Following the \item is optional text, which may contain multiple paragraphs.

Enumerations may be nested within other enumerate environments, or within any paragraph-making environment such as itemize (see Section 8.14 [itemize], page 63), up to four levels deep. This gives \LaTeX’s default for the format at each nesting level, where 1 is the top level, the outermost level.

1. arabic number followed by a period: ‘1.’, ‘2.’, ...
2. lowercase letter inside parentheses: ‘(a)’, ‘(b)’ ...
3. lowercase roman numeral followed by a period: ‘i.’, ‘ii.’ ...
4. uppercase letter followed by a period: ‘A.’, ‘B.’ ...

The enumerate environment uses the counters \enumi through \enumiv (see Chapter 13 [Counters], page 133).

For other major \LaTeX labeled list environments, see Section 8.4 [description], page 55, and Section 8.14 [itemize], page 63. For information about list layout parameters, including the default values, and for information about customizing list layout, see Section 8.16 [list], page 65. The package enumitem is useful for customizing lists.

To change the format of the label use \renewcommand (see Section 12.1 [newcommand & \renewcommand], page 112) on the commands \labelenumi through \labelenumiv. For instance, this first level list will be labelled with uppercase letters, in boldface, and without a trailing period.

\renewcommand{\labelenumi}{\textbf{\Alph{enumi}}}
\begin{enumerate}
\item Shows as boldface A
\item Shows as boldface B
\end{enumerate}

For a list of counter-labeling commands see Section 13.1 [alph Alph arabic roman Roman \fnsymbol], page 133.


8.8 eqnarray

The **eqnarray** environment is obsolete. It has infelicities, including spacing that is inconsistent with other mathematics elements. (See “Avoid eqnarray!” by Lars Madsen [https://tug.org/TUGboat/tb33-1/tb103madsen.pdf](https://tug.org/TUGboat/tb33-1/tb103madsen.pdf)). New documents should include the **amsmath** package and use the displayed mathematics environments provided there, such as the **align** environment. We include a description only for completeness and for working with old documents.

**Synopsis:**

\begin{eqnarray}
\text{first formula left} & \text{first formula middle} & \text{first formula right} \\
\vdots
\end{eqnarray}

or

\begin{eqnarray*}
\text{first formula left} & \text{first formula middle} & \text{first formula right} \\
\vdots
\end{eqnarray*}

Display a sequence of equations or inequalities. The left and right sides are typeset in display mode, while the middle is typeset in text mode.

It is similar to a three-column **array** environment, with items within a row separated by an ampersand (&), and with rows separated by double backslash (\). The starred form of line break (\*) can also be used to separate equations, and will disallow a page break there (see Section 9.1, page 99).

The unstarred form **eqnarray** places an equation number on every line (using the **equation** counter), unless that line contains a **nonumber** command. The starred form **eqnarray** omits equation numbering, while otherwise being the same.

The command \texttt{\lefteqn} is used for splitting long formulas across lines. It typesets its argument in display style flush left in a box of zero width.

This example shows three lines. The first two lines make an inequality, while the third line has not entry on the left side.

\begin{eqnarray*}
\texttt{\lefteqn} & x_1+x_2+\cdots+x_n & \text{\texttt{\\}}}
& \text{\texttt{\&}} & y_1+y_2+\cdots+y_n & \text{\texttt{\\}}}
& \text{\texttt{\&}} & z+y_3+\cdots+y_n
\end{eqnarray*}

8.9 equation

**Synopsis:**

\begin{equation}
\text{mathematical text}
\end{equation}

The same as a **displaymath** environment (see Section 8.5 [displaymath], page 56) except that **\LaTeX** puts an equation number flush to the right margin. The equation number is generated using the **equation** counter.
You should have no blank lines between \begin{equation} and \end{equation}, or \LaTeX{} will tell you that there is a missing dollar sign.

The package \texttt{amsmath} package has extensive displayed equation facilities. New documents should include this package.

\section*{8.10 \texttt{figure}}

Synopsis:

\begin{verbatim}
\begin{figure}[placement]
  \centering
  figure body
  \caption[loftitle]{title} \% optional
  \label{label} \% optional
\end{figure}
\end{verbatim}

or:

\begin{verbatim}
\begin{figure*}[placement]
  \centering
  figure body
  \caption[loftitle]{title} \% optional
  \label{label} \% optional
\end{figure*}
\end{verbatim}

Figures are for material that is not part of the normal text. An example is material that you cannot have split between two pages, such as a graphic. Because of this, \LaTeX{} does not typeset figures in sequence with normal text but instead “floats” them to a convenient place, such as the top of a following page (see Section 5.7 \texttt{[Floats]}, page 32).

The \texttt{figure body} can consist of imported graphics (see Chapter 21 \texttt{[Graphics]}, page 204), or text, \LaTeX{} commands, etc. It is typeset in a \texttt{parbox} of width \texttt{\textwidth}.

The possible values of \texttt{placement} are \texttt{h} for ‘\texttt{here}’, \texttt{t} for ‘\texttt{top}’, \texttt{b} for ‘\texttt{bottom}’, and \texttt{p} for ‘on a separate page of floats’. For the effect of these options on the float placement algorithm, see Section 5.7 \texttt{[Floats]}, page 32.

The starred form \texttt{figure*} is used when a document is in double-column mode (see Section 5.2 \texttt{[\texttt{twocolumn}]}, page 26). It produces a figure that spans both columns, at the top of the page. To add the possibility of placing at a page bottom see the discussion of \texttt{placement} \texttt{b} in Section 5.7 \texttt{[Floats]}, page 32.

The label is optional; it is used for cross references (see Chapter 7 \texttt{[Cross references]}, page 48). The optional \texttt{\caption} command specifies caption text for the figure (see Section 5.7.1 \texttt{[\texttt{\caption}]}, page 34). By default it is numbered. If \texttt{loftitle} is present, it is used in the list of figures instead of \texttt{title} (see Section 25.1 \texttt{[Table of contents etc.]}), page 233).

This example makes a figure out of a graphic. \LaTeX{} will place that graphic and its caption at the top of a page or, if it is pushed to the end of the document, on a page of floats.

\begin{verbatim}
\usepackage{graphicx} % in preamble
...
\begin{figure}[t]
  \centering
  \includegraphics[width=0.5\textwidth]{CTANlion.png}
\end{figure}
\end{verbatim}
8.11 filecontents

Synopsis:
\begin{filecontents}[option]{filename}
  text
\end{filecontents}

or
\begin{filecontents*}[option]{filename}
  text
\end{filecontents*}

Create a file named \texttt{filename} in the current directory (or the output directory, if specified; see [output directory], page 260) and write \texttt{text} to it. By default, an existing file is not overwritten.

The unstarred version of the environment \texttt{filecontents} prefixes the content of the created file with a header of \TeX{} comments; see the example below. The starred version \texttt{filecontents*} does not include the header.

The possible options are:

\begin{itemize}
  \item \texttt{force} \\
  \hspace{1em} Overwrite an existing file.
  \item \texttt{overwrite} \\
  \hspace{1em} Overwrite an existing file.
  \item \texttt{noheader} \\
  \hspace{1em} Omit the header. Equivalent to using \texttt{filecontents*}.
  \item \texttt{nosearch} \\
  \hspace{1em} Only check the current directory (and the output directory, if specified) for an existing file, not the entire search path.
\end{itemize}

These options were added in a 2019 release of \LaTeX{}.

This environment can be used anywhere in the preamble, although it often appears before the \texttt{documentclass} command. It is commonly used to create a \texttt{.bib} or other such data file from the main document source, to make the source file self-contained. Similarly, it can be used to create a custom style or class file, again making the source self-contained.

For example, this document:
\begin{verbatim}
\documentclass{article}
\begin{filecontents}{JH.sty}
\newcommand{\myname}{Jim Hef{}feron}
\end{filecontents}
\usepackage{JH}
\begin{document}
Article by \myname.
\end{document}
\end{verbatim}
produces this file \texttt{JH.sty}:
\begin{verbatim}
%% LaTeX2e file `JH.sty'
\end{verbatim}
8.12 flushleft

Synopsis:

\begin{flushleft}
 line1 \\
 line2 \\
 ...
\end{flushleft}

An environment that creates a paragraph whose lines are flush to the left-hand margin, and ragged right. If you have lines that are too long then \LaTeX{} will linebreak them in a way that avoids hyphenation and stretching or shrinking interword spaces. To force a new line use a double backslash, \. For the declaration form see Section 8.12.1 \raggedright, page 62.

This creates a box of text that is at most 3 inches wide, with the text flush left and ragged right.

\noindent\begin{minipage}{3in}
\begin{flushleft}
 A long sentence that will be broken by \LaTeX{}
 at a convenient spot. \\
 And, a fresh line forced by the double backslash.
\end{flushleft}
\end{minipage}

8.12.1 \raggedright

Synopses:

\{\raggedright ... \}

or

\begin{environment} \raggedright \\
 ...
\end{environment}

A declaration which causes lines to be flush to the left margin and ragged right. It can be used inside an environment such as \quote or in a \parbox. For the environment form see Section 8.12 [flushleft], page 62.

Unlike the \flushleft environment, the \raggedright command does not start a new paragraph; it only changes how \LaTeX{} formats paragraph units. To affect a paragraph unit’s format, the scope of the declaration must contain the blank line or \end command that ends the paragraph unit.

Here \raggedright in each second column keeps \LaTeX{} from doing awkward typesetting to fit the text into the narrow column. Note that \raggedright is inside the curly braces {\ldots} to delimit its effect.

\begin{tabular}{rp{2in}}
Team alpha \&\raggedright This team does all the real work. \&
Team beta \&\raggedright This team ensures that the water
cooler is never empty. \&
\end{tabular}

8.13 \texttt{flushright}

\begin{flushright}
\begin{tabular}{l}
line1 \\
line2 \\
\end{tabular}
\end{flushright}

An environment that creates a paragraph whose lines are flush to the right-hand margin and ragged left. If you have lines that are too long to fit the margins then \TeX will linebreak them in a way that avoids hyphenation and stretching or shrinking inter-word spaces. To force a new line use a double backslash, \\ . For the declaration form see Section 8.13.1 \texttt{\raggedleft}, page 63.

For an example related to this environment, see Section 8.12 [\texttt{flushleft}], page 62, where one just have mutatis mutandis to replace \texttt{flushleft} by \texttt{flushright}.

8.13.1 \texttt{\raggedleft}

Synopses:

\{\texttt{\raggedleft} ... \}

or

\begin{flushright}
\begin{environment} \texttt{\raggedleft}
\end{environment}
\end{flushright}

A declaration which causes lines to be flush to the right margin and ragged left. It can be used inside an environment such as \texttt{quote} or in a \texttt{parbox}. For the environment form see Section 8.13 [\texttt{flushright}], page 63.

Unlike the \texttt{flushright} environment, the \texttt{\raggedleft} command does not start a new paragraph; it only changes how \TeX formats paragraph units. To affect a paragraph unit’s formatting, the scope of the declaration must contain the blank line or \texttt{\end} command that ends the paragraph unit.

See Section 8.12.1 [\texttt{\raggedright}], page 62, for an example related to this environment; just replace \texttt{\raggedright} there by \texttt{\raggedleft}.

8.14 \texttt{itemize}

Synopsis:

\begin{itemize}
\item \texttt{optional label of first item} text of first item
\item \texttt{optional label of second item} text of second item
\end{itemize}
Produce an *unordered list*, sometimes called a bullet list. There must be at least one `\item` within the environment; having none causes the \LaTeX error `Something's wrong--perhaps a missing `\item'`.

This gives a two-item list.

```
\begin{itemize}
  \item Pencil and watercolor sketch by Cassandra
  \item Rice portrait
\end{itemize}
```

With the default locale—without loading e.g. `babel` package with another language than USenglish—as a top-level list each label would come out as a bullet, •. The format of the labeling depends on the nesting level; see below.

Start list items with the `\item` command (see Section 8.16.1 `item`, page 70). If you give `\item` an optional argument by following it with square brackets, as in `\item[Optional label]`, then by default *Optional label* will appear in bold and be flush right, so it could extend into the left margin. For labels that are flush left see the Section 8.4 `description`, page 55, environment. Following the `\item` is the text of the item, which may be empty or contain multiple paragraphs.

Unordered lists can be nested within one another, up to four levels deep. They can also be nested within other paragraph-making environments, such as `enumerate` (see Section 8.7 `enumerate`, page 57).

The `itemize` environment uses the commands `\labelitemi` through `\labelitemiv` to produce the default label (note the convention of lowercase roman numerals at the end of the command names that signify the nesting level). These are the default marks at each level.

1. • (bullet, from `\textbullet`)
2. -- (bold en-dash, from `\normalfont\bfseries\textendash`)
3. * (asterisk, from `\textasteriskcentered`)
4. · (vertically centered dot, from `\textperiodcentered`)

Change the labels with `\renewcommand`. For instance, this makes the first level use diamonds.

```
\renewcommand{\labelitemi}{$\diamond$}
```

The distance between the left margin of the enclosing environment and the left margin of the `itemize` list is determined by the parameters `\leftmargini` through `\leftmarginvi`. (This also uses the convention of using lowercase roman numerals a the end of the command name to denote the nesting level.) The defaults are: 2.5em in level 1 (2em in two-column mode), 2.2em in level 2, 1.87em in level 3, and 1.7em in level 4, with smaller values for more deeply nested levels.

For other major \LaTeX labeled list environments, see Section 8.4 `description`, page 55, and Section 8.7 `enumerate`, page 57. The `itemize`, `enumerate` and `description` environment use the same list layout parameters. For a description, including the default values, and for information about customizing list layout, see Section 8.16 `list`, page 65. The package `enumitem` is useful for customizing lists.

This example greatly reduces the margin space for outermost itemized lists.

```
\setlength{\leftmargini}{1.25em} % default 2.5em
```
Especially for lists with short items, it may be desirable to elide space between items. Here is an example defining an `itemize*` environment with no extra spacing between items, or between paragraphs within a single item (`\parskip` is not list-specific, see Section 15.3 [\parindent & \parskip], page 146):

```latex
\newenvironment{itemize*}{}
{\begin{itemize}
  \setlength{\itemsep}{0pt}
  \setlength{\parsep}{0pt}
  \setlength{\parskip}{0pt}
\end{itemize}}
```

8.15 letter environment: writing letters

This environment is used for creating letters. See Chapter 26 [Letters], page 247.

8.16 list

Synopsis:

```latex
\begin{list} {\labeling} {\spacing}
  \item [optional label of first item] text of first item
  \item [optional label of second item] text of second item
  ...
\end{list}
```

An environment for constructing lists.

Note that this environment does not typically appear in the document body. Most lists created by \LaTeX{} authors are the ones that come standard: the `description`, `enumerate`, and `itemize` environments (see Section 8.4 [description], page 55, Section 8.7 [enumerate], page 57, and Section 8.14 [itemize], page 63).

Instead, the `list` environment is most often used in macros. For example, many standard \LaTeX{} environments that do not immediately appear to be lists are in fact constructed using `list`, including `quotation`, `quote`, and `center` (see Section 8.20 [quotation & quote], page 82, see Section 8.3 [center], page 54).

This uses the `list` environment to define a new custom environment.

```latex
\newcounter{namedlistcounter} % number the items
\newenvironment{named}{}
{\begin{list}
  {Item~\Roman{namedlistcounter}.} % labeling
  {\usecounter{namedlistcounter}} % set counter
  {\setlength{\leftmargin}{3.5em}} % set spacing
\end{list}}
```

```latex
\begin{named}
\item Shows as ``Item~I.''
\item[Special label.] Shows as ``Special label.''
```
\item Shows as "Item II."
\end{named}

The mandatory first argument \texttt{labeling} specifies the default labeling of list items. It can contain text and \LaTeX\ commands, as above where it contains both \texttt{Item} and \texttt{\Roman{...}}. \LaTeX\ forms the label by putting the \texttt{labeling} argument in a box of width \texttt{\labelwidth}. If the label is wider than that, the additional material extends to the right. When making an instance of a \texttt{list} you can override the default labeling by giving \item an optional argument by including square braces and the text, as in the above \item[Special label.]; see Section 8.16.1 \texttt{[item]}, page 70.

The mandatory second argument \texttt{spacing} has a list of commands. This list can be empty. A command that can go in here is \texttt{\usecounter{countername}} (see Section 13.2 \texttt{[usecounter]}, page 134). Use this to tell \LaTeX\ to number the items using the given counter. The counter will be reset to zero each time \LaTeX\ enters the environment, and the counter is incremented by one each time \LaTeX\ encounters an \item that does not have an optional argument.

Another command that can go in \texttt{spacing} is \texttt{\makelabel}, which constructs the label box. By default it puts the contents flush right. Its only argument is the label, which it typesets in LR mode (see Chapter 17 [Modes], page 177). One example of changing its definition is that to the above \texttt{named} example, before the definition of the environment add \texttt{\newcommand{\namedmakelabel}[1]{\textsc{#1}}} and between the \texttt{\setlength} command and the parenthesis that closes the \texttt{spacing} argument also add \texttt{\let\makelabel\namedmakelabel}. Then the labels will be typeset in small caps. Similarly, changing the second code line to \texttt{\let\makelabel\fbox} puts the labels inside a framed box. Another example of the \texttt{\makelabel} command is below, in the definition of the \texttt{redlabel} environment.

Also often in \texttt{spacing} are commands to redefine the spacing for the list. Below are the spacing parameters with their default values. (Default values for derived environments such as \texttt{itemize} can be different than the values shown here.) See also the figure that follows the list. Each is a length (see Chapter 14 [Lengths], page 137). The vertical spaces are normally rubber lengths, with plus and minus components, to give \TeX\ flexibility in setting the page. Change each with a command such as \texttt{\setlength{\itemsep}{2pt plus1pt minus1pt}}. For some effects these lengths should be zero or negative.

\itemindent
Extra horizontal space indentation, beyond \texttt{leftmargin}, of the first line each item. Its default value is 0pt.

\itemsep
Vertical space between items, beyond the \texttt{\parsep}. The defaults for the first three levels in \LaTeX\’s \texttt{article}, \texttt{book}, and \texttt{report} classes at 10 point size are: 4pt plus2pt minus1pt, \texttt{\parsep} (that is, 2pt plus1pt minus1pt), and \texttt{\topsep} (that is, 2pt plus1pt minus1pt). The defaults at 11 point are: 4.5pt plus2pt minus1pt, \texttt{\parsep} (that is, 2pt plus1pt minus1pt), and \texttt{\topsep} (that is, 2pt plus1pt minus1pt). The defaults at 12 point are: 5pt plus2.5pt minus1pt, \texttt{\parsep} (that is, 2.5pt plus1pt minus1pt), and \texttt{\topsep} (that is, 2.5pt plus1pt minus1pt).
\labelsep
Horizontal space between the label and text of an item. The default for \TeX’s ‘article’, ‘book’, and ‘report’ classes is 0.5em.

\labelwidth
Horizontal width. The box containing the label is nominally this wide. If \makelabel returns text that is wider than this then the first line of the item will be indented to make room for this extra material. If \makelabel returns text of width less than or equal to \labelwidth then \TeX’s default is that the label is typeset flush right in a box of this width.

The left edge of the label box is \leftmargin+\itemindent-\labelsep-\labelwidth from the left margin of the enclosing environment.

The default for \TeX’s ‘article’, ‘book’, and ‘report’ classes at the top level is \leftmargini-\labelsep, (which is 2em in one column mode and 1.5em in two column mode). At the second level it is \leftmarginii-\labelsep, and at the third level it is \leftmarginiii-\labelsep. These definitions make the label’s left edge coincide with the left margin of the enclosing environment.

\leftmargin
Horizontal space between the left margin of the enclosing environment (or the left margin of the page if this is a top-level list), and the left margin of this list. It must be non-negative.

In the standard \TeX document classes, a top-level list has this set to the value of \leftmargini, while a list that is nested inside a top-level list has this margin set to \leftmarginii. More deeply nested lists get the values of \leftmarginiii through \leftmarginvi. (Nesting greater than level five generates the error message ‘Too deeply nested’.)

The defaults for the first three levels in \TeX’s ‘article’, ‘book’, and ‘report’ classes are: \leftmargini is 2.5em (in two column mode, 2em), \leftmarginii is 2.2em, and \leftmarginiii is 1.87em.

\listparindent
Horizontal space of additional line indentation, beyond \leftmargin, for second and subsequent paragraphs within a list item. A negative value makes this an “outdent”. Its default value is 0pt.

\parsep
Vertical space between paragraphs within an item. The defaults for the first three levels in \TeX’s ‘article’, ‘book’, and ‘report’ classes at 10 point size are: 4pt plus2pt minus1pt, 2pt plus1pt minus1pt, and 0pt. The defaults at 11 point size are: 4.5pt plus2pt minus1pt, 2pt plus1pt minus1pt, and 0pt. The defaults at 12 point size are: 5pt plus2.5pt minus1pt, 2.5pt plus1pt minus1pt, and 0pt.

\partopsep
Vertical space added, beyond \topsep+\parsep, to the top and bottom of the entire environment if the list instance is preceded by a blank line. (A blank line in the \TeX source before the list changes spacing at both the top and bottom of the list; whether the line following the list is blank does not matter.)
The defaults for the first three levels in \LaTeX\’s \texttt{article}, \texttt{book}, and \texttt{report} classes at 10 point size are: 2pt plus1 minus1pt, 2pt plus1pt minus1pt, and 1pt plus0pt minus1pt. The defaults at 11 point are: 3pt plus1pt minus1pt, 3pt plus1pt minus1pt, and 1pt plus0pt minus1pt. The defaults at 12 point are: 3pt plus2pt minus3pt, 3pt plus2pt minus2pt, and 1pt plus0pt minus1pt.

\rightmargin

Horizontal space between the right margin of the list and the right margin of the enclosing environment. Its default value is 0pt. It must be non-negative.

\topsep

Vertical space added to both the top and bottom of the list, in addition to \parskip (see Section 15.3 \texttt{\parindent & \parskip}, page 146). The defaults for the first three levels in \LaTeX\’s \texttt{article}, \texttt{book}, and \texttt{report} classes at 10 point size are: 8pt plus2pt minus4pt, 4pt plus2pt minus1pt, and 2pt plus1pt minus1pt. The defaults at 11 point are: 9pt plus3pt minus5pt, 4.5pt plus2pt minus1pt, and 2pt plus1pt minus1pt. The defaults at 12 point are: 10pt plus4pt minus6pt, 5pt plus2.5pt minus1pt, and 2.5pt plus1pt minus1pt.

This shows the horizontal and vertical distances.

The lengths shown are listed below. The key relationship is that the right edge of the bracket for $h1$ equals the right edge of the bracket for $h4$, so that the left edge of the label box is at $h3+h4-(h0+h1)$.

$\topsep + \parskip$ if the list environment does not start a new paragraph, and $\topsep+\parskip+\partopsep$ if it does

$\parskip$

$\itemsep$\parskip

Same as $v0$. (This space is affected by whether a blank line appears in the source above the environment; whether a blank line appears in the source below the environment does not matter.)

$\labelwidth$
The list's left and right margins, shown above as $h3$ and $h5$, are with respect to the ones provided by the surrounding environment, or with respect to the page margins for a top-level list. The line width used for typesetting the list items is $\texttt{\linewidth}$ (see Section 5.5 [Page layout parameters], page 28). For instance, set the list's left margin to be one quarter of the distance between the left and right margins of the enclosing environment with $\texttt{\setlength{\leftmargin}{0.25\linewidth}}$.

Page breaking in a list structure is controlled by the three parameters below. For each, the $\LaTeX$ default is $-\texttt{@lowpenalty}$, that is, $-51$. Because it is negative, it somewhat encourages a page break at each spot. Change it with, e.g., $\texttt{@beginparpenalty=9999}$; a value of 10000 prohibits a page break.

$\texttt{@beginparpenalty}$

The page breaking penalty for breaking before the list (default $-51$).

$\texttt{@itempenalty}$

The page breaking penalty for breaking before a list item (default $-51$).

$\texttt{@endparpenalty}$

The page breaking penalty for breaking after a list (default $-51$).

The package $\texttt{enumitem}$ is useful for customizing lists.

This example has the labels in red. They are numbered, and the left edge of the label lines up with the left edge of the item text. See Section 13.2 [$\texttt{\usecounter}$], page 134.

\begin{verbatim}
\usepackage{color}
\newcounter{cnt}
\newcommand{\makeredlabel}[1]{\textcolor{red}{#1.}}
\newenvironment{redlabel}{\begin{list}{\arabic{cnt}}}{\end{list}}
\begin{redlabel}
\begin{arabic{cnt}}
\usecounter{cnt}
\setlength{\labelwidth}{0em}
\setlength{\labelsep}{0.5em}
\setlength{\leftmargin}{1.5em}
\setlength{\itemindent}{0.5em} \% equals \labelwidth+\labelsep
\let\makelabel=\makeredlabel
\end{arabic{cnt}}
\end{redlabel}
\end{verbatim}
8.16.1 \item: An entry in a list

Synopsis:

\item text of item

or

\item[optional-label] text of item

An entry in a list. The entries are prefixed by a label, whose default depends on the list type.

Because the optional label is surrounded by square brackets ‘[...]’, if you have an item whose text starts with [, you have to hide the bracket inside curly braces, as in: \item {[]} is an open square bracket; otherwise, \LaTeX will think it marks the start of an optional label.

Similarly, if the item does have the optional label and you need a close square bracket inside that label, you must hide it in the same way: \item[Close square bracket, {[]}]. See Section 2.5 [\LaTeX command syntax], page 7.

In this example the enumerate list has two items that use the default label and one that uses the optional label.

\begin{enumerate}
  \item Moe
  \item[sometimes] Shemp
  \item Larry
\end{enumerate}

The first item is labelled ‘1.’, the second item is labelled ‘sometimes’, and the third item is labelled ‘2.’. Because of the optional label in the second item, the third item is not labelled ‘3.’.

8.16.2 trivlist: A restricted form of list

Synopsis:

\begin{trivlist}
...
\end{trivlist}

A restricted version of the list environment, in which margins are not indented and an \item without an optional argument produces no text. It is most often used in macros, to define an environment where the \item command is part of the environment’s definition. For instance, the center environment is defined essentially like this:

\newenvironment{center}
{\begin{trivlist}\centering\item\relax}
{\end{trivlist}}

Using trivlist in this way allows the macro to inherit some common code: combining vertical space of two adjacent environments; detecting whether the text following the environment should be considered a new paragraph or a continuation of the previous one; adjusting the left and right margins for possible nested list environments.

Specifically, trivlist uses the current values of the list parameters (see Section 8.16 [list], page 65), except that \parsep is set to the value of \parskip, and \leftmargin, \labelwidth, and \itemindent are set to zero.
This example outputs the items as two paragraphs, except that (by default) they have no paragraph indent and are vertically separated.

\begin{trivlist}
\item The \textit{Surprise} is not old; no one would call her old.
\item She has a bluff bow, lovely lines.
\end{trivlist}

8.17 math

Synopsis:

\begin{math}
math
\end{math}

The math environment inserts given math material within the running text. \(...\) and \$...\$ are synonyms. See Chapter 16 [Math formulas], page 148.

8.18 minipage

Synopses:

\begin{minipage}{width}
contents
\end{minipage}

or

\begin{minipage}[position][height][inner-pos]{width}
contents
\end{minipage}

Put contents into a box that is width wide. This is like a small version of a page; it can contain its own footnotes, itemized lists, etc. (There are some restrictions, including that it cannot have floats.) This box will not be broken across pages. So minipage is similar to \parbox (see Section 20.3 [parbox], page 199) but here you can have paragraphs.

This example will be 3 inches wide, and has two paragraphs.

\begin{minipage}{3in}
Stephen Kleene was a founder of the Theory of Computation.

He was a student of Church, wrote three influential texts,
was President of the Association for Symbolic Logic,
and won the National Medal of Science.
\end{minipage}

See below for a discussion of the paragraph indent inside a minipage.

The required argument width is a rigid length (see Chapter 14 [Lengths], page 137). It gives the width of the box into which contents are typeset.

There are three optional arguments, position, height, and inner-pos. You need not include all three. For example, get the default position and set the height with \begin{minipage}{c}[2.54cm]{\columnwidth} contents \end{minipage}. (Get the natural height with an empty argument, \[\].)
The optional argument \textit{position} governs how the \texttt{minipage} vertically aligns with the surrounding material.

\begin{description}
\item[c] (synonym \texttt{m}) Default. Positions the \texttt{minipage} so its vertical center lines up with the center of the adjacent text line.
\item[t] Align the baseline of the top line in the \texttt{minipage} with the baseline of the surrounding text (plain \TeX's \texttt{vtop}).
\item[b] Align the baseline of the bottom line in the \texttt{minipage} with the baseline of the surrounding text (plain \TeX's \texttt{vbox}).
\end{description}

To see the effects of these, contrast running this
\begin{verbatim}
---\begin{minipage}\[c\]{0.25in}
first\\ second\\ third
\end{minipage}
\end{verbatim}
with the results of changing \texttt{c} to \texttt{b} or \texttt{t}.

The optional argument \textit{height} is a rigid length (see Chapter 14 [Lengths], page 137). It sets the height of the \texttt{minipage}. You can enter any value larger than, or equal to, or smaller than the \texttt{minipage}'s natural height and \LaTeX{} will not give an error or warning. You can also set it to a height of zero or a negative value.

The final optional argument \textit{inner-pos} controls the placement of \textit{contents} inside the box. These are the possible values are (the default is the value of \textit{position}).

\begin{description}
\item[t] Place \textit{contents} at the top of the box.
\item[c] Place it in the vertical center.
\item[b] Place it at the box bottom.
\item[s] Stretch \textit{contents} out vertically; it must contain vertically stretchable space.
\end{description}

The \textit{inner-pos} argument makes sense when the \textit{height} option is set to a value larger than the \texttt{minipage}'s natural height. To see the effect of the options, run this example with the various choices in place of \texttt{b}.

\begin{verbatim}
Text before
\begin{center}
---\begin{minipage}[c][3in][b]{0.25\textwidth}
first\\ second\\ third
\end{minipage}
\end{center}
Text after
\end{verbatim}

By default paragraphs are not indented in a \texttt{minipage}. Change that with a command such as \texttt{\setlength{\parindent}{1pc}} at the start of \textit{contents}.

Footnotes in a \texttt{minipage} environment are handled in a way that is particularly useful for putting footnotes in figures or tables. A \texttt{\footnote} or \texttt{\footnotetext} command puts the footnote at the bottom of the minipage instead of at the bottom of the page, and it uses the \texttt{\mpfootnote} counter instead of the ordinary \texttt{footnote} counter (see Chapter 13 [Counters], page 133).
This puts the footnote at the bottom of the table, not the bottom of the page.
\begin{center} % center the minipage on the line
\begin{minipage}{2.5in} % center the table inside the minipage
\begin{tabular}{ll}
\textsc{Monarch} & \textsc{Reign} \\
Elizabeth II & 63 years\footnote{to date} \\
Victoria & 63 years \\
George III & 59 years
\end{tabular}
\end{minipage}
\end{center}

If you nest minipages then there is an oddness when using footnotes. Footnotes appear at the bottom of the text ended by the next \end{minipage} which may not be their logical place.

This puts a table containing data side by side with a map graphic. They are vertically centered.

% siunitx to have the S column specifier, % which aligns numbers on their decimal point.
\usepackage{siunitx}
\newcommand*{%vcenteredhbox}{\begin{tabular}{@{}c@{}}#1\end{tabular}}
...
\begin{center}
vcenteredhbox{\includegraphics[width=0.3\textwidth]{nyc.png}}
\hspace{0.1\textwidth}
\begin{minipage}{0.5\textwidth}
\begin{tabular}{r|S}
\multicolumn{1}{r}{Borough} & \{Pop. (million)} \ \hline
The Bronx & 1.5 \\
Brooklyn & 2.6 \\
Manhattan & 1.6 \\
Queens & 2.3 \\
Staten Island & 0.5
\end{tabular}
\end{minipage}
\end{center}

8.19 picture

Synopses:
\begin{picture}(width,height)
Chapter 8: Environments

picture command
\end{picture}

or
\begin{picture}(width, height)(xoffset, yoffset)
picture command
\end{picture}

Where there may be any number of picture command’s.

An environment to create simple pictures containing lines, arrows, boxes, circles, and text. This environment is not obsolete, but new documents typically use much more powerful graphics creation systems, such as TikZ, PSTricks, MetaPost, or Asymptote. None of these are covered in this document; see CTAN.

To start, here’s an example showing the parallelogram law for adding vectors.

\setlength{\unitlength}{1cm}
\begin{picture}(6,6) % picture box will be 6cm wide by 6cm tall
  \put(0,0){\vector(2,1){4}} % for every 2 over this vector goes 1 up
  \put(2,1){\makebox(0,0)[l]{first leg}}
  \put(4,2){\vector(1,2){2}}
  \put(5,4){\makebox(0,0)[l]{second leg}}
  \put(0,0){\vector(1,1){6}}
  \put(3,3){\makebox(0,0)[r]{sum}}
\end{picture}

The picture environment has one required argument, a pair of positive real numbers (width, height). Multiply these by the value \unitlength to get the nominal size of the output, i.e. the space that \LaTeX reserves on the output page. This nominal size need not be how large the picture really is; \LaTeX will draw things from the picture outside the picture’s box.

This environment also has an optional argument (xoffset, yoffset). It is used to shift the origin. Unlike most optional arguments, this one is not contained in square brackets. As with the required argument, it consists of a pair of two real numbers, but these may also be negative or null. Multiply these by \unitlength to get the coordinates of the point at the lower-left corner of the picture.

For example, if \unitlength has been set to 1mm, the command

\begin{picture}(100,200)(10,20)
produces a box of width 100 millimeters and height 200 millimeters. The picture’s origin is the point (10mm,20mm) and so the lower-left corner is there, and the upper-right corner is at (110mm,220mm). When you first draw a picture you typically omit the optional argument, leaving the origin at the lower-left corner. If you then want to modify your picture by shifting everything, you can just add the appropriate optional argument.

Each picture command tells \LaTeX where to put something by providing its position. A position is a pair such as (2.4,-5) giving the x- and y-coordinates. A coordinate is a not a length, it is a real number (it may have a decimal point or a minus sign). It specifies a length in multiples of the unit length \unitlength, so if \unitlength has been set to 1cm, then the coordinate 2.54 specifies a length of 2.54 centimeters.
\LaTeX's default for $\text{\texttt{unitlength}}$ is 1pt. It is a rigid length (see Chapter 14 [Lengths], page 137). Change it with the $\text{\texttt{setlength}}$ command (see Section 14.2 [\texttt{setlength}], page 139). Make this change only outside of a \texttt{picture} environment.

The \texttt{picture} environment supports using standard arithmetic expressions as well as numbers.

Coordinates are given with respect to an origin, which is by default at the lower-left corner of the picture. Note that when a position appears as an argument, as with $\text{\texttt{put}(1,2)}\{\ldots\}$, it is not enclosed in braces since the parentheses serve to delimit the argument. Also, unlike in some computer graphics systems, larger y-coordinates are further up the page, for example, $y = 1$ is above $y = 0$.

There are four ways to put things in a picture: \texttt{put}, \texttt{multiput}, \texttt{qbezier}, and \texttt{graphpaper}. The most often used is \texttt{put}. This

\begin{verbatim}
\put(11.3,-0.3){...}
\end{verbatim}

places the object with its reference point at coordinates (11.3, −0.3). The reference points for various objects will be described below. The \texttt{put} command creates an LR box (see Chapter 17 [Modes], page 177). Anything that can go in an \texttt{mbox} (see Section 20.1 [\texttt{mbox} & \texttt{makebox}], page 197) can go in the text argument of the \texttt{put} command. The reference point will be the lower left corner of the box. In this picture

\begin{verbatim}
\setlength{\textwidth}{1cm}
...\begin{picture}(1,1)
 \put(0,0){\line(1,0){1}}
 \put(0,0){\line(1,1){1}}
 \end{picture}
\end{verbatim}

the three dots are just slightly left of the point of the angle formed by the two lines. (Also, \texttt{\line(1,1){1}} does not call for a line of length one; rather the line has a change in the x coordinate of 1.)

The \texttt{multiput}, \texttt{qbezier}, and \texttt{graphpaper} commands are described below.

You can also use this environment to place arbitrary material at an exact location. For example:

\begin{verbatim}
\usepackage{color,graphicx} \% in preamble
...
\begin{center}
\setlength{\textwidth}{\textwidth}
\begin{picture}(1,1) \% leave space, \textwidth wide and tall
 \put(0,0){\includegraphics[width=\textwidth]{desertedisland.jpg}}
 \put(0.25,0.35){\textcolor{red}{X Treasure here}}
\end{picture}
\end{center}
\end{verbatim}

The red X will be precisely a quarter of the \texttt{\textwidth} from the left margin, and 0.35\texttt{\textwidth} up from the bottom of the picture. Another example of this usage is to put similar code in the page header to get repeat material on each of a document’s pages.

\subsection{\texttt{put}}

Synopsis:

\begin{verbatim}
\put(xcoord,ycoord){content}
\end{verbatim}
Place content at the coordinate (xcoord, ycoord). See the discussion of coordinates and \unitlength in Section 8.19 [picture], page 73. The content is processed in LR mode (see Chapter 17 [Modes], page 177) so it cannot contain line breaks.

This includes the text into the picture.

\put(4.5,2.5){Apply the \textit{unpoke} move}

The reference point, the location (4.5,2.5), is the lower left of the text, at the bottom left of the ‘A’.

8.19.2 \multiput

Synopsis:
\multiput(x,y)(\textit{delta}_x,\textit{delta}_y){\textit{num-copies}}{\textit{obj}}

Copy obj a total of \textit{num-copies} times, with an increment of \textit{delta}_x,\textit{delta}_y. The obj first appears at position \((x,y)\), then at \((x + \delta_x, y + \delta_y)\), and so on.

This draws a simple grid with every fifth line in bold (see also Section 8.19.4 \[graphpaper\], page 77).

\begin{verbatim}
\begin{picture}(10,10)
   \linethickness{0.05mm}
   \multiput(0,0)(1,0){10}{\line(0,1){10}}
   \multiput(0,0)(0,1){10}{\line(1,0){10}}
   \linethickness{0.5mm}
   \multiput(0,0)(5,0){3}{\line(0,1){10}}
   \multiput(0,0)(0,5){3}{\line(1,0){10}}
\end{picture}
\end{verbatim}

8.19.3 \qbezier

Synopsis:
\qbezier(x_1,y_1)(x_2,y_2)(x_3,y_3)
\qbezier[\textit{num}](x_1,y_1)(x_2,y_2)(x_3,y_3)

Draw a quadratic Bezier curve whose control points are given by the three required arguments \((x_1,y_1)\), \((x_2,y_2)\), and \((x_3,y_3)\). That is, the curve runs from \((x_1,y_1)\) to \((x_3,y_3)\), is quadratic, and is such that the tangent line at \((x_1,y_1)\) passes through \((x_2,y_2)\), as does the tangent line at \((x_3,y_3)\).

This draws a curve from the coordinate \((1,1)\) to \((1,0)\).

\qbezier(1,1)(1.25,0.75)(1,0)

The curve’s tangent line at \((1,1)\) contains \((1.25,0.75)\), as does the curve’s tangent line at \((1,0)\).

The optional argument \textit{num} gives the number of calculated intermediate points. The default is to draw a smooth curve whose maximum number of points is \qbeziermax (change this value with \renewcommand).

This draws a rectangle with a wavy top, using \qbezier for that curve.

\begin{verbatim}
\begin{picture}(8,4)
   \put(0,0){\vector(1,0){8}} \% x axis
   \put(0,0){\vector(0,1){4}} \% y axis
\end{picture}
\end{verbatim}
8.19.4 \texttt{\texttt{graphpaper}}

Synopsis:

\begin{verbatim}
\graphpaper(x_{init},y_{init})(x_{dimen},y_{dimen})
\graphpaper[spacing](x_{init},y_{init})(x_{dimen},y_{dimen})
\end{verbatim}

Draw a coordinate grid. Requires the \texttt{graphpap} package. The grid’s origin is \((x_{init},y_{init})\). Grid lines come every \texttt{spacing} units (the default is 10). The grid extends \texttt{x_dimen} units to the right and \texttt{y_dimen} units up. All arguments must be positive integers.

This makes a grid with seven vertical lines and eleven horizontal lines.

\begin{verbatim}
\usepackage{graphpap} % in preamble
... 
\begin{picture}(6,20) % in document body
\graphpaper[2](0,0)(12,20)
\end{picture}
\end{verbatim}

The lines are numbered every ten units.

8.19.5 \texttt{\texttt{\texttt{line}}}

Synopsis:

\begin{verbatim}
\line(x_{run},y_{rise})\{travel\}
\end{verbatim}

Draw a line. It slopes such that it vertically rises \texttt{y_rise} for every horizontal \texttt{x_run}. The \texttt{travel} is the total horizontal change—it is not the length of the vector, it is the change in \texttt{x}. In the special case of vertical lines, where \((x_{run},y_{rise})=(0,1)\), the \texttt{travel} gives the change in \texttt{y}.

This draws a line starting at coordinates (1,3).

\begin{verbatim}
\put(1,3){\line(2,5){4}}
\end{verbatim}

For every over 2, this line will go up 5. Because \texttt{travel} specifies that this goes over 4, it must go up 10. Thus its endpoint is \( (1,3) + (4,10) = (5,13) \). In particular, note that \texttt{travel} = 4 is not the length of the line, it is the change in \texttt{x}.

The arguments \texttt{x_run} and \texttt{y_rise} are integers that can be positive, negative, or zero. (If both are 0 then \LaTeX{} treats the second as 1.) With \begin{verbatim}\put(x_{init},y_{init})\{\line(x_{run},y_{rise})\{\texttt{travel}\}\end{verbatim}, if \texttt{x_run} is negative then the line’s ending point has a first coordinate that is less than \texttt{x_init}. If \texttt{y_rise} is negative then the line’s ending point has a second coordinate that is less than \texttt{y_init}.

If \texttt{travel} is negative then you get LaTeX Error: Bad \texttt{\texttt{line}} or \texttt{\texttt{vector}} argument.

Standard \LaTeX{} can only draw lines with a limited range of slopes because these lines are made by putting together line segments from pre-made fonts. The two numbers \texttt{x_run} and
y_rise must have integer values from \(-6\) through \(6\). Also, they must be relatively prime, so that \((x\_run,y\_rise)\) can be \((2,1)\) but not \((4,2)\) (if you choose the latter then instead of lines you get sequences of arrowheads; the solution is to switch to the former). To get lines of arbitrary slope and plenty of other shapes in a system like picture, see the package pict2e (https://ctan.org/pkg/pict2e). Another solution is to use a full-featured graphics system such as TikZ, PSTricks, MetaPost, or Asymptote.

8.19.6 \texttt{\linethickness}

Synopsis:

\begin{verbatim}
\linethickness{dim}
\end{verbatim}

Declares the thickness of subsequent horizontal and vertical lines in a picture to be \(dim\), which must be a positive length (see Chapter 14 \texttt{[Lengths]}, page 137). It differs from \texttt{\thinlines} and \texttt{\thicklines} in that it does not affect the thickness of slanted lines, circles, or ovals (see Section 8.19.10 \texttt{[\oval]}, page 78).

8.19.7 \texttt{\thinlines}

Declaration to set the thickness of subsequent lines, circles, and ovals in a picture environment to be \(0.4\) pt. This is the default thickness, so this command is unnecessary unless the thickness has been changed with either Section 8.19.6 \texttt{[\linethickness]}, page 78, or Section 8.19.8 \texttt{[\thicklines]}, page 78.

8.19.8 \texttt{\thicklines}

Declaration to set the thickness of subsequent lines, circles, and ovals in a picture environment to be \(0.8\) pt. See also Section 8.19.6 \texttt{[\linethickness]}, page 78, and Section 8.19.7 \texttt{[\thinlines]}, page 78. This command is illustrated in the Trapezoidal Rule example of Section 8.19.3 \texttt{[\qbezier]}, page 76.

8.19.9 \texttt{\circle}

Synopsis:

\begin{verbatim}
\circle{diameter}
\circle*[diameter]
\end{verbatim}

Produces a circle with a diameter as close as possible to the specified one. The \texttt{*} form produces a filled-in circle.

This draws a circle of radius \(6\), centered at \((5,7)\).

\begin{verbatim}
\put(5,7){\circle{6}}
\end{verbatim}

The available radii for \texttt{\circle} are, in points, the even numbers from \(2\) to \(20\), inclusive. For \texttt{\circle*} they are all the integers from \(1\) to \(15\).

8.19.10 \texttt{\oval}

Synopsis:

\begin{verbatim}
\oval(width,height)
\oval(width,height)[portion]
\end{verbatim}

Produce a rectangle with rounded corners, hereinafter referred to as an oval. The optional argument \texttt{portion} allows you to produce only half or a quarter of the oval. For half an oval take \texttt{portion} to be one of these.
Produce only one quarter of the oval by setting portion to tr, br, bl, or tl.
This draws the top half of an oval that is 3 wide and 7 tall.
\put(5,7){\oval(3,7)[t]}
The (5,7) is the center of the entire oval, not just the center of the top half.
These shapes are not ellipses. They are rectangles whose corners are made with quarter circles. These circles have a maximum radius of 20 pt (see Section 8.19.9 \[circle], page 78, for the sizes). Thus large ovals are just frames with a small amount of corner rounding.

8.19.11 \shortstack
Synopsis:
\shortstack[position]{line 1 \ ... }
Produce a vertical stack of objects.
This labels the y axis by writing the word ‘y’ above the word ‘axis’.
\setlength{\unitlength}{1cm}
\begin{picture}(5,2.5)(-0.75,0)
\put(0,0){\vector(1,0){4}} % x axis
\put(0,0){\vector(0,1){2}} % y
\put(-0.2,2){\makebox(0,0)[r]{\shortstack{$y$\ axis}}}
\end{picture}
For a short stack, the reference point is the lower left of the stack. In the above example the \makebox (see Section 20.1 \mbox & \makebox, page 197) puts the stack flush right in a zero width box so in total the short stack sits slightly to the left of the y axis.
The valid positions are:
  r   Make objects flush right
  l   Make objects flush left
  c   Center objects (default)
Separate objects into lines with \. These stacks are short in that, unlike in a \texttt{tabular} or \texttt{array} environment, here the rows are not spaced out to be of even baseline skips. Thus, in \texttt{\shortstack{X\o\o\X}} the first and last rows are taller than the middle two, and therefore the baseline skip between the two middle rows is smaller than that between the third and last row. You can adjust row heights and depths either by putting in the usual interline spacing with \texttt{\shortstack{X\strut o\o\X}} (see Section 19.13 \strut, page 193), or explicitly, via an zero-width box \texttt{\shortstack{X \rule{0pt}{12pt} o\o\X}} or by using \texttt{\}'s optional argument \texttt{\shortstack{X\[2pt] o\o\X}}.
The \texttt{\shortstack} command is also available outside the \texttt{picture} environment.
8.19.12 \texttt{\textbackslash vector}

Synopsis:

\texttt{\textbackslash vector(x\_run,y\_rise)\{travel\}}

Draw a line ending in an arrow. The slope of that line is: it vertically rises $y\_rise$ for every horizontal $x\_run$. The \textit{travel} is the total horizontal change—it is not the length of the vector, it is the change in $x$. In the special case of vertical vectors, if $(x\_run,y\_rise) = (0,1)$, then \textit{travel} gives the change in $y$.

For an example see Section 8.19 [picture], page 73.

For elaboration on $x\_run$ and $y\_rise$ see Section 8.19.5 [line], page 77. As there, the values of $x\_run$ and $y\_rise$ are limited. For \texttt{\textbackslash vector} you must choose integers between $-4$ and $4$, inclusive. Also, the two you choose must be relatively prime. Thus, \texttt{\vector(2,1)\{4\}} is acceptable but \texttt{\vector(4,2)\{4\}} is not (if you use the latter then you get a sequence of arrowheads).

8.19.13 \texttt{\textbackslash makebox (picture)}

Synopsis:

\texttt{\makebox\{rect-width,rec-height\}\{text\}}

\texttt{\makebox\{rect-width,rec-height\}[position]\{text\}}

Make a box to hold \textit{text}. This command fits with the picture environment, although you can use it outside of there, because \texttt{rec-width} and \texttt{rec-height} are numbers specifying distances in terms of the \texttt{unitlength} (see Section 8.19 [picture], page 73). This command is similar to the normal \texttt{\makebox} command (see Section 20.1 [mbox & \makebox], page 197) except here that you must specify the width and height. This command is fragile (see Section 12.11 [\protect], page 123).

This makes a box of length 3.5 times \texttt{unitlength} and height 4 times \texttt{unitlength}.

\texttt{\put(1,2)\{\makebox\{3.5,4\}\{...\}}

The optional argument \texttt{position} specifies where in the box the \textit{text} appears. The default is to center it, both horizontally and vertically. To place it somewhere else, use a string with one or two of these letters.

- \texttt{t} Puts \textit{text} the top of the box.
- \texttt{b} Put \textit{text} at the bottom.
- \texttt{l} Put \textit{text} on the left.
- \texttt{r} Put \textit{text} on the right.

8.19.14 \texttt{\textbackslash framebox (picture)}

Synopsis:

\texttt{\framebox\{rect-width,rec-height\}\{text\}}

\texttt{\framebox\{rect-width,rec-height\}[position]\{text\}}

This is the same as Section 8.19.13 [\makebox (picture)], page 80, except that it puts a frame around the outside of the box that it creates. The reference point is the bottom left corner of the frame. This command fits with the picture environment, although you can use it outside of there, because lengths are numbers specifying the distance in terms
This example creates a frame 2.5 inches by 3 inches and puts the text in the center.
\begin{verbatim}
\setlength{\unitlength}{1in}
\framebox(2.5,3){test text}
\end{verbatim}

The required arguments are that the rectangle has overall width \textit{rect-width} units and height \textit{rect-height} units.

The optional argument \textit{position} specifies the position of \textit{text}; see Section 8.19.13 [\makebox (picture)], page 80, for the values that it can take.

The rule has thickness \textit{\fboxrule} and there is a blank space \textit{\fboxsep} between the frame and the contents of the box.

For this command, you must specify the \textit{width} and \textit{height}. If you want to just put a frame around some contents whose dimension is determined in some other way then either use \texttt{\fbox} (see Section 20.2 [\fbox \& \framebox], page 198) or \texttt{\frame} (see Section 8.19.15 [\frame], page 81).

\subsection{8.19.15 \frame}

Synopsis:
\begin{verbatim}
\frame{contents}
\end{verbatim}

Puts a rectangular frame around \textit{contents}. The reference point is the bottom left corner of the frame. In contrast to \texttt{\framebox} (see Section 8.19.14 [\framebox (picture)], page 80), this command puts no extra space between the frame and the object. It is fragile (see Section 12.11 [\protect], page 123).

\subsection{8.19.16 \dashbox}

Synopsis:
\begin{verbatim}
\dashbox{dash-len}(rect-width,rect-height){text}
\dashbox{dash-len}(rect-width,rect-height)[position]{text}
\end{verbatim}

Create a dashed rectangle around \textit{text}. This command fits with the \texttt{picture} environment, although you can use it outside of there, because lengths are numbers specifying the distance in terms of the \texttt{\unitlength} (see Section 8.19 [\unitlength], page 73).

The required arguments are: dashes are \textit{dash-len} units long, with the same length gap, and the rectangle has overall width \textit{rect-width} units and height \textit{rect-height} units.

The optional argument \textit{position} specifies the position of \textit{text}; see Section 8.19.13 [\makebox (picture)], page 80, for the values that it can take.

This shows that you can use non-integer value for \textit{dash-len}.
\begin{verbatim}
\put(0,0){\dashbox{0.1}(5,0.5){My hovercraft is full of eels.}}
\end{verbatim}

Each dash will be $0.1\unitlength$ long, the box’s width is $5\unitlength$ and its height is $0.5\unitlength$.

As in that example, a dashed box looks best when \textit{rect-width} and \textit{rect-height} are multiples of the \textit{dash-len}. 

of the \texttt{\unitlength} (see Section 8.19 [\unitlength], page 73). This command is fragile (see Section 12.11 [\protect], page 123).

This example creates a frame 2.5 inches by 3 inches and puts the text in the center.
\begin{verbatim}
\setlength{\unitlength}{1in}
\framebox(2.5,3){test text}
\end{verbatim}

The required arguments are that the rectangle has overall width \textit{rect-width} units and height \textit{rect-height} units.

The optional argument \textit{position} specifies the position of \textit{text}; see Section 12.11 [\protect], page 123).
8.20 quotation & quote

Synopsis:
\begin{quotation}
text
\end{quotation}
or
\begin{quote}
text
\end{quote}

Include a quotation. Both environments indent margins on both sides by \leftmargin and the text is right-justified.

They differ in how they treat paragraphs. In the \texttt{quotation} environment, paragraphs are indented by 1.5 em and the space between paragraphs is small, 0pt plus 1pt. In the \texttt{quote} environment, paragraphs are not indented and there is vertical space between paragraphs (it is the rubber length \parsep).

\begin{quotation}
\begin{small}\it
Four score and seven years ago
... shall not perish from the earth.
\end{small}\it
\hspace{1em plus 1fill}---Abraham Lincoln
\end{quotation}

8.21 tabbing

Synopsis:
\begin{tabbing}
row1col1 \= row1col2 ... \\ \\
row2col1 \> row2col2 ... \\ \\
... \end{tabbing}

Align text in columns, by setting tab stops and tabbing to them much as was done on a typewriter. This is less often used than the environments \texttt{tabular} (see Section 8.23 \texttt{tabular}, page 85) or \texttt{array} (see Section 8.2 \texttt{array}, page 53) because in those the width of each column need not be constant and need not be known in advance.

This example has a first line where the tab stops are set to explicit widths, ended by a \texttt{\kill} command (which is described below):

\begin{tabbing}
\hspace{1.2in}\=\hspace{1in}\=
\end{tabbing}

\hspace{1.2in}\=\hspace{1in}\=
\texttt{Ship} \>\texttt{Guns} \>\texttt{Year} \\ \\
\texttt{Ship} \>14 \>1800 \\ \\
\texttt{Ship} \>24 \>1803 \\ \\
\texttt{Ship} \>38 \>1804 \\ \\
\texttt{Ship} \>28 \>1805 \\ \\
\end{tabbing}

Both the \texttt{tabbing} environment and the more widely-used \texttt{tabular} environment put text in columns. The most important distinction is that in \texttt{tabular} the width of columns is
determined automatically by \LaTeX, while in \texttt{tabbing} the user sets the tab stops. Another distinction is that \texttt{tabular} generates a box, but \texttt{tabbing} can be broken across pages. Finally, while \texttt{tabular} can be used in any mode, \texttt{tabbing} can be used only in paragraph mode and it always starts a new paragraph, without indentation.

Moreover, as shown in the example above, there is no need to use the starred form of the \texttt{\hspace} command at the beginning of a tabbed row. The right margin of the \texttt{tabbing} environment is the end of line, so that the width of the environment is $\texttt{\linewidth}$.

The \texttt{tabbing} environment contains a sequence of \textit{tabbed rows}. The first tabbed row begins immediately after \texttt{\begin{tabbing}} and each row ends with $\texttt{\} or \texttt{\kill}$. The last row may omit the $\texttt{\} and end with just $\texttt{\end{tabbing}}$.

At any point the \texttt{tabbing} environment has a \textit{current tab stop pattern}, a sequence of $n > 0$ tab stops, numbered 0, 1, etc. These create $n$ corresponding columns. Tab stop 0 is always the left margin, defined by the enclosing environment. Tab stop number $i$ is set if it is assigned a horizontal position on the page. Tab stop number $i$ can only be set if all the stops 0, $\ldots$, $i - 1$ have already been set; normally later stops are to the right of earlier ones.

By default any text typeset in a \texttt{tabbing} environment is typeset ragged right and left-aligned on the current tab stop. Typesetting is done in LR mode (see Chapter 17 [Modes], page 177).

The following commands can be used inside a \texttt{tabbing} environment. They are all fragile (see Section 12.11 [\texttt{\protect}], page 123).

$\backslash\backslash$ (tabbing)

End a tabbed line and typeset it.

$\texttt{\=}$(tabbing)

Sets a tab stop at the current position.

$\texttt{\>}$ (tabbing)

Advances to the next tab stop.

$\texttt{\<}$

Put following text to the left of the local margin (without changing the margin). Can only be used at the start of the line.

$\texttt{\+}$

Moves the left margin of the next and all the following commands one tab stop to the right, beginning tabbed line if necessary.

$\texttt{\-}$

Moves the left margin of the next and all the following commands one tab stop to the left, beginning tabbed line if necessary.

$\texttt{\'} $ (tabbing)

Moves everything that you have typed so far in the current column, i.e., everything from the most recent $\texttt{\>}$, $\texttt{\<}$, $\texttt{\'}$, $\texttt{\} or \texttt{\kill}$ command, to the previous column and aligned to the right, against the current column’s tab stop.

$\texttt{\`}$ (tabbing)

Allows you to put text flush right against any tab stop, including tab stop 0. However, it can’t move text to the right of the last column because there’s no tab stop there. The $\texttt{\`}$ command moves all the text that follows it, up to the $\texttt{\} or \texttt{\end{tabbing}}$ command that ends the line, to the right margin of the \texttt{tabbing} environment. There must be no $\texttt{\>}$ or $\texttt{\'}$ command between the $\texttt{\`}$ and the $\texttt{\} or \texttt{\end{tabbing}}$ command that ends the line.
In a `tabbing` environment, the commands `\=`, `\'` and `\` do not produce accents as usual (see Section 23.5 [Accents], page 224). Instead, use the commands `\a=`, `\a'` and `\a`.

`\kill` Sets tab stops without producing text. Works just like `\` except that it throws away the current line instead of producing output for it. Any `\=`, `\+` or `\-` commands in that line remain in effect.

`\poptabs` Restores the tab stop positions saved by the last `\pushtabs`.

`\pushtabs` Saves all current tab stop positions. Useful for temporarily changing tab stop positions in the middle of a `tabbing` environment.

`\tabbingsep` Distance of the text moved by `\` to left of current tab stop.

This example typesets a Pascal function:

```latex
\begin{tabbing}
function \= fact(n \ : integer) \ : integer;\\
\> begin \= \+ \\
\> if \= n \> > 1 then \+ \\
\> \> fact := n \* fact(n-1) \- \\
\> \> \> else \+ \- \\
\> \> \> \> fact := 1; \-\- \\
\end{tabbing}
```

The output looks like this.

```
function fact(n : integer) : integer;
begin
  if n > 1 then
    fact := n * fact(n-1);
  else
    fact := 1;
end;
```

This example is just for illustration of the environment. To actually typeset computer code in typewriter like this, a verbatim environment (see Section 8.27 [verbatim], page 96) would normally be best. For pretty-printed code, there are quite a few packages, including `algorithm2e`, `fancyvrb`, `listings`, and `minted`.

### 8.22 table

Synopsis:

```latex
\begin{table}[placement]
  \tablebody
  \caption[loftitle]{title} \% optional
  \label{label} \% also optional
\end{table}
```
A class of floats (see Section 5.7 [Floats], page 32). They cannot be split across pages and so they are not typeset in sequence with the normal text but instead are floated to a convenient place, such as the top of a following page.

This example table environment contains a tabular

\begin{table}
\centering\small
\begin{tabular}{ll}
\multicolumn{1}{c}{\textit{Author}} & \multicolumn{1}{c}{\textit{Piece}} \\
Bach & Cello Suite Number 1 \\
Beethoven & Cello Sonata Number 3 \\
Brahms & Cello Sonata Number 1
\hline
\end{tabular}
\caption{Top cello pieces}
\label{tab:cello}
\end{table}

but you can put many different kinds of content in a table: the table body may contain text, \LaTeX \ commands, graphics, etc. It is typeset in a parbox of width \textwidth.

For the possible values of placement and their effect on the float placement algorithm, see Section 5.7 [Floats], page 32.

The label is optional; it is used for cross references (see Chapter 7 [Cross references], page 48). The \caption command is also optional. It specifies caption text title for the table (see Section 5.7.1 [caption], page 34). By default it is numbered. If its optional \texttt{listtitle} is present then that text is used in the list of tables instead of title (see Section 25.1 [Table of contents etc.], page 233).

In this example the table and caption will float to the bottom of a page, unless it is pushed to a float page at the end.

\begin{table}[b]
\centering
\begin{tabular}{r|p{2in}}
\hline
One & The loneliest number \hline
Two & Can be as sad as one. It's the loneliest number since the number one.
\end{tabular}
\caption{Cardinal virtues}
\label{tab:CardinalVirtues}
\end{table}

8.23 tabular

Synopsis:

\begin{tabular}{pos}{cols}
\texttt{column 1 entry} & \texttt{column 2 entry} & \ldots & \texttt{column n entry} \\
\ldots
\end{tabular}
Produce a table, a box consisting of a sequence of horizontal rows. Each row consists of items that are aligned vertically in columns. This illustrates many of the features.

\begin{tabular}{l|l}
| \textit{Player name} & \textit{Career home runs} |
|---------------------|-------------------------|
| Hank Aaron          & 755                     |
| Babe Ruth           & 714                     |
\end{tabular}

The output will have two left-aligned columns with a vertical bar between them. This is specified in \texttt{tabular}'s argument \{l|l\}. Put the entries into different columns by separating them with an ampersand, &. The end of each row is marked with a double backslash,\. Put a horizontal rule below a row, after a double backslash, with \texttt{\hline}. After the last row the \textbackslash is optional, unless an \texttt{\hline} command follows to put a rule below the table.

The required and optional arguments to \texttt{tabular} consist of:

\textbf{pos} Optional. Specifies the table’s vertical position. The default is to align the table so its vertical center matches the baseline of the surrounding text. There are two other possible alignments: t aligns the table so its top row matches the baseline of the surrounding text, and b aligns on the bottom row. This only has an effect if there is other text. In the common case of a \texttt{tabular} alone in a \texttt{center} environment this option makes no difference.

\textbf{cols} Required. Specifies the formatting of columns. It consists of a sequence of the following specifiers, corresponding to the types of column and intercolumn material.

l A column of left-aligned items.

r A column of right-aligned items.

c A column of centered items.

A vertical line the full height and depth of the environment.

@{text or space} Insert text or space at this location in every row. The text or space material is typeset in LR mode. This text is fragile (see Section 12.11 \protect, page 123).

If between two column specifiers there is no @-expression then \LaTeX’s \texttt{book}, \texttt{article}, and \texttt{report} classes will put on either side of each column a space of width \texttt{\tabcolsep}, which by default is 6 pt. That is, by default adjacent columns are separated by 12 pt (so \texttt{\tabcolsep} is misleadingly named since it is only half of the separation between tabular columns). In addition, a space of
\tabcolsep also comes before the first column and after the final column, unless you put a @\{\ldots\} there.

If you override the default and use an @-expression then \LaTeX{} does not insert \tabcolsep so you must insert any desired space yourself, as in @\{\hspace{1em}\}.

An empty expression @\{} will eliminate the space. In particular, sometimes you want to eliminate the space before the first column or after the last one, as in the example below where the tabular lines need to lie on the left margin.

\begin{flushleft}
\begin{tabular}{@{}l}
\ldots
\end{tabular}
\end{flushleft}

The next example shows text, a decimal point between the columns, arranged so the numbers in the table are aligned on it.

\begin{tabular}{r@{$.$}l}
\$3$ & \$14$\\
\$9$ & \$80665$
\end{tabular}

An \extracolsep\{wd\} command in an @-expression causes an extra space of width \texttt{wd} to appear to the left of all subsequent columns, until countermanded by another \extracolsep. Unlike ordinary intercolumn space, this extra space is not suppressed by an @-expression. An \extracolsep command can be used only in an @-expression in the \texttt{cols} argument. Below, \LaTeX{} inserts the right amount of intercolumn space to make the entire table 4 inches wide.

\begin{tabular*}{4in}{l@{\extracolsep{\fill}}l}
Seven times down, eight times up \ldots\\
& such is life!
\end{tabular*}

To insert commands that are automatically executed before a given column, load the \texttt{array} package and use the >\{\ldots\} specifier.

p\{wd\}

Each item in the column is typeset in a parbox of width \texttt{wd}, as if it were the argument of a \parbox[t]{\texttt{wd}\{\ldots\}} command.

A line break double backslash $\backslash\backslash$ may not appear in the item, except inside an environment like \texttt{minipage}, \texttt{array}, or \texttt{tabular}, or inside an explicit \parbox, or in the scope of a \texttt{centering}, \texttt{raggedright}, or \texttt{raggedleft} declaration (when used in a p-column element these declarations must appear inside braces, as with {\texttt{centering \ldots \backslash\backslash}}). Otherwise \LaTeX{} will misinterpret the double backslash as ending the \texttt{tabular} row. Instead, to get a line break in there use \texttt{\newline} (see Section 9.3 [\texttt{\newline}], page 101).
\*{num}{cols}
Equivalent to \texttt{num} copies of \texttt{cols}, where \texttt{num} is a positive integer and \texttt{cols} is a list of specifiers. Thus the specifier \texttt{\begin{tabular}{|*{3}{|l|r}|}} is equivalent to the specifier \texttt{\begin{tabular}{|l|r|l|r|}}}. Note that \texttt{cols} may contain another \*\-expression.

\textit{width} Required for \texttt{tabular*}, not allowed for \texttt{tabular}. Specifies the width of the \texttt{tabular*} environment. The space between columns should be rubber, as with @{\extracolsep{\fill}}, to allow the table to stretch or shrink to make the specified width, or else you are likely to get the \texttt{Underfull hbox (badness 10000) in alignment ...} warning.

Parameters that control formatting:

\texttt{\arrayrulewidth}
A length that is the thickness of the rule created by |, \texttt{\hline}, and \texttt{\vline} in the \texttt{tabular} and \texttt{array} environments. The default is ‘.4pt’. Change it as in \texttt{\setlength{\arrayrulewidth}{0.8pt}}.

\texttt{\arraystretch}
A factor by which the spacing between rows in the \texttt{tabular} and \texttt{array} environments is multiplied. The default is ‘1’, for no scaling. Change it as \texttt{\renewcommand{\arraystretch}{1.2}}.

\texttt{\doublerulesep}
A length that is the distance between the vertical rules produced by the \texttt{||} specifier. The default is ‘2pt’.

\texttt{\tabcolsep}
A length that is half of the space between columns. The default is ‘6pt’. Change it with \texttt{\setlength}.

The following commands can be used inside the body of a \texttt{tabular} environment, the first two inside an entry and the second two between lines:

\textbf{8.23.1 \texttt{\multicolumn}}

Synopsis:

\texttt{\multicolumn\{numcols\}\{cols\}\{text\}}
Make an \texttt{array} or \texttt{tabular} entry that spans several columns. The first argument \texttt{numcols} gives the number of columns to span. The second argument \texttt{cols} specifies the formatting of the entry, with \texttt{c} for centered, \texttt{l} for flush left, or \texttt{r} for flush right. The third argument \texttt{text} gives the contents of that entry.

In this example, in the first row, the second and third columns are spanned by the single heading ‘Name’.

\texttt{\begin{tabular}{|lcc|}
\texttt{\textit{ID}} &\multicolumn{2}{c}{\texttt{\textit{Name}}} &\texttt{\textit{Age}} \\
\hline
978-0-393-03701-2 &O’Brian &Patrick &55 \\
...
\end{tabular}

What counts as a column is: the column format specifier for the \texttt{array} or \texttt{tabular} environment is broken into parts, where each part (except the first) begins with \texttt{l}, \texttt{c}, \texttt{r}, or \texttt{p}. So from \texttt{\begin{tabular}{|r|ccp{1.5in}|} \texttt{\begin{tabular}{|r\texttt{|}c\texttt{|}c\texttt{|}r\texttt{|}p{1.5in}\texttt{|}} \texttt{\end{tabular}}}} the parts are \texttt{|r|}, \texttt{c}, \texttt{c}, and \texttt{p(1.5in)}.

The \texttt{cols} argument overrides the \texttt{array} or \texttt{tabular} environment’s intercolumn area default adjoining this multicolumn entry. To affect that area, this argument can contain vertical bars \texttt{|} indicating the placement of vertical rules, and \texttt{@\{\ldots\}} expressions. Thus if \texttt{cols} is \texttt{‘|c|’} then this multicolumn entry will be centered and a vertical rule will come in the intercolumn area before it and after it. This table details the exact behavior.

\begin{tabular}{|cc|c|c|}
\multicolumn{1}{r}{w} % entry one \\
&\multicolumn{1}{|r|}{x} % entry two \\
&\multicolumn{1}{|r}{y} % entry three \\
\&z % entry four \\
\end{tabular}

Before the first entry the output will not have a vertical rule because the \texttt{\multicolumn} has the \texttt{cols} specifier \texttt{‘r’} with no initial vertical bar. Between entry one and entry two there will be a vertical rule; although the first \texttt{cols} does not have an ending vertical bar, the second \texttt{cols} does have a starting one. Between entry two and entry three there is a single vertical rule; despite that the \texttt{cols} in both of the surrounding \texttt{\multicolumn}’s call for a vertical rule, you only get one rule. Between entry three and entry four there is no vertical rule; the default calls for one but the \texttt{cols} in the entry three \texttt{\multicolumn} leaves it out, and that takes precedence. Finally, following entry four there is a vertical rule because of the default.

The number of spanned columns \texttt{numcols} can be \texttt{1}. Besides giving the ability to change the horizontal alignment, this also is useful to override for one row the \texttt{tabular} definition’s default intercolumn area specification, including the placement of vertical rules.

In the example below, in the \texttt{tabular} definition the first column is specified to default to left justified but in the first row the entry is centered with \texttt{\multicolumn{1}{c}{\textsc{Period}}}. Also in the first row, the second and third columns are spanned by a single entry with \texttt{\multicolumn{2}{c}{\textsc{Span}}}, overriding the specification to center those two columns on the page range endash.

\begin{tabular}{l|r@{--}l}
\multicolumn{1}{c}{\textsc{Period}} &
\multicolumn{2}{c}{\textsc{Span}} \hline
Baroque & 1600 & 1760 \\ Classical & 1730 & 1820 \\ Romantic & 1780 & 1910 \\ Impressionistic & 1875 & 1925
\end{tabular}

Although the \texttt{tabular} specification by default puts a vertical rule between the first and second columns, no such vertical rule appears in the first row here. That’s because there is no vertical bar in the \texttt{cols} part of the first row’s first \texttt{\multicolumn} command.
8.23.2 \vline

Draw a vertical line in a tabular or array environment extending the full height and depth of an entry's row. Can also be used in an @-expression, although its synonym vertical bar | is more common. This command is rarely used in the body of a table; typically a table’s vertical lines are specified in tabular's cols argument and overridden as needed with \multicolumn (see Section 8.23 [tabular], page 85).

The example below illustrates some pitfalls. In the first row’s second entry the \hfill moves the \vline to the left edge of the cell. But that is different than putting it halfway between the two columns, so between the first and second columns there are two vertical rules, with the one from the \{c|cc} specifier coming before the one produced by the \vline\hfill. In contrast, the first row’s third entry shows the usual way to put a vertical bar between two columns. In the second row, the ghi is the widest entry in its column so in the \vline\hfill the \hfill has no effect and the vertical line in that entry appears immediately next to the g, with no whitespace.

\begin{tabular}{c|cc}
  x &\vline\hfill y &\multicolumn{1}{|r}{z} \\
  \multicolumn{1}{c|}{abc} &\hfill def &\vline\hfill ghi
\end{tabular}

8.23.3 \cline

Synopsis:

\cline{i-j}

In an array or tabular environment, draw a horizontal rule beginning in column i and ending in column j. The dash, -, must appear in the mandatory argument. To span a single column use the number twice, as with \cline{2-2}.

This example puts two horizontal lines between the first and second rows, one line in the first column only, and the other spanning the third and fourth columns. The two lines are side-by-side, at the same height.

\begin{tabular}{llrr}
  a &b &c &d \\
  \cline{1-1} \cline{3-4}
  e &f &g &h
\end{tabular}

8.23.4 \hline

Draw a horizontal line the width of the enclosing tabular or array environment. It’s most commonly used to draw a line at the top, bottom, and between the rows of a table.

In this example the top of the table has two horizontal rules, one above the other, that span both columns. The bottom of the table has a single rule spanning both columns. Because of the \hline, the tabular second row’s line ending double backslash \ is required.

\begin{tabular}{ll}
  \hline\hline
  Baseball \&Red Sox \\
  Basketball \&Celtics \hline
\end{tabular}
8.24 thebibliography

Synopsis:

\begin{thebibliography}{widest-label}
\bibitem[cite_key]
... \\
end{thebibliography}

Produce a bibliography or reference list. There are two ways to produce bibliographic lists. This environment is suitable when you have only a few references and can maintain the list by hand. See Section 8.24.4 [Using BibTeX], page 93, for a more sophisticated approach.

This shows the environment with two entries.

This work is based on \cite{latexdps}.
Together they are \cite{latexdps, texbook}.
... \\
\begin{thebibliography}{9}
\bibitem{latexdps}
Leslie Lamport. \\
\textit{\LaTeX{}: a document preparation system}.  \\
Addison-Wesley, Reading, Massachusetts, 1993.
\bibitem{texbook}
Donald Ervin Knuth. \\
\textit{The \TeX{} book}.  \\
Addison-Wesley, Reading, Massachusetts, 1983.
\end{thebibliography}

This styles the first reference as ‘[1] Leslie …’, and so that … based on \cite{latexdps} produces the matching ‘… based on [1]’. The second \cite produces ‘[1, 2]’. You must compile the document twice to resolve these references.

The mandatory argument widest-label is text that, when typeset, is as wide as the widest item label produced by the \bibitem commands. The tradition is to use 9 for bibliographies with less than 10 references, 99 for ones with less than 100, etc.

The bibliographic list is headed by a title such as ‘Bibliography’. To change it there are two cases. In the book and report classes, where the top level sectioning is \chapter and the default title is ‘Bibliography’, that title is in the macro \bibname. For article, where the class’s top level sectioning is \section and the default is ‘References’, the title is in macro \refname. Change it by redefining the command, as with \renewcommand{\refname}{Cited references}, after \begin{document}.

Language support packages such as babel will automatically redefine \refname or \bibname to fit the selected language.

See Section 8.16 [list], page 65, for the list layout control parameters.

8.24.1 \bibitem

Synopsis:

\bibitem[cite_key] \\
or \\
\bibitem[label]{cite_key}
Generate an entry labeled by default by a number generated using the `enumi` counter. The citation key `cite_key` can be any string of letters, numbers, and punctuation symbols (but not comma).

See Section 8.24 [thebibliography], page 91, for an example.

When provided, the optional `label` becomes the entry label and the `enumi` counter is not incremented. With this

```latex
\begin{thebibliography}
\bibitem[Lamport 1993]{latexdps} Leslie Lamport.
\textit{\LaTeX{}: a document preparation system}.
Addison-Wesley, Reading, Massachusetts, 1993.
\bibitem{texbook} Donald Ervin Knuth.
\textit{The \TeX book}.
Addison-Wesley, Reading, Massachusetts, 1983.
\end{thebibliography}
```

the first entry will be styled as ‘[Lamport 1993] Leslie ...’ (The amount of horizontal space that \LaTeX{} leaves for the label depends on the `widest-label` argument of the `thebibliography` environment; see Section 8.24 [thebibliography], page 91.) Similarly, ...

Based on \cite{latexdps} will produce ‘... based on [Lamport 1994]’.

If you mix `\bibitem` entries having a `label` with those that do not then \LaTeX{} will number the unlabelled ones sequentially. In the example above the `texbook` entry will appear as ‘[1] Donald ...’, despite that it is the second entry.

If you use the same `\cite{latexdps}` twice then you get ‘\LaTeX Warning: There were multiply-defined labels’.

Under the hood, \LaTeX{} remembers the `\cite{latexdps}` information because \\bibitem writes it to the auxiliary file `jobname.aux` (see Section 28.3 [Jobname], page 262). For instance, the above example causes the two \bibcite{latexdps}{Lamport, 1993} and \bibcite{texbook}{1} to appear in that file. The .aux file is read by the \\begin{document} command and then the information is available for \cite commands. This explains why you need to run \LaTeX twice to resolve references: once to write it out and once to read it in.

Because of this two-pass algorithm, when you add a \bibitem or change its `\cite{latexdps}` you may get ‘\LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right’. Fix it by recompiling.

### 8.24.2 `\cite`

**Synopsis:**

```
cite{keys}
```

or

```
cite[subcites]{keys}
```

Generate as output a citation to the references associated with `keys`. The mandatory `keys` is a citation key, or a comma-separated list of citation keys (see Section 8.24.1 [\bibitem], page 91).
This
\begin{thebibliography}
\bibitem{texbook}
Donald Ervin Knuth.
\textit{The \TeX\ book}.
Addison-Wesley, Reading, Massachusetts, 1983.
\end{thebibliography}
produces output like ‘... source is [1]’. You can change the appearance of the citation and of the reference by using bibliography styles if you generate automatically the thebibliography environment. More information in Section 8.24.4 [Using Bib\TeX{}], page 93.

The optional argument \texttt{subcite} is appended to the citation. For example, \texttt{See 14.3 in \cite[p.~314]{texbook}} might produce ‘See 14.3 in [1, p. 314]’.

In addition to what appears in the output, \texttt{\cite} writes information to the auxiliary file \texttt{jobname.aux} (see Section 28.3 [Jobname], page 262). For instance, \texttt{\cite{latexdps}} writes ‘citation[latexdps]’ to that file. This information is used by Bib\TeX{} to include in your reference list only those works that you have actually cited; see Section 8.24.3 [\texttt{\nocite}], page 93, also.

If \texttt{keys} is not in your bibliography information then you get ‘\LaTeX\ Warning: There were undefined references’, and in the output the citation shows as a boldface question mark between square brackets. There are two possible causes. If you have mistyped something, as in \texttt{\cite{texbok}} then you need to correct the spelling. On the other hand, if you have just added or modified the bibliographic information and so changed the \texttt{.aux} file (see Section 8.24.1 [\texttt{\bibitem}], page 91) then the fix may be to run \LaTeX{} again.

\textbf{8.24.3 \texttt{\nocite}}

Synopsis:
\begin{verbatim}
\nocite{keys}
\end{verbatim}

Produces no output but writes \texttt{keys} to the auxiliary file \texttt{jobname.aux} (see Section 28.3 [Jobname], page 262).

The mandatory argument \texttt{keys} is a comma-separated list of one or more citation keys (see Section 8.24.1 [\texttt{\bibitem}], page 91). This information is used by Bib\TeX{} to include these works in your reference list even though you have not explicitly cited them (see Section 8.24.2 [\texttt{\cite}], page 92).

\textbf{8.24.4 Using Bib\TeX{}}

As described in \texttt{thebibliography} (see Section 8.24 [thebibliography], page 91), a sophisticated approach to managing bibliographies is provided by the Bib\TeX{} program. This is only an introduction; see the full documentation on CTAN (see Section 2.8 [CTAN], page 8).

With Bib\TeX{}, you don’t use the \texttt{thebibliography} environment directly (see Section 8.24 [thebibliography], page 91). Instead, include these lines:
\begin{verbatim}
\bibliographystyle{bibstyle}
\bibliography{bibfile1, bibfile2, ...}
\end{verbatim}
The bibstyle refers to a file bibstyle.bst, which defines how your citations will look. The standard bibstyle's distributed with Bib\TeX are:

- **alpha** Labels are formed from name of author and year of publication. The bibliographic items are sorted alphabetically.

- **plain** Labels are integers. Sort the bibliographic items alphabetically.

- **unsrt** Like plain, but entries are in order of citation.

- **abbrev** Like plain, but more compact labels.

Many, many other Bib\TeX style files exist, tailored to the demands of various publications. See the CTAN topic [https://ctan.org/topic/bibtext-sty](https://ctan.org/topic/bibtext-sty).

The \texttt{\textbackslash bibliography} command is what actually produces the bibliography. Its argument is a comma-separated list, referring to files named \texttt{bibfile1.bib}, \texttt{bibfile2.bib}, \ldots These contain your database in Bib\TeX format. This shows a typical couple of entries in that format.

```latex
@book{texbook,
  title = {The {{\TeX}}book},
  author = {D.E. Knuth},
  isbn = {0201134489},
  series = {Computers \& typesetting},
  year = {1983},
  publisher = {Addison-Wesley}
}
@book{sexbook,
  author = {W.H. Masters and V.E. Johnson},
  title = {Human Sexual Response},
  year = {1966},
  publisher = {Bantam Books}
}
```

Only the bibliographic entries referred to via \texttt{\textbackslash cite} and \texttt{\textbackslash nocite} will be listed in the document's bibliography. Thus you can keep all your sources together in one file, or a small number of files, and rely on Bib\TeX to include in this document only those that you used.

With Bib\TeX, the keys argument to \texttt{\textbackslash nocite} can also be the single character '*'. This means to implicitly cite all entries from all given bibliographies.

### 8.24.4.1 Bib\TeX error messages

If you forget to use \texttt{\textbackslash bibliography} or \texttt{\textbackslash bibliographystyle} in your document (or, less likely, any \texttt{\textbackslash cite} or \texttt{\textbackslash nocite} command), Bib\TeX will issue an error message. Because Bib\TeX can be used with any program, not just L\TeX, the error messages refer to the internal commands read by Bib\TeX (from the .aux file), rather than the user-level commands described above.

Here is a table showing internal commands mentioned in the Bib\TeX errors, and the corresponding user-level commands.

```
\bibdata  \bibliography
\bibstyle
\bibliographystyle
```
For example, if your document has no \bibliographystyle command, BibTeX complains as follows:

I found no \bibstyle command---while reading file document.aux

**8.25 theorem**

Synopsis:

\begin{theorem}
  theorem body
\end{theorem}

Produces 'Theorem \textit{n}' in boldface followed by \textit{theorem body} in italics. The numbering possibilities for \textit{n} are described under \newtheorem (see Section 12.9 \newtheorem, page 120).

\begin{itemize}
  \item \newtheorem{lem}{Lemma} % in preamble
  \newtheorem{thm}{Theorem}
  \item text of lemma
  \end{itemize}

The next result follows immediately.

\begin{thm}[Gauss] % put 'Gauss' in parens after theorem head
  text of theorem
\end{thm}

Most new documents use the packages amsthm and amsmath from the American Mathematical Society. Among other things these packages include a large number of options for theorem environments, such as styling options.

**8.26 titlepage**

Synopsis:

\begin{titlepage}
  ... text and spacing ...
\end{titlepage}

Create a title page, a page with no printed page number or heading and with succeeding pages numbered starting with page one.

In this example all formatting, including vertical spacing, is left to the author.

\begin{itemize}
  \item \begin{titlepage}
    \vspace*{\stretch{1}}
    \begin{center}
      \texttt{\huge\bfseries Thesis} \texttt{[[1ex]}
      \texttt{title} \texttt{[[6.5ex]}
      \texttt{\large\bfseries Author name} \texttt{[ ]}
    \end{center}
  \end{titlepage}
\end{itemize}
8.27 verbatim

Synopsis:

\begin{verbatim}
literal-text
\end{verbatim}

A paragraph-making environment in which \LaTeX{} produces as output exactly what you type as input. For instance inside literal-text the backslash \ character does not start commands, it produces a printed \', and carriage returns and blanks are taken literally. The output appears in a monospaced typewriter-like font (\tt).

\begin{verbatim}
Symbol swearing: %&$#!.
\end{verbatim}

The only restriction on literal-text is that it cannot include the string \end{verbatim}.

You cannot use the verbatim environment in the argument to macros, for instance in the argument to a \section. This is not the same as commands being fragile (see Section 12.11 \protect, page 123), instead it just cannot work, as the verbatim environment changes the catcode regime before processing its contents, and restore it immediately afterward, nevertheless with a macro argument the content of the argument has already be converted to a token list along the catcode regime in effect when the macro was called. However, the cprotect package can help with this.

One common use of verbatim input is to typeset computer code. Some packages offer many features not provided by the verbatim environment; two of the most popular are listings and minted. For example, they are capable of pretty-printing, line numbering, and selecting parts of files for a continuing listing.

A package that provides many more options for verbatim environments is fancyvrb. Another is verbatimbox.
For a list of all the relevant packages, see CTAN (see Section 2.8 [CTAN], page 8), particularly the topics listing (https://ctan.org/topic/listing) and verbatim (https://ctan.org/topic/verbatim).

8.27.1 \verb
Synopsis:
\verb char literal-text char
\verb* char literal-text char

Typeset literal-text as it is input, including special characters and spaces, using the typewriter (\tt) font.

This example shows two different invocations of \verb.

This is \verb!literally! the biggest pumpkin ever.
And this is the best squash, \verb+literally+!

The first \verb has its literal-text surrounded with exclamation point, !. The second instead uses plus, +, because the exclamation point is part of literal-text.

The single-character delimiter char surrounds literal-text—it must be the same character before and after. No spaces come between \verb or \verb* and char, or between char and literal-text, or between literal-text and the second occurrence of char (the synopsis shows a space only to distinguish one component from the other). The delimiter must not appear in literal-text. The literal-text cannot include a line break.

The *-form differs only in that spaces are printed with a visible space character. (Namely, \hfil.)

The output from this will include a visible space on both side of word 'with':

The command's first argument is \verb*!filename with extension! and ...

For typesetting Internet addresses, urls, the package url is a better option than the \verb command, since it allows line breaks.

You cannot use \verb in the argument to a macro, for instance in the argument to a \section. It is not a question of \verb being fragile (see Section 12.11 \protect, page 123), instead it just cannot work, as the \verb command changes the catcode regime before reading its argument, and restore it immediately afterward, nevertheless with a macro argument the content of the argument has already be converted to a token list along the catcode regime in effect when the macro was called. However, the cprotect package can help with this.

8.28 verse
Synopsis:
\begin{verse}
  line1 \\linebreak
  line2 \\linebreak
  ...
\end{verse}

An environment for poetry.
Here are two lines from Shakespeare’s Romeo and Juliet.

Then plainly know my heart’s dear love is set \\
On the fair daughter of rich Capulet.

Separate the lines of each stanza with \\
, and use one or more blank lines to separate the stanzas.

\begin{verse}
\makebox[\linewidth][c]{\textit{Shut Not Your Doors} ---Walt Whitman}
\\[1\baselineskip] Shut not your doors to me proud libraries, \\
For that which was lacking on all your well-fill'd shelves, \\
quad yet needed most, I bring, \\
Forth from the war emerging, a book I have made, \\
The words of my book nothing, the drift of it every thing, \\
A book separate, not link'd with the rest nor felt by the intellect, \\
But you ye untold latencies will thrill to every page.
\end{verse}

The output has margins indented on the left and the right, paragraphs are not indented, and the text is not right-justified.
9 Line breaking

The first thing \TeX{} does when processing ordinary text is to translate your input file into a sequence of glyphs and spaces. To produce a printed document, this sequence must be broken into lines (and these lines must be broken into pages).

\TeX{} usually does the line (and page) breaking in the text body for you but in some environments you manually force line breaks.

A common workflow is to get a final version of the document content before taking a final pass through and considering line breaks (and page breaks). This differs from word processing, where you are formatting text as you input it. Putting these off until the end prevents a lot of fiddling with breaks that will change anyway.

9.1 \\ 

Synopsis, one of:

```
\\
\\[morespace]
```

or one of:

```
\\*
\\*[morespace]
```

End the current line. The optional argument *morespace* specifies extra vertical space to be inserted before the next line. This is a rubber length (see Chapter 14 \[Lengths\], page 137) and can be negative. The text before the line break is set at its normal length, that is, it is not stretched to fill out the line width. This command is fragile (see Section 12.11 \[\protect\], page 123).

```
\title{My story: \\
[0.25in]
a tale of woe}
```

The starred form, `\\\*', tells \TeX{} not to start a new page between the two lines, by issuing a `\break{}'.

Explicit line breaks in the main text body are unusual in \TeX{}. In particular, don’t start new paragraphs with `\\\'. Instead leave a blank line between the two paragraphs. And don’t put in a sequence of `\\’s to make vertical space. Instead use `\vspace{length}', or `\leavevmode\vspace{length}', or `\vspace*{length}' if you want the space to not be thrown out at the top of a new page (see Section 19.14 \[\vspace\], page 194).

The `\\' command is mostly used outside of the main flow of text such as in a tabular or array environment or in an equation environment.

The `\\' command is a synonym for `\newline' (see Section 9.3 \[\newline\], page 101) under ordinary circumstances (an example of an exception is the `p{\ldots}' column in a tabular environment; see Section 8.23 \[tabular\], page 85).

The `\\' command is a macro, and its definition changes by context so that its definition in normal text, a center environment, a flushleft environment, and a tabular are all different. In normal text when it forces a linebreak it is essentially a shorthand for `\newline. It does not end horizontal mode or end the paragraph, it just inserts some glue and penalties so that when the paragraph does end a linebreak will occur at that point, with the short line padded with white space.
You get ‘LaTeX Error: There’s no line here to end’ if you use `\` to ask for a new line, rather than to end the current line. An example is if you have `\begin{document}` `\` or, more likely, something like this.

\begin{center}
\begin{minipage}{0.5\textwidth}
\`
In that vertical space put your mark.
\end{minipage}
\end{center}

Fix it by replacing the double backslash with something like `\vspace{\baselineskip}`.

### 9.2 \texttt{\obeycr} & \texttt{\restorecr}

The `\texttt{\obeycr}` command makes a return in the input file (`\texttt{\textasciicircum\textasciicircum M}`, internally) the same as `\`, followed by `\texttt{\relax}`. So each new line in the input will also be a new line in the output. The `\texttt{\restorecr}` command restores normal line-breaking behavior.

This is not the way to show verbatim text or computer code. Use `verbatim` (see Section 8.27 [verbatim], page 96) instead.

With \LaTeX’s usual defaults, this

```
aaa
bbb

\obeycr
ccc
ddd
eee

\restorecr
fff
ggg

hhh
iii
```

produces output like this.

```
  aaa  bbb
  ccc
ddd
eee

fff  ggg
  hhh  iii
```

The indents are paragraph indents.
Chapter 9: Line breaking

9.3 \newline

In ordinary text, this ends a line in a way that does not right-justify it, so the text before the end of line is not stretched. That is, in paragraph mode (see Chapter 17 [Modes], page 177), the \newline command is equivalent to double-backslash (see Section 9.1 [\\], page 99). This command is fragile (see Section 12.11 [\protect], page 123).

However, the two commands are different inside a tabular or array environment. In a column with a specifier producing a paragraph box such as typically p{...}, \newline will insert a line end inside of the column; that is, it does not break the entire tabular row. To break the entire row use \ or its equivalent \tabularnewline.

This will print ‘Name:’ and ‘Address:’ as two lines in a single cell of the table.

\begin{tabular}{p{1in}@{\hspace{2in}}p{1in}}
  Name: & Address: \\
\end{tabular}

The ‘Date:’ will be baseline-aligned with ‘Name:’.

9.4 \- (discretionary hyphen)

Tell \LaTeX{} that it may hyphenate the word at that point. When you insert \- commands in a word, the word will only be hyphenated at those points and not at any of the other hyphenation points that \LaTeX{} might otherwise have chosen. This command is robust (see Section 12.11 [\protect], page 123).

\LaTeX{} is good at hyphenating and usually finds most of the correct hyphenation points, while almost never using an incorrect one. The \- command is for exceptional cases.

For example, \LaTeX{} does not ordinarily hyphenate words containing a hyphen. Below, the long and hyphenated word means \LaTeX{} has to put in unacceptably large spaces to set the narrow column.

\begin{tabular}{rp{1.75in}}
  Isaac Asimov & The strain of \\
  & anti-intellectualism \\
  & % an\-ti-in\-tel\-lec\-tu\-al\-ism \\
  & has been a constant thread winding its way through our \\
  & political and cultural life, nurtured by \\
  & the false notion that democracy means that \\
  & `my ignorance is just as good as your knowledge'. \\
\end{tabular}

Commenting out the third line and uncommenting the fourth makes a much better fit.

The \- command only allows \LaTeX{} to break there, it does not require that it break there. You can force a split with something like Hef-\linebreak feron. Of course, if you later change the text then this forced break may look out of place, so this approach requires care.

9.5 \slash: breakable ‘/’

The \slash command produces a ‘/’ character and then a penalty of the same value as an explicit ‘-’ character (\exhyphenpenalty). This allows \TeX{} to break a line at the ‘/’,
similar to a hyphen. Hyphenation is allowed in the word part preceding the ‘/’, but not after. For example:

The input\slash output of the program is complicated.

9.6 \texttt{\textbackslash discretion\textbackslash ary (generalized hyphenation point)}

Synopsis:

\texttt{\textbackslash discretion\textbackslash ary\{pre-break\}\{post-break\}\{no-break\}}

Handle word changes around hyphens. This command is not often used in \LaTeX\ documents.

If a line break occurs at the point where \texttt{\textbackslash discretion\textbackslash ary} appears then \TeX\ puts \texttt{pre-break} at the end of the current line and puts \texttt{post-break} at the start of the next line. If there is no line break here then \TeX\ puts \texttt{no-break}.

In ‘difficult’ the three letters \texttt{ffi} form a ligature. But \TeX\ can nonetheless break between the two ‘f’’s with this.

\texttt{di\textbackslash discretion\textbackslash ary\{f-\}\{fi\}\{ffi\}\textbackslash cult}

Note that users do not have to do this. It is typically handled automatically by \TeX\’s hyphenation algorithm.

9.7 \texttt{\textbackslash fussy \& \textbackslash sloppy}

Declarations to make \TeX\ more picky or less picky about line breaking. Declaring \texttt{\textbackslash fussy} usually avoids too much space between words, at the cost of an occasional overfull box. Conversely, \texttt{\textbackslash sloppy} avoids overfull boxes while suffering loose interword spacing.

The default is \texttt{\textbackslash fussy}. Line breaking in a paragraph is controlled by whichever declaration is current at the end of the paragraph, i.e., at the blank line or \texttt{\textbackslash par} or displayed equation ending that paragraph. So to affect the line breaks, include that paragraph-ending material in the scope of the command.

9.7.1 sloppypar

Synopsis:

\begin{verbatim}
\begin{sloppypar}
... paragraphs ...
\end{sloppypar}
\end{verbatim}

Typeset the paragraphs with \texttt{\textbackslash sloppy} in effect (see Section 9.7 [\texttt{\textbackslash fussy \& \textbackslash sloppy}], page 102). Use this to locally adjust line breaking, to avoid ‘Overfull box’ or ‘Underfull box’ errors.

The example is simple.

\begin{verbatim}
\begin{sloppypar}
Her plan for the morning thus settled, she sat quietly down to her book after breakfast, resolving to remain in the same place and the same employment till the clock struck one; and from habitude very little incommoded by the remarks and ejaculations of Mrs.\ Allen, whose vacancy of mind and incapacity for thinking were such, that as she never talked a great deal, so she could never be entirely
\end{verbatim}
silent; and, therefore, while she sat at her work, if she lost her
needle or broke her thread, if she heard a carriage in the street,
or saw a speck upon her gown, she must observe it aloud, whether
there were anyone at leisure to answer her or not.
\end{sloppypar}

9.8 \texttt{\hyphenation}

Synopsis:
\texttt{\hyphenation\{word1 ...\}}
Declares allowed hyphenation points within the words in the list. The words in that
list are separated by spaces. Show permitted points for hyphenation with an ASCII dash character, \texttt{-}.

Here is an example:
\texttt{\hyphenation\{hat-er il-lit-e-ra-ti tru-th-i-ness\}}
Use lowercase letters. \TeX{} will only hyphenate if the word matches exactly; no inflections
are tried. Multiple \texttt{\hyphenation} commands accumulate.

9.9 \texttt{\linebreak} & \texttt{\nolinebreak}

Synopses, one of:
\texttt{\linebreak}
\texttt{\linebreak[zero-to-four]}
or one of these.
\texttt{\nolinebreak}
\texttt{\nolinebreak[zero-to-four]}

Encourage or discourage a line break. The optional \texttt{zero-to-four} is an integer lying
between 0 and 4 that allows you to soften the instruction. The default is 4, so that without
the optional argument these commands entirely force or prevent the break. But for instance,
\texttt{\nolinebreak[1]} is a suggestion that another place may be better. The higher the number,
the more insistent the request. Both commands are fragile (see Section 12.11 \texttt{\protect},
page 123).

Here we tell \LaTeX{} that a good place to put a linebreak is after the standard legal text.
\texttt{\boilerplatelegal{} \linebreak[2]}
We especially encourage applications from members of traditionally
underrepresented groups.

When you issue \texttt{\linebreak}, the spaces in the line are stretched out so that the break
point reaches the right margin. See Section 9.1 \texttt{\\}, page 99, and Section 9.3 \texttt{\newline},
page 101, to have the spaces not stretched out.
10 Page breaking

Ordinarily \LaTeX{} automatically takes care of breaking output into pages with its usual aplomb. But if you are writing commands, or tweaking the final version of a document, then you may need to understand how to influence its actions.

\LaTeX{}’s algorithm for splitting a document into pages is more complex than just waiting until there is enough material to fill a page and outputting the result. Instead, \LaTeX{} typesets more material than would fit on the page and then chooses a break that is optimal in some way (it has the smallest \textit{badness}). An example of the advantage of this approach is that if the page has some vertical space that can be stretched or shrunk, such as with rubber lengths between paragraphs, then \LaTeX{} can use that to avoid widow lines (where a new page starts with the last line of a paragraph; \LaTeX{} can squeeze the extra line onto the first page) and orphans (where the first line of paragraph is at the end of a page; \LaTeX{} can stretch the material of the first page so the extra line falls on the second page). Another example is where \LaTeX{} uses available vertical shrinkage to fit on a page not just the header for a new section but also the first two lines of that section.

But \LaTeX{} does not optimize over the entire document’s set of page breaks. So it can happen that the first page break is great but the second one is lousy; to break the current page \LaTeX{} doesn’t look as far ahead as the next page break. So occasionally you may want to influence page breaks while preparing a final version of a document.

See Chapter 5 [Layout], page 26, for more material that is relevant to page breaking.

10.1 \texttt{\clearpage} & \texttt{\cleardoublepage}

Synopsis:

\texttt{\clearpage}

or

\texttt{\cleardoublepage}

End the current page and output all of the pending floating figures and tables (see Section 5.7 [Floats], page 32). If there are too many floats to fit on the page then \LaTeX{} will put in extra pages containing only floats. In two-sided printing, \texttt{\cleardoublepage} also makes the next page of content a right-hand page, an odd-numbered page, if necessary inserting a blank page. The \texttt{\clearpage} command is robust while \texttt{\cleardoublepage} is fragile (see Section 12.11 [\protect], page 123).

\LaTeX{}’s page breaks are optimized so ordinarily you only use this command in a document body to polish the final version, or inside commands.

The \texttt{\cleardoublepage} command will put in a blank page, but it will have the running headers and footers. To get a really blank page, use this command.

\texttt{\let\origdoublepage\cleardoublepage}
\texttt{\newcommand{\clearemptydoublepage}{\%}
\clearpage
\{\pagestyle{empty}\origdoublepage\%}
If you want \LaTeX{}’s standard \texttt{\chapter}\ command to do this then add the line
\texttt{\let\cleardoublepage\clearemptydoublepage}. (Of course this affects all uses of
\texttt{\cleardoublepage}, not just the one in \texttt{\chapter}.)

The command \texttt{\newpage} (see Section 10.2 [\newpage], page 105) also ends the current
page, but without clearing pending floats. And, if \LaTeX{} is in two-column mode then
\texttt{\newpage} ends the current column while \texttt{\clearpage} and \texttt{\cleardoublepage} end the
current page.

\section*{10.2 \newpage}

Synopsis:

\begin{verbatim}
\newpage
\end{verbatim}

End the current page. This command is robust (see Section 12.11 [\protect], page 123).
\LaTeX{}’s page breaks are optimized so ordinarily you only use this command in a document
body to polish the final version, or inside commands.

While the commands \texttt{\clearpage} and \texttt{\cleardoublepage} also end the current page,
in addition they clear pending floats (see Section 10.1 [\clearpage & \cleardoublepage],
page 104). And, if \LaTeX{} is in two-column mode then \texttt{\clearpage} and \texttt{\cleardoublepage}
end the current page, possibly leaving an empty column, while \texttt{\newpage} only ends the
current column.

In contrast with \texttt{\pagebreak} (see Section 10.4 [\pagebreak & \nopagebreak], page 106),
the \texttt{\newpage} command will cause the new page to start right where requested. This

\begin{verbatim}
Four score and seven years ago our fathers brought forth on this
continent,
\newpage
noindent a new nation, conceived in Liberty, and dedicated to the
proposition that all men are created equal.
\end{verbatim}

makes a new page start after ‘\texttt{continent}’, and the cut-off line is not right justified. In
addition, \texttt{\newpage} does not vertically stretch out the page, as \texttt{\pagebreak} does.

\section*{10.3 \enlargethispage}

Synopsis, one of:

\begin{verbatim}
\enlargethispage{size}
\enlargethispage*{size}
\end{verbatim}

Enlarge the \texttt{\textheight} for the current page. The required argument \texttt{size} must be a
rigid length (see Chapter 14 [Lengths], page 137). It may be positive or negative. This
command is fragile (see Section 12.11 [\protect], page 123).

A common strategy is to wait until you have the final text of a document, and then pass
through it tweaking line and page breaks. This command allows you some page size leeway.

This will allow one extra line on the current page.

\begin{verbatim}
\enlargethispage{\baselineskip}
\end{verbatim}

The starred form \texttt{\enlargethispage*} tries to squeeze the material together on the
page as much as possible, for the common use case of getting one more line on the page.
This is often used together with an explicit \texttt{\pagebreak}. 
10.4 `\pagebreak` & `\nopagebreak`

Synopses:

```
\pagebreak
\pagebreak[zero-to-four]
```

or

```
\nopagebreak
\nopagebreak[zero-to-four]
```

Encourage or discourage a page break. The optional `zero-to-four` is an integer that allows you to soften the request. The default is 4, so that without the optional argument these commands entirely force or prevent the break. But for instance `\nopagebreak[1]` suggests to \LaTeX{} that another spot might be preferable. The higher the number, the more insistent the request. Both commands are fragile (see Section 12.11 [\protect], page 123).

\LaTeX{}’s page endings are optimized so ordinarily you only use these commands in a document body to polish the final version, or inside commands.

If you use these inside a paragraph, they apply to the point following the line in which they appear. So this

```
Four score and seven years ago our fathers brought forth on this continent,
\pagebreak
a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.
```

does not give a page break at ‘continent’, but instead at ‘nation’, since that is where \LaTeX{} breaks that line. In addition, with `\pagebreak` the vertical space on the page is stretched out where possible so that it extends to the normal bottom margin. This can look strange, and if `\flushbottom` is in effect this can cause you to get ‘\texttt{Underfull \vbox (badness 10000) has occurred while \output is active}’. See Section 10.2 [\newpage], page 105, for a command that does not have these effects.

A declaration `\samepage` and corresponding `samepage` environment try to only allow breaks between paragraphs. They are not perfectly reliable. For more on keeping material on the same page, see the FAQ entry [https://texfaq.org/FAQ-nopagebrk](https://texfaq.org/FAQ-nopagebrk).
11 Footnotes

Place a footnote at the bottom of the current page, as here.

Noël Coward quipped that having to read a footnote is like having to go downstairs to answer the door, while in the midst of making love.\footnote{\% I wouldn't know, I don't read footnotes.}

You can put multiple footnotes on a page. If the footnote text becomes too long then it will flow to the next page.

You can also produce footnotes by combining the \footnotemark and the \footnotetext commands, which is useful in special circumstances.

To make bibliographic references come out as footnotes you need to include a bibliographic style with that behavior (see Section 8.24.4 [Using BibTeX], page 93).

11.1 \footnote

Synopsis, one of:

\footnote{text}
\footnote[number]{text}

Place a footnote text at the bottom of the current page, with a footnote marker at the current position in the text.

There are over a thousand footnotes in Gibbon's \textit{Decline and Fall of the Roman Empire}.\footnote{After reading an early version with endnotes David Hume complained, ``One is also plagued with his Notes, according to the present Method of printing the Book'' and suggested that they ``only to be printed at the Margin or the Bottom of the Page.''}

The optional argument number allows you to specify the number of the footnote. If you use this then L\TeX{} does not increment the footnote counter.

By default, L\TeX{} uses arabic numbers as footnote markers. Change this with something like \texttt{\renewcommand{\thefootnote}{\fnsymbol{footnote}}}, which uses a sequence of symbols (see Section 13.1 [alph Alph arabic roman Roman fnsymbol], page 133). To make this change global put that in the preamble. If you make the change local then you may want to reset the counter with \texttt{\setcounter{footnote}{0}}.

L\TeX{} determines the spacing of footnotes with two parameters.

\footnoterule

Produces the rule separating the main text on a page from the page’s footnotes. Default dimensions in the standard document classes (except \texttt{slides}, where it does not appear) are: vertical thickness of 0.4pt, and horizontal size of 0.4\texttt{columnwidth} long. Change the rule with something like this.

% \footnoterule is expanded in vertical mode, thus \kern
% commands ensure that no vertical space is created,
% and the rule is separated vertically with 2pt
% above the note text.
\renewcommand*{\footnoterule}{% 
  \kern -3pt % This -3 is negative 
  \hrule width \textwidth height 1pt % of the sum of this 1 
  \kern 2pt} % and this 2

\footnotesep
The height of the strut placed at the beginning of the footnote (see Section 19.13 [\strut], page 193). By default, this is set to the normal strut for \footnotesize fonts (see Section 4.3 [Font sizes], page 22), therefore there is no extra space between footnotes. This is ‘6.65pt’ for ‘10pt’, ‘7.7pt’ for ‘11pt’, and ‘8.4pt’ for ‘12pt’. Change it as with \setlength{\footnotesep}{11pt}.

The \footnote command is fragile (see Section 12.11 [\protect], page 123).
\LaTeX’s default puts many restrictions on where you can use a \footnote; for instance, you cannot use it in an argument to a sectioning command such as \chapter (it can only be used in outer paragraph mode; see Chapter 17 [Modes], page 177). There are some workarounds; see following sections.

In a minipage environment the \footnote command uses the mpfootnote counter instead of the footnote counter, so they are numbered independently. They are shown at the bottom of the environment, not at the bottom of the page. And by default they are shown alphabetically. See Section 8.18 [minipage], page 71, and Section 11.5 [Footnotes in a table], page 110.

11.2 \footnotemark
Synopsis, one of:
\footnotemark
\footnotemark[number]

Put the current footnote mark in the text. To specify associated text for the footnote see Section 11.3 [\footnotetext], page 109. The optional argument number causes the command to use that number to determine the footnote mark. This command can be used in inner paragraph mode (see Chapter 17 [Modes], page 177).

If you use \footnotemark without the optional argument then it increments the footnote counter, but if you use the optional number then it does not. The next example produces several consecutive footnote markers referring to the same footnote.

The first theorem\footnote{Due to Gauss.} and the second theorem\footnotemark[value{footnote}] and the third theorem.\footnotemark[value{footnote}]

If there are intervening footnotes then you must remember the value of the number of the common mark. This example gives the same institutional affiliation to both the first and third authors (\thanks is a version of \footnote), by explicitly specifying the number of the footnote (‘1’).

\title{A Treatise on the Binomial Theorem}
\author{J Moriarty\thanks{University of Leeds}}
  \and A C Doyle\thanks{Durham University}
  \and S Holmes\footnotemark[1]
This example accomplishes the same by using the package `cleveref`.

\usepackage{cleveref}[2012/02/15] % in preamble
\crefformat{footnote}{#2\footnotemark[#1]#3}

... The theorem is from Evers.\footnote{Tinker, Evers, 1994.}
The corollary is from Chance.\footnote{Evers, Chance, 1990.}
But the key lemma is from Tinker.\cref{fn:TE}

It will work with the package `hyperref`.

This uses a counter to remember the footnote number. The third sentence is followed by the same footnote marker as the first.

\newcounter{footnoteValueSaver}
All babies are illogical.\footnote{Lewis Carroll.}\setcounter{footnoteValueSaver}{\value{footnote}}
Nobody is despised who can manage a crocodile.\footnote{Captain Hook.}
Illogical persons are despised.\footnotemark[\value{footnoteValueSaver}]
Therefore, anyone who can manage a crocodile is not a baby.

\section{footnotetext}

Synopsis, one of:

\footnotetext{text}
\footnotetext[number]{text}

Place \textit{text} at the bottom of the page as a footnote. It pairs with \footnotemark (see Section 11.2 [footnotemark], page 108) and can come anywhere after that command, but must appear in outer paragraph mode (see Chapter 17 [Modes], page 177). The optional argument \textit{number} changes the number of the footnote mark.

See Section 11.2 [footnotemark], page 108, and Section 11.5 [Footnotes in a table], page 110, for usage examples.

\subsection{Footnotes in section headings}

Putting a footnote in a section heading, as in:

\section{Full sets\protect\footnote{This material due to ...}}

causes the footnote to appear at the bottom of the page where the section starts, as usual, but also at the bottom of the table of contents, where it is not likely to be desired. The simplest way to have it not appear on the table of contents is to use the optional argument to \section.

\section[Please]{Please\footnote{\protect Don't footnote in chapter and section headers!}}

No \texttt{\protect} is needed in front of \footnote here because what gets moved to the table of contents is the optional argument.
11.5 Footnotes in a table

Inside a `tabular` or `array` environment the \footnote command does not work; there is a footnote mark in the table cell but the footnote text does not appear. The solution is to use a `minipage` environment as here (see Section 8.18 [minipage], page 71).

\begin{center}
\begin{minipage}{\textwidth}
\centering
\begin{tabular}{l|l}
\textsc{Ship} & \textsc{Book} \\
\hline
\textit{HMS Sophie} & \textit{Master and Commander} \\
\textit{HMS Polychrest} & \textit{Post Captain} \\
\textit{HMS Lively} & \textit{Post Captain} \\
\textit{HMS Surprise} & \textit{A number of books}\footnote{Starting with \textit{HMS Surprise}.}
\end{tabular}
\end{minipage}
\end{center}

Inside a `minipage`, footnote marks are lowercase letters. Change that with something like \renewcommand{\thempfootnote}{\arabic{mpfootnote}} (see Section 13.1 [alph \Alph arabic \roman \Roman \fnsymbol], page 133).

The footnotes in the prior example appear at the bottom of the `minipage`. To have them appear at the bottom of the main page, as part of the regular footnote sequence, use the \footnotemark and \footnotetext pair and make a new counter.

\newcounter{mpFootnoteValueSaver}
\begin{center}
\begin{minipage}{\textwidth}
\setcounter{mpFootnoteValueSaver}{\value{footnote}} \centering
\begin{tabular}{l|l}
\textsc{Woman} & \textsc{Relationship} \\
Mona & Attached\footnotemark \\
Diana Villiers & Eventual wife \\
Christine Hatherleigh Wood & Fiance\footnotemark
\end{tabular}
\end{minipage} \\
\stepcounter{mpFootnoteValueSaver} \footnotetext{\value{mpFootnoteValueSaver}}{\textit{Little is known other than her death.}} \\
\stepcounter{mpFootnoteValueSaver} \footnotetext{\value{mpFootnoteValueSaver}}{\textit{Relationship is unresolved.}}
\end{center}

For a floating `table` environment (see Section 8.22 [table], page 84), use the `tablefootnote` package.

\usepackage{tablefootnote} % in preamble

\begin{table}
\begin{tabular}{l|l} \textsc{Date} & \textsc{Campaign} \\ \hline 1862 & Fort Donelson \\ 1863 & Vicksburg \\ 1865 & Army of Northern Virginia \tablefootnote{Ending the war.} \\
\end{tabular}
\caption{Forces captured by US Grant}
\end{table}

The footnote appears at the page bottom and is numbered in sequence with other footnotes.

11.6 Footnotes of footnotes

Particularly in the humanities, authors can have multiple classes of footnotes, including having footnotes of footnotes. The package \texttt{bigfoot} extends \LaTeX{}'s default footnote mechanism in many ways, including allow these two, as in this example.

\begin{verbatim}
\usepackage{bigfoot} % in preamble
\DeclareNewFootnote{Default}
\DeclareNewFootnote{from}[alph] % create class \footnotefrom{}
\end{verbatim}

The third theorem is a partial converse of the second. \footnotefrom{Noted in Wilson.}

\footnote{Second edition only.}
12 Definitions

\LaTeX{} has support for making new commands of many different kinds.

### 12.1 \texttt{\newcommand} & \texttt{\renewcommand}

Synopses, one of (three regular forms, three starred forms):

\begin{verbatim}
\newcommand{\cmd}{defn}
\newcommand{\cmd}[nargs]{defn}
\newcommand{\cmd}[nargs][optargdefault]{defn}
\newcommand*{\cmd}{defn}
\newcommand*{\cmd}[nargs]{defn}
\newcommand*{\cmd}[nargs][optargdefault]{defn}
\end{verbatim}

or the same six possibilities with \texttt{\renewcommand} instead of \texttt{\newcommand}:

\begin{verbatim}
\renewcommand{\cmd}{defn}
\renewcommand{\cmd}[nargs]{defn}
\renewcommand{\cmd}[nargs][optargdefault]{defn}
\renewcommand*{\cmd}{defn}
\renewcommand*{\cmd}[nargs]{defn}
\renewcommand*{\cmd}[nargs][optargdefault]{defn}
\end{verbatim}

Define or redefine a command (see also \texttt{\DeclareRobustCommand} in \textsection{}12.14 [Class and package commands], page 125).

The starred form of these two forbids the arguments from containing multiple paragraphs of text (i.e., a \texttt{\par} token; in plain \LaTeX{} terms: the commands are not \texttt{\long}). With the default form, arguments can be multiple paragraphs.

These are the parameters (examples follow):

- **cmd** Required; \texttt{\cmd} is the command name. It must begin with a backslash, \	extbackslash{}, and must not begin with the four character string \texttt{\end}. For \texttt{\newcommand}, it must not be already defined. For \texttt{\renewcommand}, this name must already be defined.

- **nargs** Optional; an integer from 0 to 9, specifying the number of arguments that the command takes, including any optional argument. Omitting this argument is the same as specifying 0, meaning that the command has no arguments. If you redefine a command, the new version can have a different number of arguments than the old version.

- **optargdefault** Optional; if this argument is present then the first argument of \texttt{\cmd} is optional, with default value \texttt{optargdefault} (which may be the empty string). If \texttt{optargdefault} is not present then \texttt{\cmd} does not take an optional argument.

That is, if \texttt{\cmd} is called with a following argument in square brackets, as in \texttt{\cmd[optval] {...}...}, then within \texttt{defn} the parameter \texttt{#1} is set to \texttt{optval}. On the other hand, if \texttt{\cmd} is called without following square brackets then within \texttt{defn} the parameter \texttt{#1} is set to \texttt{optargdefault}. In either case, the required arguments start with \texttt{#2}.
Omitting \texttt{[optargdefault]} from the definition is entirely different from giving the square brackets with empty contents, as in \texttt{[ ]}. The former says the command being defined takes no optional argument, so \texttt{#1} is the first required argument (if \texttt{nargs} \texttt{\geq} 1); the latter sets the optional argument \texttt{#1} to the empty string as the default, if no optional argument was given in the call.

Similarly, omitting \texttt{[optval]} from a call is also entirely different from giving the square brackets with empty contents. The former sets \texttt{#1} to the value of \texttt{optval} (assuming the command was defined to take an optional argument); the latter sets \texttt{#1} to the empty string, just as with any other value.

If a command is not defined to take an optional argument, but is called with an optional argument, the results are unpredictable: there may be a \LaTeX error, there may be incorrect typeset output, or both.

defn Required; the text to be substituted for every occurrence of \texttt{\cmd}. The parameters \texttt{#1}, \texttt{#2}, \ldots, \texttt{#nargs} are replaced by the values supplied when the command is called (or by \texttt{optargdefault} in the case of an optional argument not specified in the call, as just explained).

\TeX ignores blanks in the source following a control word (see Section 12.1.1 \textit{[Control sequences]}, page 114), as in \texttt{\cmd}. If you want a space there, one solution is to type \texttt{\{} after the command (\texttt{\cmd{}}), and another solution is to use an explicit control space (\texttt{\cmd}).

A simple example of defining a new command: \texttt{\newcommand{\RS}{Robin Smith}} results in \texttt{\RS} being replaced by the longer text. Redefining an existing command is similar: \texttt{\renewcommand{\qedsymbol}{\small QED}}.

If you use \texttt{\newcommand} and the command name has already been used then you get something like \texttt{\LaTeX Error: Command \fred already defined. Or name \end\ldots illegal, see p.192 of the manual}. Similarly, If you use \texttt{\renewcommand} and the command name has not been defined then you get something like \texttt{\LaTeX Error: \hank undefined}.

Here the first definition creates a command with no arguments, and the second, a command with one required argument:

\begin{verbatim}
\newcommand{\student}{Ms~O'Leary}
\newcommand{\defref}[1]{Definition\ref{#1}}
\end{verbatim}

Use the first as in \texttt{I highly recommend \student} to you. The second has a variable argument, so that \texttt{\defref{def:basis}} expands to \texttt{Definition\ref{def:basis}}, which ultimately expands to something like \texttt{Definition\ref{3.14}}.

Similarly, but with two required arguments: \texttt{\newcommand{\nbym}[2]{#1 \times #2}} is invoked as \texttt{\nbym[2]{k}}.

This example has an optional argument.

\begin{verbatim}
\newcommand{\salutation}[1][Sir or Madam]{Dear #1:}
\end{verbatim}

Then \texttt{\salutation} gives ‘Dear Sir or Madam:’ while \texttt{\salutation[John]} gives ‘Dear John.’. And \texttt{\salutation[]} gives ‘Dear :’.

This example has an optional argument and two required arguments.

\begin{verbatim}
\newcommand{\lawyers}[3][company]{#2, #3, and~#1}
\end{verbatim}
I employ \lawyers{Howe}{Dewey}{Cheatem}.
The output is ‘I employ Dewey, Cheatem, and Howe.’. The optional argument, Howe, is associated with #1, while Dewey and Cheatem are associated with #2 and #3. Because of the optional argument, \lawyers{Dewey}{Cheatem} will give the output ‘I employ Dewey, Cheatem, and company.’.

The braces around defn do not define a group, that is, they do not delimit the scope of the result of expanding defn. For example, with \newcommand{\shipname}{\it #1}, in this sentence,

The \shipname{Monitor} met the \shipname{Merrimac}.

the words ‘met the’, and the period, would incorrectly be in italics. The solution is to put another pair of braces inside the definition: \newcommand{\shipname}[1]{\{\it #1\}}.

12.1.1 Control sequence, control word and control symbol

When reading input \TeX \text{ converts the stream of read characters into a sequence of tokens. When } \TeX \text{ sees a backslash } \backslash, \text{ it will handle the following characters in a special way in order to make a control sequence token.} \text{ The control sequences fall into two categories:}

- control word, when the control sequence is gathered from a \backslash followed by at least one ASCII letter (A-Z and a-z), followed by at least one non-letter.
- control symbol, when the control sequence is gathered from a \backslash followed by one non-letter character.

The sequence of characters so found after the \backslash is also called the control sequence name. Blanks after a control word are ignored and do not produce any whitespace in the output (see Section 12.1 [newcommand & \renewcommand], page 112, and Section 19.6 [SPACE], page 188).

Just as the \relax command does nothing, the following input will simply print ‘Hello!’: We use visible spaces ‘␣’ instead of blanks:

\relax
Hello!

This is because blanks after \relax, including the newline, are ignored, and blanks at the beginning of a line are also ignored (see [Leading blanks], page 188).

12.2 \providecommand

Synopese, one of:

\providecommand{\cmd}[\defn]
\providecommand{\cmd}[\nargs][\defn]
\providecommand{\cmd}[\nargs][\optargdefault][\defn]
\providecommand*{\cmd}[\defn]
\providecommand*{\cmd}[\nargs][\optargdefault][\defn]

Defines a command, as long as no command of this name already exists. If no command of this name already exists then this has the same effect as \newcommand. If a command of this name already exists then this definition does nothing. This is particularly useful in a
file that may be loaded more than once, such as a style file. See Section 12.1 \newcommand & \renewcommand, page 112, for the description of the arguments.

This example
\providecommand{\myaffiliation}{Saint Michael's College}
\providecommand{\myaffiliation}{Lycée Henri IV}
From \myaffiliation.

outputs ‘From Saint Michael's College.’. Unlike \newcommand, the repeated use of \providecommand to (try to) define \myaffiliation does not give an error.

12.3 \makeatletter & \makeatother

Synopsis:
\makeatletter
... definition of commands with @ in their name . .
\makeatother

Use this pair when you redefine \LaTeX commands that are named with an at-sign character ‘@’. The \makeatletter declaration makes the at-sign character have the category code of a letter, code 11. The \makeatother declaration sets the category code of the at-sign to code 12, its default value.

As \LaTeX reads characters, it assigns each one a category code, or catcode. For instance, it assigns the backslash character ‘\’ the catcode 0. Command names consist of a category 0 character, ordinarily backslash, followed by letters, category 11 characters (except that a command name can also consist of a category 0 character followed by a single non-letter symbol).

\LaTeX’s source code has the convention that some commands use @ in their name. These commands are mainly intended for package or class writers. The convention prevents authors who are just using a package or class from accidentally replacing such a command with one of their own, because by default the at-sign has catcode 12.

Use the pair \makeatletter and \makeatother inside a .tex file, typically in the preamble, when you are defining or redefining commands named with @, by having them surround your definition. Don’t use these inside .sty or .cls files since the \usepackage and \documentclass commands already arrange that the at-sign has the character code of a letter, catcode 11.

For a comprehensive list of macros with an at-sign in their names see https://ctan.org/pkg/macros2e.

In this example the class file has a command \thesis@universityname that the user wants to change. These three lines should go in the preamble, before the \begin{document}.
\makeatletter
\renewcommand{\thesis@universityname}{Saint Michael's College}
\makeatother

12.4 \@ifstar

Synopsis:
\newcommand{\mycmd}{\@ifstar{\mycmd@star}{\mycmd@nostar}}
Many standard \LaTeX environments or commands have a variant with the same name but ending with a star character *, an asterisk. Examples are the \texttt{table} and \texttt{table*} environments and the \texttt{section} and \texttt{section*} commands.

When defining environments, following this pattern is straightforward because \texttt{newenvironment} and \texttt{renewenvironment} allow the environment name to contain a star. So you just have to write \texttt{newenvironment{myenv}} or \texttt{newenvironment{myenv*}} and continue the definition as usual. For commands the situation is more complex as the star not being a letter cannot be part of the command name. As in the synopsis above, there will be a user-called command, given above as \texttt{mycmd}, which peeks ahead to see if it is followed by a star. For instance, \LaTeX does not really have a \texttt{section*} command; instead, the \texttt{section} command peeks ahead. This command does not accept arguments but instead expands to one of two commands that do accept arguments. In the synopsis these two are \texttt{mycmd@nostar} and \texttt{mycmd@star}. They could take the same number of arguments or a different number, or no arguments at all. As always, in a \LaTeX document a command using an at-sign @ in its name must be enclosed inside a \texttt{makeatletter ... makeatother} block (see Section 12.3 \texttt{makeatletter & makeatother}, page 115).

This example of \texttt{@ifstar} defines the command \texttt{ciel} and a variant \texttt{ciel*}. Both have one required argument. A call to \texttt{ciel{blue}} will return "not starry blue sky" while \texttt{ciel*{night}} will return "starry night sky".

\begin{verbatim}
\makeatletter
\newcommand*{\ciel@unstarred}[1]{not starry #1 sky}
\newcommand*{\ciel@starred}[1]{starry #1 sky}
\newcommand*{\ciel}{\@ifstar{\ciel@starred}{\ciel@unstarred}}
\makeatother
\end{verbatim}

In the next example, the starred variant takes a different number of arguments than the unstarred one. With this definition, Agent 007's "My name is \texttt{agentsecret*{Bond}}, \texttt{agentsecret{James}{Bond}}." is equivalent to entering the commands "My name is \texttt{textsc{Bond}}, \texttt{textit{James} textsc{Bond}}."

\begin{verbatim}
\makeatletter
\newcommand*{\agentsecret@unstarred}[2]{\textit{#1} \textsc{#2}}
\newcommand*{\agentsecret@starred}[1]{\textsc{#1}}
\newcommand*{\agentsecret}{\@ifstar{\agentsecret@starred}{\agentsecret@unstarred}}
\makeatother
\end{verbatim}

After a command name, a star is handled similarly to an optional argument. (This differs from environment names in which the star is part of the name itself and as such could be in any position.) Thus, it is technically possible to put any number of spaces between the command and the star. Thus \texttt{agentsecret*{Bond}} and \texttt{agentsecret *{Bond}} are equivalent. However, the standard practice is not to insert any such spaces.

There are two alternative ways to accomplish the work of \texttt{@ifstar}. (1) The \texttt{suffix} package allows the construct \texttt{newcommand\mycommand{unstarred-variant}} followed by \texttt{WithSuffix\newcommand\mycommand*{starred-variant}}. (2) \LaTeX provides the \texttt{xparse} package, which allows this code:

\begin{verbatim}
\NewDocumentCommand\foo{s}{\IfBooleanTF#1 {\texttt{starred-variant}}}\%
\end{verbatim}
\newcounter: Allocating a counter

Synopsis, one of:
\newcounter{\textit{countername}}
\newcounter{\textit{countername}}[\textit{supercounter}]

Globally defines a new counter named \textit{countername} and initialize it to zero (see Chapter 13 [Counters], page 133).

The name \textit{countername} must consist of letters only. It does not begin with a backslash. This name must not already be in use by another counter.

When you use the optional argument [\textit{supercounter}] then the counter \textit{countername} will be reset to zero whenever \textit{supercounter} is incremented. For example, ordinarily \texttt{subsection} is numbered within \texttt{section} so that any time you increment \texttt{section}, either with \texttt{\stepcounter} (see Section 13.7 \texttt{\stepcounter}, page 136) or \texttt{\refstepcounter} (see Section 13.6 \texttt{\refstepcounter}, page 135), then \LaTeX{} will reset \texttt{subsection} to zero.

This example
\newcounter{asuper} \setcounter{asuper}{1}
\newcounter{asub}[asuper] \setcounter{asub}{3} % Note `asuper'
The value of asuper is \arabic{asuper} and of asub is \arabic{asub}.
\stepcounter{asuper}
Now asuper is \arabic{asuper} while asub is \arabic{asub}.

produces ‘The value of asuper is 1 and that of asub is 3’ and ‘Now asuper is 2 while asub is 0’.

If the counter already exists, for instance by entering \texttt{asuper} twice, then you get something like ‘\LaTeX{} Error: Command \texttt{\textbackslash{}asuper} already defined. Or name \texttt{\textbackslash{}end... illegal, see p.192 of the manual.’.

If you use the optional argument then the super counter must already exist. Entering \newcounter{\texttt{\textbackslash{}jh}}[\texttt{lh}] when \texttt{lh} is not a defined counter will get you ‘\LaTeX{} Error: No counter \texttt{\textbackslash{}lh} defined.’

\newlength

Synopsis:
\newlength{\texttt{\textbackslash{}len}}

Allocate a new length register (see Chapter 14 [Lengths], page 137). The required argument \texttt{\textbackslash{}len} has to be a control sequence (see Section 12.1.1 [Control sequences], page 114), and as such must begin with a backslash, \texttt{\textbackslash{}} under normal circumstances. The new register holds rubber lengths such as 72.27pt or \texttt{\textbackslash{}in plus.2in minus.1in} (a \LaTeX{} length register is what plain \TeX{} calls a \texttt{skip} register). The initial value is zero. The control sequence \texttt{\textbackslash{}len} must not be already defined.

An example:
\newlength{\texttt{\textbackslash{}graphichgt}}
If you forget the backslash then you get ‘Missing control sequence inserted’. If the control sequence already exists then you get something like ‘LaTeX Error: Command \graphichgt already defined. Or name \end... illegal, see p.192 of the manual’.

12.7 \newsavebox

Synopsis:
\newsavebox{\cmd}

Define \cmd, the string consisting of a backslash followed by \cmd, to refer to a new bin for storing material. These bins hold material that has been typeset, to use multiple times or to measure or manipulate (see Chapter 20 [Boxes], page 197). The bin name \cmd is required, must start with a backslash, \, and must not already be a defined command. This command is fragile (see Section 12.11 [\protect], page 123).

This allocates a bin and then puts typeset material into it.
\newsavebox{\logobox}
\savebox{\logobox}{LoGo}

Our logo is \usebox{\logobox}.

The output is ‘Our logo is LoGo’.

If there is an already defined bin then you get something like ‘LaTeX Error: Command \logobox already defined. Or name \end... illegal, see p.192 of the manual’.

The allocation of a box is global.

12.8 \newenvironment & \renewenvironment

Synopses, one of:
\newenvironment{\env}{beg\{def\}}{end\{def\}}
\newenvironment{\env}[\nargs]{beg\{def\}}{end\{def\}}
\newenvironment{\env}[\nargs][\optargdefault]{beg\{def\}}{end\{def\}}
\newenvironment*{\env}{beg\{def\}}{end\{def\}}
\newenvironment*{\env}[\nargs]{beg\{def\}}{end\{def\}}
\newenvironment*{\env}[\nargs][\optargdefault]{beg\{def\}}{end\{def\}}

or one of these.
\renewenvironment{\env}{beg\{def\}}{end\{def\}}
\renewenvironment{\env}[\nargs]{beg\{def\}}{end\{def\}}
\renewenvironment{\env}[\nargs][\optargdefault]{beg\{def\}}{end\{def\}}
\renewenvironment*{\env}{beg\{def\}}{end\{def\}}
\renewenvironment*{\env}[\nargs]{beg\{def\}}{end\{def\}}
\renewenvironment*{\env}[\nargs][\optargdefault]{beg\{def\}}{end\{def\}}

Define or redefine the environment \env, that is, create the construct \begin{\env} ... \body ... \end{\env}.

The starred form of these commands requires that the arguments not contain multiple paragraphs of text. However, the body of these environments can contain multiple paragraphs.

\env Required; the environment name. It consists only of letters or the * character, and thus does not begin with backslash, \. It must not begin with the string
end. For \texttt{\newenvironment}, the name \texttt{env} must not be the name of an already existing environment, and also the command \texttt{\env} must be undefined. For \texttt{\renewenvironment}, \texttt{env} must be the name of an existing environment.

\textbf{nargs} Optional; an integer from 0 to 9 denoting the number of arguments of that the environment takes. When you use the environment these arguments appear after the \texttt{\begin}, as in \texttt{\begin\{env\}\{arg1\} ... \{argn\}}. Omitting this is equivalent to setting it to 0; the environment will have no arguments. When redefining an environment, the new version can have a different number of arguments than the old version.

\textbf{optargdefault} Optional; if this is present then the first argument of the defined environment is optional, with default value \texttt{optargdefault} (which may be the empty string). If this is not in the definition then the environment does not take an optional argument.

That is, when \texttt{optargdefault} is present in the definition of the environment then you can start the environment with square brackets, as in \texttt{\begin\{env\}[optval]\{\ldots\} \ldots \end\{env\}}. In this case, within \texttt{begdefn} the parameter \#1 is set to the value of \texttt{optval}. If you call \texttt{\begin\{env\}} without square brackets, then within \texttt{begdefn} the parameter \#1 is set to the value of the default \texttt{optargdefault}. In either case, any required arguments start with \#2.

Omitting \texttt{[myval]} in the call is different than having the square brackets with no contents, as in \texttt{[]} . The former results in \#1 expanding to \texttt{optargdefault}; the latter results in \#1 expanding to the empty string.

\textbf{begdef} Required; the text expanded at every occurrence of \texttt{\begin\{env\}}. Within \texttt{begdef}, the parameters \#1, \#2, ... \#nargs, are replaced by the values that you supply when you call the environment; see the examples below.

\textbf{enddef} Required; the text expanded at every occurrence of \texttt{\end\{env\}}. This may not contain any parameters, that is, you cannot use \#1, \#2, etc., here (but see the final example below).

All environments, that is to say the \texttt{begdef} code, the environment body, and the \texttt{enddef} code, are processed within a group. Thus, in the first example below, the effect of the \texttt{\small} is limited to the quote and does not extend to material following the environment.

If you try to define an environment and the name has already been used then you get something like ‘\LaTeX Error: Command \texttt{fred} already defined. Or name \texttt{\end... illegal, see p.192 of the manual’. If you try to redefine an environment and the name has not yet been used then you get something like ‘\LaTeX Error: Environment \texttt{hank undefined.’.}

This example gives an environment like \LaTeX’s \texttt{quotation} except that it will be set in smaller type.

\begin{verbatim}
\newenvironment{smallquote}{\%
  \small\begin{quotation}
  \%
  }{\%
  \end{quotation}
}
\end{verbatim}
This has an argument, which is set in boldface at the start of a paragraph.
\newenvironment{point}[1]{%
\noindent\textbf{#1}\}%
\}

This one shows the use of a optional argument; it gives a quotation environment that cites the author.
\newenvironment{citequote}[1][Shakespeare]{%
\begin{quotation}
\noindent\textit{#1}:
}\end{quotation}
}
The author's name is optional, and defaults to 'Shakespeare'. In the document, use the environment like this.
\begin{citequote}[Lincoln]
...
\end{citequote}

The final example shows how to save the value of an argument to use in \enddef, in this case in a box (see Section 20.5 [\sbox & \savebox], page 201).
\newsavebox{\quoteauthor}
\newenvironment{citequote}[1][Shakespeare]{%
\sbox\quoteauthor{#1}%
\begin{quotation}
}\hspace{1em plus 1fill}---\usebox{\quoteauthor}
\end{quotation}
}

\newtheoremm

Synopses:
\newtheorem{name}{title}
\newtheorem{name}[numbered_within]{title}
\newtheorem{name}[numbered_like]{title}

Define a new theorem-like environment. You can specify one of \textit{numbered\_within} and \textit{numbered\_like}, or neither, but not both.

The first form, \newtheorem{name}{title}, creates an environment that will be labelled with \textit{title}; see the first example below.

The second form, \newtheorem{name}[numbered\_within]{title}, creates an environment whose counter is subordinate to the existing counter \textit{numbered\_within}, so this counter will be reset when \textit{numbered\_within} is reset. See the second example below.

The third form \newtheorem{name}[numbered\_like]{title}, with optional argument between the two required arguments, creates an environment whose counter will share the previously defined counter \textit{numbered\_like}. See the third example.
This command creates a counter named \textit{name}. In addition, unless the optional argument \textit{numbered_like} is used, inside of the theorem-like environment the current \texttt{ref} value will be that of \texttt{thenumbered_within} (see Section 7.3 \texttt{[ref]}, page 49).

This declaration is global. It is fragile (see Section 12.11 \texttt{[protect]}, page 123).

Arguments:

\texttt{name} The name of the environment. It is a string of letters. It must not begin with a backslash, \texttt{	extbackslash}. It must not be the name of an existing environment, and the command name \texttt{name} must not already be defined.

\texttt{title} The text to be printed at the beginning of the environment, before the number. For example, ‘Theorem’.

\texttt{numbered_within} Optional; the name of an already defined counter, usually a sectional unit such as \texttt{chapter} or \texttt{section}. When the \texttt{numbered_within} counter is reset then the \texttt{name} environment’s counter will also be reset.

If this optional argument is not used then the command \texttt{thename} is set to \texttt{arabic\{name\}}.

\texttt{numbered_like} Optional; the name of an already defined theorem-like environment. The new environment will be numbered in sequence with \texttt{numbered_like}.

Without any optional arguments the environments are numbered sequentially. The example below has a declaration in the preamble that results in ‘Definition 1’ and ‘Definition 2’ in the output.

\begin{verbatim}
\newtheorem{defn}{Definition}
\begin{document}
\section{...}
\begin{defn}
First def
\end{defn}

\section{...}
\begin{defn}
Second def
\end{defn}
\end{document}
\end{verbatim}

This example has the same document body as the prior one. But here \texttt{newtheorem}’s optional argument \texttt{numbered_within} is given as \texttt{section}, so the output is like ‘Definition 1.1’ and ‘Definition 2.1’.

\begin{verbatim}
\newtheorem{defn}{Definition}[section]
\begin{document}
\section{...}
\begin{defn}
First def
\end{defn}
\end{document}
\end{verbatim}
In the next example there are two declarations in the preamble, the second of which calls for the new \texttt{thm} environment to use the same counter as \texttt{defn}. It gives ‘Definition 1.1’, followed by ‘Theorem 2.1’ and ‘Definition 2.2’.

\begin{verbatim}
\newtheorem{defn}{Definition}
\newtheorem{thm}[defn]{Theorem}
\begin{document}
\section{...}
\begin{defn}
First def
\end{defn}
\section{...}
\begin{thm}
First thm
\end{thm}
\begin{defn}
Second def
\end{defn}
\end{document}
\end{verbatim}

12.10 \newfont

This command is obsolete. This description is here only to help with old documents. New documents should define fonts in families through the New Font Selection Scheme which allows you to, for example, associate a boldface with a roman (see Chapter 4 [Fonts], page 14).

Synopsis:

\begin{verbatim}
\newfont{\cmd}{font description}
\end{verbatim}

Define a command \texttt{\cmd} that will change the current font. The control sequence must not already be defined. It must begin with a backslash, \.

The \textit{font description} consists of a \texttt{fontname} and an optional \textit{at clause}. \LaTeX{} will look on your system for a file named \texttt{fontname.tfm}. The at clause can have the form either \texttt{at dimen} or \texttt{scaled factor}, where a \texttt{factor} of ‘1000’ means no scaling. For \LaTeX{}’s purposes, all this does is scale all the character and other font dimensions relative to the font’s design size, which is a value defined in the .tfm file.

This defines two equivalent fonts and typesets a few characters in each.

\begin{verbatim}
\newfont{\testfontat}{cmb10 at 11pt}
\newfont{\testfontscaled}{cmb10 scaled 1100}
\testfontat abc
\testfontscaled abc
\end{verbatim}
12.11 \protect

All \LaTeX\ commands are either fragile or robust. A fragile command can break when it is used in the argument to certain other commands, typically those that write material to the table of contents, the cross-reference file, etc. To prevent fragile commands from causing errors, one solution is to precede them with the command \protect.

For example, when \LaTeX\ runs the \section{section name} command it writes the section name text to the .aux auxiliary file, moving it there for use elsewhere in the document such as in the table of contents. Such an argument that is used in multiple places is referred to as a moving argument. A command is fragile if it can expand during this process into invalid \TeX\ code. Some examples of moving arguments are those that appear in the \caption{...} command (see Section 8.10 [figure], page 60), in the \thanks{...} command (see Section 18.1 [\maketitle], page 179), and in \@-expressions in the \tabular\ and \array\ environments (see Section 8.23 [\tabular], page 85).

If you get strange errors from commands used in moving arguments, try preceding it with \protect. Each fragile command must be protected with their own \protect.

Although usually a \protect\ command doesn’t hurt, length commands such as \parindent should not be preceded by a \protect\ command (see Chapter 14 [Lengths], page 137. Nor can a \protect\ command be used in the argument to \addtocounter\ or \setcounter\ command (see Section 13.4 [\setcounter], page 135, and Section 13.5 [\addtocounter], page 135. These commands are already robust.

As of the October 2019 release of \LaTeX\ (https://www.latex-project.org/news/\latex2e-news/ltnews30.pdf), most commands that had been previously fragile were fixed to be robust. For example, any command taking an optional argument, such as \root\ or \raisebox, was fragile, but is now robust. Similarly, $(\ldots)$ math was fragile and is now robust ($\ldots$) has always been robust).

Perhaps the most commonly used remaining fragile command is \verb; for example,
\begin{verbatim}
\begin{figure}
...
\caption{This \verb\command\ causes an error.}
\end{figure}
\end{verbatim}

Adding \protect does not help here. It’s usually feasible to rewrite the caption (or section heading or whatever) to use \texttt, often the simplest solution.

Alternatively, to use \verb, you can apply the \cprotect\ command from \cprotect\ package (https://ctan.org/pkg/cprotect) to the \caption:
\begin{verbatim}
\cprotect\caption{This \verb\command\ is ok with \verb\cprotect\!}
\end{verbatim}
\cprotect also allows use of \begin...\end environments in moving arguments, where they are normally not allowed, via a similar prefix command \cprotEnv.

12.12 \ignorespaces & \ignorespacesafterend

Synopsis:
\begin{verbatim}
\ignorespaces
or
\ignorespacesafterend
\end{verbatim}
Both commands cause \LaTeX{} to ignore blanks (that is, characters of catcode 10 such as space or tabulation) after the end of the command up to the first box or non-blank character. The first is a primitive command of \TeX{}, and the second is \LaTeX{}-specific.

The \texttt{\ignorespaces} is often used when defining commands via \texttt{\newcommand}, or \texttt{\newenvironment}, or \texttt{\def}. The example below illustrates. It allows a user to show the points values for quiz questions in the margin but it is inconvenient because, as shown in the \texttt{enumerate} list, users must not put any space between the command and the question text.

\begin{verbatim}
\newcommand{\points}[1]{\makebox[0pt]{\makebox[10em][l]{#1~pts}}}
\begin{enumerate}
  \item\points{10}no extra space output here
  \item\points{15} extra space between the number and the `extra'
\end{enumerate}
\end{verbatim}

The solution is to change to this.

\begin{verbatim}
\newcommand{\points}[1]{\makebox[0pt]{\makebox[10em][l]{#1~pts}}\ignorespaces}
\end{verbatim}

A second example shows blanks being removed from the front of text. The commands below allow a user to uniformly attach a title to names. But, as given, if a title accidentally starts with a space then \texttt{\fullname} will reproduce that.

\begin{verbatim}
\newcommand{\honorific}[1]{\def\honorific{#1}} % remember title
\newcommand{\fullname}[1]{\honorific~#1} % put title before name
\begin{tabular}{|l|}
  \honorific{Mr/Ms} \fullname{Jones} \ \ % no extra space here
  \honorific{ Mr/Ms} \ fullname{Jones} % extra space before title
\end{tabular}
\end{verbatim}

To fix this, change to \texttt{\newcommand{\fullname}[1]{\ignorespaces\honorific~#1}}.

The \texttt{\ignorespaces} is also often used in a \texttt{\newenvironment} at the end of the \texttt{begin} clause, as in \texttt{\begin{newenvironment}{env name}{... \ignorespaces}{...}}.

To strip blanks off the end of an environment use \texttt{\ignorespacesafterend}. An example is that this will show a much larger vertical space between the first and second environments than between the second and third.

\begin{verbatim}
\newenvironment{eq}{\begin{equation}}{\end{equation}}
\begin{eq}
e=mc^2
\end{eq}
\begin{equation}F=ma
\end{equation}
\begin{equation}E=IR
\end{equation}
\end{verbatim}

Putting a comment character \% immediately after the \texttt{\end{eq}} will make the vertical space disappear, but that is inconvenient. The solution is to change to \texttt{\newenvironment{eq}{\begin{equation}}{\end{equation}\ignorespacesafterend}}.
12.13 xspace package

This is an add-on package, not part of core \TeX. Synopsis:

\usepackage{xspace}
\newcommand{...}{...\xspace}

The \xspace macro, when used at the end of a command definition, adds a space unless the command is followed by certain punctuation characters.

After a control sequence that is a control word (see Section 12.1.1 [Control sequences], page 114, as opposed to control symbols such as \$), \TeX gobbles blank characters. Thus, in the first sentence below, the output has ‘Vermont’ placed snugly against the period, without any intervening space, despite the space in the input.

\newcommand{\VT}{Vermont}
Our college is in \VT.
\VT{} summers are nice.

But because of the gobbling, the second sentence needs the empty curly braces or else there would be no space separating ‘Vermont’ from ‘summers’. (Many authors instead use a backslash-space \ for this. See Section 19.6 [(SPACE)], page 188.)

The xspace package provides \xspace. It is for writing commands which are designed to be used mainly in text. It must be placed at the very end of the definition of these commands. It inserts a space after that command unless what immediately follows is in a list of exceptions. In this example, the empty braces are not needed.

\newcommand{\VT}{Vermont\xspace}
Our college is in \VT.
\VT{} summers are nice.

The default exception list contains the characters ,.,’/?;&:!~-, the open curly brace and the backslash-space command discussed above, and the commands \footnote or \footnotemark. You can add to that list as with \xspaceaddexceptions{\myfni \myfnii} which adds \myfni and \myfnii to the list; and you can remove from that list as with \xspaceremoveexception{!}.

A comment: many experts prefer not to use \xspace. Putting it in a definition means that the command will usually get the spacing right. But it isn’t easy to predict when to enter empty braces because \xspace will get it wrong, such as when it is followed by another command, and so \xspace can make editing material harder and more error-prone than instead of always inserting the empty braces.

12.14 Class and package commands

These are commands designed to help writers of classes or packages.

12.14.1 \AtBeginDvi & \AtEndDvi

Synopsis:

\AtBeginDvi{code}
\AtEndDvi{code}
\texttt{\AtBeginDvi} saves, in a box register, \textit{code} to be executed at the beginning of the shipout of the first page of the document. Despite the name, it applies to DVI, PDF, and XDV output. It fills the \texttt{shipout/firstpage} hook; new code should use that hook directly.

Similarly, \texttt{\AtEndDvi} (previously available only with the \texttt{atenddvi} package) is code executed when finalizing the main output document.

\subsection*{12.14.2 \texttt{\AtEndOfClass} \& \texttt{\AtEndOfPackage}}

Synopses:
\begin{verbatim}
\AtEndOfClass{code}
\AtEndOfClass{code}
\AtEndOfPackage{code}
\AtEndOfPackage{code}
\end{verbatim}

Hooks to insert \textit{code} to be executed when \LaTeX{} finishes processing the current class resp. package.

These hooks can be used multiple times; each \textit{code} segment will be executed in the order called. Many packages and classes use these commands.

See also Section 8.6.1 [\texttt{\AtBeginDocument}], page 57.

\subsection*{12.14.3 \texttt{\CheckCommand}}

Synopsis:
\begin{verbatim}
\CheckCommand{cmd}[num][default]{definition}
\CheckCommand*{cmd}{num}[default]{definition}
\end{verbatim}

Like \texttt{\newcommand} (see Section 12.1 [\texttt{\newcommand \& \renewcommand}], page 112) but does not define \textit{cmd}; instead it checks that the current definition of \textit{cmd} is exactly as given by \textit{definition} and is or is not \texttt{\long} as expected. A long command is a command that accepts \texttt{\par} within an argument.

With the unstarred version of \texttt{\CheckCommand}, \textit{cmd} is expected to be \texttt{\long}; with the starred version, \textit{cmd} must not be \texttt{\long}

\texttt{\CheckCommand} raises an error when the check fails. This allows you to check before you start redefining \textit{cmd} yourself that no other package has already redefined this command.

\subsection*{12.14.4 \texttt{\ClassError} \& \texttt{\PackageError} and others}

Produce error, warning, and informational messages for classes:
\begin{verbatim}
\ClassError{class name}{error-text}{help-text}
\ClassWarning{class name}{warning-text}
\ClassWarningNoLine{class name}{warning-text}
\ClassInfo{class name}{info-text}
\ClassInfoNoLine{class name}{info-text}
\end{verbatim}

and the same for packages:
\begin{verbatim}
\PackageError{package name}{error-text}{help-text}
\PackageWarning{package name}{warning-text}
\PackageWarningNoLine{package name}{warning-text}
\PackageInfo{package name}{info-text}
\PackageInfoNoLine{package name}{info-text}
\end{verbatim}

For \texttt{\ClassError} and \texttt{\PackageError} the message is \textit{error-text}, followed by \LaTeX{}'s ‘?’ error prompt. If the user then asks for help by typing \texttt{h}, they see the \textit{help text}.
The four \texttt{Warning} commands are similar except that they write \texttt{warning-text} on the screen with no error prompt. The four \texttt{Info} commands write \texttt{info-text} only in the transcript file. The \texttt{NoLine} versions omit the number of the line generating the message, while the other versions do show that number.

To format the messages, including the \texttt{help-text}: use \texttt{\protect} to stop a command from expanding, get a line break with \texttt{\MessageBreak}, and get a space with \texttt{\space} when a space character is ignored, most commonly after a command.

\LaTeX{} appends a period to the messages.

\subsection*{12.14.5 \texttt{CurrentOption}}

Expands to the name of the option currently being processed. This can only be used within the \texttt{code} argument of either \texttt{\DeclareOption} or \texttt{\DeclareOption*}.

\subsection*{12.14.6 \texttt{DeclareOption}}

Synopsis:

\begin{verbatim}
\DeclareOption{option}{code}
\DeclareOption*[option]{code}
\end{verbatim}

Define an option a user can include in their \texttt{\documentclass} command. For example, a class \texttt{smcmemo} could have an option \texttt{logo} allowing users to put the institutional logo on the first page. The document would start with \texttt{\documentclass[logo]{smcmemo}}. To enable this, the class file must contain \texttt{\DeclareOption{logo}{code}} (and later, \texttt{\ProcessOptions}).

If you request an option that has not been declared, by default this will produce a warning like \texttt{Unused global option(s): [badoption]}. This can be changed by using \texttt{\DeclareOption*[code]}, which executes \texttt{code} for any unknown option.

For example, many classes extend an existing class, using code such as \texttt{\LoadClass\{article\}} (see [\texttt{\LoadClass}], page 129). In this case, it makes sense to pass any otherwise-unknown options to the underlying class, like this:

\begin{verbatim}
\DeclareOption*{}
\PassOptionsToClass{\CurrentOption}{article}"
\end{verbatim}

As another example, our class \texttt{smcmemo} might allow users to keep lists of memo recipients in external files, so the user could invoke \texttt{\documentclass[math]{smcmemo}} and it will read the file \texttt{math.memo}. This code inputs the file if it exists, while if it doesn’t, the option is passed to the \texttt{article} class:

\begin{verbatim}
\DeclareOption*{\InputIfFileExists{\CurrentOption.memo}}{}
\PassOptionsToClass{\CurrentOption}{article}}
\end{verbatim}

\subsection*{12.14.7 \texttt{DeclareRobustCommand}}

Synopsis:

\begin{verbatim}
\DeclareRobustCommand{cmd}[num][default]{definition}
\DeclareRobustCommand*[same parameters
\end{verbatim}
The \DeclareRobustCommand and its starred form are generally like \newcommand and \newcommand* (see Section 12.1 [\newcommand & \renewcommand], page 112), with the addition that they define a so-called robust command, even if some code within the definition is fragile. (For a discussion of robust and fragile commands, see Section 12.11 [\protect], page 123.)

Also unlike \newcommand, these do not give an error if macro cmd already exists; instead, a log message is put into the transcript file if a command is redefined. Thus, \DeclareRobustCommand can be used to define new robust commands or to redefine existing commands, making them robust.

The starred form, \DeclareRobustCommand*, disallows the arguments from containing multiple paragraphs, just like the starred form of \newcommand and \renewcommand. The meaning of the arguments is the same.

Commands defined this way are a bit less efficient than those defined using \newcommand so unless the command’s data is fragile and the command is used within a moving argument, use \newcommand.

Related to this, the etoolbox package offers three commands and their starred forms: \newrobustcmd(*) \renewrobustcmd(*), and \providerobustcmd(*). They are similar to \newcommand, \renewcommand, and \providecommand and their own starred forms, but define a robust cmd. They have two possible advantages compared to \DeclareRobustCommand:

1. They use the low-level e-\TeX protection mechanism rather than the higher-level \LaTeX \protect mechanism, so they do not incur the slight loss of performance mentioned above, and

2. They make the same distinction between \new..., \renew..., and \provide..., as the standard commands. That is, they do not just write a log message when you redefine cmd that already exists; you need to use either \renew... or \provide..., or you get an error. This may or may not be a benefit.

\begin{align*}
\textbf{12.14.8} & \textbf{\\textbackslash ExecuteOptions} \\
& \textbf{Synopsis:} \textbackslash ExecuteOptions\{option-list\} \\
& \text{For each option option \textbackslash in option-list, in order, this command executes the command} \textbackslash ds@option. \text{If this command is not defined then that option is silently ignored.} \\
& \text{This can be used to provide a default option list before \textbackslash ProcessOptions. For example, if in a class file you want the default to be 11pt fonts then you could specify} \textbackslash ExecuteOptions\{11pt\}\textbackslash ProcessOptions;\textbackslash relax.
\end{align*}

\begin{align*}
\textbf{12.14.9} \textbf{\\textbackslash IfFileExists \& \textbackslash InputIfFileExists} \\
& \textbf{Synopses:} \textbackslash IfFileExists\{filename\}\{true-code\}\{false-code\} \\
& \textbackslash InputIfFileExists\{filename\}\{true-code\}\{false-code\} \\
& \textbackslash IfFileExists\text{executes true-code if \LaTeX finds the file filename or false-code otherwise.}\text{In the first case it executing true-code and then inputs the file. Thus the command} \\
& \textbackslash IfFileExists\{img.pdf\}\{\%} \\
& \textbackslash includegraphics\{img.pdf\}\}
\end{align*}
This command looks for the file in all search paths that \TeX uses, not only in the current directory. To look only in the current directory do something like \IfFileExists{./filename}{true-code}{false-code}. If you ask for a filename without a .tex extension then \TeX will first look for the file by appending the .tex; for more on how \TeX handles file extensions see Section 24.3 [\input], page 232.

\InputIfFileExists is similar, but, as the name states, automatically \input\ filename if it exists. The true-code is executed just before the \input; if the file doesn’t exist, the false-code is executed. An example:

\InputIfFileExists{mypkg.cfg}
  {\PackageInfo{Loading mypkg.cfg for configuration information}}
  {\PackageInfo{No mypkg.cfg found}}

12.14.10 \LoadClass & \LoadClassWithOptions

Synopses:

\LoadClass[options-list]{class-name}[release-date]
\LoadClassWithOptions{class-name}[release-date]

Load a class, as with \documentclass[options-list]{class-name}[release-date]. An example: \LoadClass{twoside}{article}.

The options-list, if present, is a comma-separated list. The release-date is also optional. If present it must have the form YYYY/MM/DD.

If you request release-date and the date of the package installed on your system is earlier, then you get a warning on the screen and in the log like this:

You have requested, on input line 4, version `2038/01/19' of document class article, but only version `2014/09/29 v1.4h Standard LaTeX document class' is available.

The command version \LoadClassWithOptions uses the list of options for the current class. This means it ignores any options passed to it via \PassOptionsToClass. This is a convenience command that lets you build classes on existing ones, such as the standard article class, without having to track which options were passed.

12.14.11 \NeedsTeXFormat

Synopsis:

\NeedsTeXFormat{format}[format-date]

Specifies the format that this class must be run under. Often issued as the first line of a class file, and most often used as: \NeedsTeXFormat{LaTeX2e}. When a document using that class is processed, the format being run must exactly match the format name given, including case. If it does not match then execution stops with an error like ‘This file needs format `LaTeX2e' but this is `plain'.’.

To require a version of the format that you know to have certain features, include the optional format-date on which those features were implemented. If present, it must be in
the form YYYY/MM/DD. If the format version installed on your system is earlier than format

date then you get a warning like this.

You have requested release `2038/01/20' of LaTeX, but only
release `2016/02/01' is available.

12.14.12 \OptionNotUsed

Adds the current option to the list of unused options. Can only be used within the code
argument of either \DeclareOption or \DeclareOption*.

12.14.13 \PassOptionsToClass & \PassOptionsToPackage

Synopses:

\PassOptionsToClass{options}{classname}
\PassOptionsToPackage{option}{pkgname}

Adds the options in the comma-separated list options to the options used by any future
\RequirePackage or \usepackage command for the class classname or the package pkgname,
respectively.

The reason for these commands is that although you may load a package any number of
times with no options, if you can specify options only the first time you load the package.
Loading a package with options more than once will get you an error like Option clash for
package foo.. \LaTeX throws an error even if there is no conflict between the options.

If your own code is bringing in a package twice then you can combine the calls; for
example, replacing the two

\RequirePackage[landscape]{geometry}
\RequirePackage[margins=1in]{geometry}

with the single command

\RequirePackage[landscape,margins=1in]{geometry}

However, suppose you are loading firstpkg and inside that package it loads secondpkg,
and you need secondpkg to be loaded with option draft. Then before load the first package
you must tell \LaTeX about the desired options for the second package, like this:

\PassOptionsToPackage{draft}{secondpkg}
\RequirePackage{firstpkg}

If firstpkg.sty loads an option in conflict with what you want then you may have to alter
its source, or yours.

These commands are useful for general users as well as class and package writers. For
instance, suppose a user wants to load the graphicx package with the option draft and
also wants to use a class foo that loads the graphicx package, but without that option.
The user could start their \LaTeX file with \PassOptionsToPackage{draft}{graphicx}
\documentclass{foo}.

12.14.14 \ProcessOptions

Synopsis:

\ProcessOptions@options
\ProcessOptions*\@options
Execute the code for each option that the user has invoked. Invoke it in the class file as \ProcessOptions\relax (because of the existence of the starred version, described below).

Options come in two types. *Local options* have been specified for this particular package in \usepackage[options], \RequirePackage[options], or the options argument of \PassOptionsToPackage{options}. *Global options* are those given by the class user in \documentclass[options]. If an option is specified both locally and globally then it is local.

When \ProcessOptions is called for a package pkg.sty, the following happens:
1. For each option *option* so far declared with \DeclareOption, \ProcessOptions looks to see if that option is either global or local for pkg. If so, then it executes the declared code. This is done in the order in which these options were given in pkg.sty.
2. For each remaining local option, it executes the command \ds@option if it has been defined somewhere (other than by a \DeclareOption); otherwise, it executes the default option code given in \DeclareOption*. If no default option code has been declared then it gives an error message. This is done in the order in which these options were specified.

When \ProcessOptions is called for a class it works in the same way except that all options are local, and the default code for \DeclareOption* is \OptionNotUsed rather than an error.

The starred version \ProcessOptions* executes the options in the order specified in the calling commands, rather than in the order of declaration in the class or package. For a package, this means that the global options are processed first.

12.14.15 \ProvidesClass & \ProvidesPackage

**Synopses:**

\ProvidesClass{clsname}[release-date [info-text]]
\ProvidesPackage{pkgname}[release-date [info-text]]

Identifies the class or package being defined, printing a message to the screen and the log file.

When you load a class or package, for example with \documentclass{smcmemo} or \usepackage{test}, \LaTeX inputs a file (smcmemo.cls and test.sty, respectively). If the name of the file does not match the class or package name declared in it then you get a warning. Thus, if you invoke \documentclass{smcmemo}, and the file smcmemo.cls has the statement \ProvidesClass{foo} then you get a warning like You have requested document class `smcmemo', but the document class provides `foo'. This warning does not prevent \LaTeX from processing the rest of the class file normally.

If you include the optional argument then you must include a date, before any spaces, of the form YYYY/MM/DD. The rest of the optional argument is free-form, although it traditionally identifies the class. It is written to the screen during compilation and to the log file. Thus, if your file smcmemo.cls contains the line \ProvidesClass{smcmemo}[2008/06/01 v1.0 SMC memo class] and your document's first line is \documentclass{smcmemo} then you will see Document Class: smcmemo 2008/06/01 v1.0 SMC memo class.

The date in the optional argument allows class and package users to ask to be warned if the version of the class or package is earlier than release date. For instance, a user could
enter \documentclass{smcmemo}[2018/10/12] or \usepackage{foo}[[2017/07/07]] to require a class or package with certain features by specifying that it must be released no earlier than the given date. Perhaps more importantly, the date serves as documentation of the last release. (In practice, package users rarely include a date, and class users almost never do.)

12.14.16 \ProvidesFile

Synopsis:

\ProvidesFile{filename}[info-text]

Declare a file other than the main class and package files, such as a configuration or font definition file. It writes the given information to the log file, essentially like \ProvidesClass and \ProvidesPackage (see the previous section).

For example:

\ProvidesFile{smcmemo.cfg}[2017/10/12 config file for smcmemo.cls]

writes this into the log:

File: smcmemo.cfg 2017/10/12 config file for smcmemo.cls

12.14.17 \RequirePackage & \RequirePackageWithOptions

Synopsis:

\RequirePackage[option-list]{pkgname}[release-date]
\RequirePackageWithOptions{pkgname}[release-date]

Load a package, like the command \usepackage (see [Additional packages], page 12). An example:

\RequirePackage[landscape,margin=1in]{geometry}

The initial optional argument option-list, if present, must be a comma-separated list. The trailing optional argument release-date, if present, must have the form YYYY/MM/DD. If the release date of the package as installed on your system is earlier than release-date then you get a warning like ‘You have requested, on input line 9, version `2017/07/03' of package jhtest, but only version `2000/01/01' is available’.

The \RequirePackageWithOptions variant uses the list of options for the current class. This means it ignores any options passed to it via \PassOptionsToClass. This is a convenience command to allow easily building classes on existing ones without having to track which options were passed.

The difference between \usepackage and \RequirePackage is small. The \usepackage command is intended to be used in documents, while \RequirePackage is intended for package and class files. The most significant difference in practice is that \RequirePackage can be used in a document before the \documentclass command, while \usepackage gives an error there. The most common need for this nowadays is for the \DocumentMetadata command (see Section 2.7 \[DocumentMetadata], page 8).

The \TeX{} development team strongly recommends use of these and related commands over Plain \TeX{}’s \input; see the Class Guide (https://ctan.org/pkg/clsguide).
13 Counters

Everything \LaTeX numbers for you has a counter associated with it. The name of the counter is often the same as the name of the environment or command associated with the number, except that the counter’s name has no backslash \. Thus, associated with the \chapter command is the chapter counter that keeps track of the chapter number.

Below is a list of the counters used in \LaTeX’s standard document classes to control numbering.

<table>
<thead>
<tr>
<th>part</th>
<th>paragraph</th>
<th>figure</th>
<th>enumi</th>
</tr>
</thead>
<tbody>
<tr>
<td>chapter</td>
<td>subparagraph</td>
<td>table</td>
<td>enumii</td>
</tr>
<tr>
<td>section</td>
<td>page</td>
<td>footnote</td>
<td>enumiii</td>
</tr>
<tr>
<td>subsection</td>
<td>equation</td>
<td>mpfootnote</td>
<td>enumiv</td>
</tr>
<tr>
<td>subsubsection</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The mpfootnote counter is used by the \footnote command inside of a minipage (see Section 8.18 [minipage], page 71). The counters enumi through enumiv are used in the enumerate environment, for up to four levels of nesting (see Section 8.7 [enumerate], page 57).

Counters can have any integer value but they are typically positive.

New counters are created with \newcounter. See Section 12.5 [\newcounter], page 117.

13.1 \texttt{\textalpha} \texttt{\\textAlpha} \texttt{\textarabic} \texttt{\textroman} \texttt{\\textRoman} \texttt{\textfnsymbol}: Printing counters

Print the value of a counter, in a specified style. For instance, if the counter counter has the value 1 then a \texttt{\textalpha\{counter\}} in your source will result in a lowercase letter a appearing in the output.

All of these commands take a single counter as an argument, for instance, \texttt{\textalpha\{enumi\}}. Note that the counter name does not start with a backslash.

\texttt{\textalpha\{counter\}}

Print the value of counter in lowercase letters: ‘a’, ‘b’, . . . If the counter’s value is less than 1 or more than 26 then you get ‘\LaTeX Error: Counter too large.’

\texttt{\textAlpha\{counter\}}

Print in uppercase letters: ‘A’, ‘B’, . . . If the counter’s value is less than 1 or more than 26 then you get ‘\LaTeX Error: Counter too large.’

\texttt{\textarabic\{counter\}}

Print in Arabic numbers such as ‘5’ or ‘-2’.

\texttt{\textroman\{counter\}}

Print in lowercase roman numerals: ‘i’, ‘ii’, . . . If the counter’s value is less than 1 then you get no warning or error but \LaTeX does not print anything in the output.

\texttt{\textRoman\{counter\}}

Print in uppercase roman numerals: ‘I’, ‘II’, . . . If the counter’s value is less than 1 then you get no warning or error but \LaTeX does not print anything in the output.
\fnsymbol{counter}

Prints the value of counter using a sequence of nine symbols that are traditionally used for labeling footnotes. The value of counter should be between 1 and 9, inclusive. If the counter's value is less than 0 or more than 9 then you get ‘\LaTeX Error: Counter too large’, while if it is 0 then you get no error or warning but \LaTeX does not output anything.

Here are the symbols:

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Command</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>asterisk</td>
<td>\ast</td>
<td>\ast</td>
</tr>
<tr>
<td>2</td>
<td>dagger</td>
<td>\dagger</td>
<td>†</td>
</tr>
<tr>
<td>3</td>
<td>ddagger</td>
<td>\ddagger</td>
<td>‡</td>
</tr>
<tr>
<td>4</td>
<td>section-sign</td>
<td>\S</td>
<td>§</td>
</tr>
<tr>
<td>5</td>
<td>paragraph-sign</td>
<td>\P</td>
<td>¶</td>
</tr>
<tr>
<td>6</td>
<td>double-vert</td>
<td>\parallel</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>double-asterisk</td>
<td>\ast\ast</td>
<td>\ast\ast</td>
</tr>
<tr>
<td>8</td>
<td>double-dagger</td>
<td>\dagger\dagger</td>
<td>††</td>
</tr>
<tr>
<td>9</td>
<td>double-ddagger</td>
<td>\ddagger\ddagger</td>
<td>‡‡</td>
</tr>
</tbody>
</table>

13.2 \usecounter

Synopsis:

\usecounter{counter}

Used in the second argument of the list environment (see Section 8.16 [list], page 65), this declares that list items will be numbered by counter. It initializes counter to zero, and arranges that when \item is called without its optional argument then counter is incremented by \refstepcounter, making its value be the current ref value (see Section 7.3 [ref], page 49). This command is fragile (see Section 12.11 [protect], page 123).

Put in the document preamble, this example makes a new list environment enumerated with testcounter:

\newcounter{testcounter}
\newenvironment{test}{% \begin{list}{}{% \usecounter{testcounter} 
 }{% \end{list} 
}

13.3 \value

Synopsis:

\value{counter}

Expands to the value of the counter counter. (Note that the name of a counter does not begin with a backslash.)

This example outputs ‘Test counter is 6. Other counter is 5.’.

\newcounter{test} \setcounter{test}{5}
\newcounter{other} \setcounter{other}{\value{test}} \addtocounter{test}{1}

Test counter is \arabic{test}.
Other counter is \arabic{other}.

The \value command is not used for typesetting the value of the counter. For that, see Section 13.1 [\alph Alph \arabic \roman \Roman \fnsymbol], page 133.

It is often used in \setcounter or \addtocounter but \value can be used anywhere that \LaTeX expects a number, such as in \hspace{\value{foo}\parindent}. It must not be preceded by \protect (see Section 12.11 [\protect], page 123).

This example inserts \hspace{4\parindent}.
\setcounter{myctr}{3} \addtocounter{myctr}{1} \hspace{\value{myctr}\parindent}

13.4 \setcounter

Synopsis:
\setcounter{counter}{value}

Globally set the counter counter to have the value of the value argument, which must be an integer. Thus, you can set a counter’s value as \setcounter{section}{5}. Note that the counter name does not start with a backslash.

In this example if the counter theorem has value 12 then the second line will print ‘XII’.
\setcounter{exercise}{\value{theorem}}
Here it is in Roman: \Roman{exercise}.

13.5 \addtocounter

Synopsis:
\addtocounter{counter}{value}

Globally increment counter by the amount specified by the value argument, which may be negative.

In this example the section value appears as ‘VII’.
\setcounter{section}{5} \addtocounter{section}{2} Here it is in Roman: \Roman{section}.

13.6 \refstepcounter

Synopsis:
\refstepcounter{counter}

Globally increments the value of counter by one, as does \stepcounter (see Section 13.7 [\stepcounter], page 136). The difference is that this command resets the value of any counter numbered within it. (For the definition of “counters numbered within”, see Section 12.5 [\newcounter], page 117.)
In addition, this command also defines the current \ref value to be the result of \thecounter.

While the counter value is set globally, the \ref value is set locally, i.e., inside the current group.

13.7 \stepcounter

Synopsis:

\stepcounter{counter}

Globally adds one to counter and resets all counters numbered within it. (For the definition of “counters numbered within”, see Section 12.5 [newcounter], page 117.)

This command differs from \restepcounter in that this one does not influence references; that is, it does not define the current \ref value to be the result of \thecounter (see Section 13.6 [restepcounter], page 135).

13.8 \day & \month & \year

LaTeX defines the counter \day for the day of the month (nominally with value between 1 and 31), \month for the month of the year (nominally with value between 1 and 12), and \year for the year. When TeX starts up, they are set from the current values on the system. The related command \today produces a string representing the current day (see Section 23.9 [today], page 228).

They counters are not updated as the job progresses so in principle they could be incorrect by the end. In addition, TeX does no sanity check:

\day=-2 \month=13 \year=-4 \today

gives no error or warning and results in the output ‘-2, -4’ (the bogus month value produces no output).

See Section 28.2 [Command line input], page 261, to force the date to a given value from the command line.
14 Lengths

A *length* is a measure of distance. Many \TeX commands take a length as an argument.

Lengths come in two types. A *rigid length* such as 10pt does not contain a *plus* or *minus* component. (Plain \TeX calls this a *dimen*.) A *rubber length* (what plain \TeX calls a *skip* or *glue*) such as with 1cm plus0.05cm minus0.01cm can contain either or both of those components. In that rubber length, the 1cm is the *natural length* while the other two, the *plus* and *minus* components, allow \TeX to stretch or shrink the length to optimize placement.

The illustrations below use these two commands.

\begin{verbatim}
% make a black bar 10pt tall and #1 wide
\newcommand{\blackbar}{\rule{#1}{10pt}}

% Make a box around #2 that is #1 wide (excluding the border)
\newcommand{\showhbox}{\fboxsep=0pt\fbox{\hbox to #1{#2}}}
\end{verbatim}

This next example uses those commands to show a black bar 100 points long between ‘ABC’ and ‘XYZ’. This length is rigid.

ABC\showhbox{100pt}{\blackbar{100pt}}XYZ

As for rubber lengths, shrinking is simpler one: with 1cm minus 0.05cm, the natural length is 1 cm but \TeX can shrink it down as far as 0.95 cm. Beyond that, \TeX refuses to shrink any more. Thus, below the first one works fine, producing a space of 98 points between the two bars.

ABC\showhbox{300pt}{\blackbar{101pt}\hspace{100pt minus 2pt}\blackbar{101pt}}YYY

But the second one gets a warning like ‘Overfull \hbox (1.0pt too wide) detected at line 17’. In the output the first ‘Y’ is overwritten by the end of the black bar, because the box’s material is wider than the 300 pt allocated, as \TeX has refused to shrink the total to less than 309 points.

Stretching is like shrinking except that if \TeX is asked to stretch beyond the given amount, it will do it. Here the first line is fine, producing a space of 110 points between the bars.

ABC\showhbox{300pt}{\blackbar{95pt}\hspace{100pt plus 10pt}\blackbar{95pt}}YYY

But in the second line \TeX needs a stretch of 10 points and only 1 point was specified. \TeX stretches the space to the required length but it gives you a warning like ‘Underfull \hbox (badness 10000) detected at line 22’. (We won’t discuss badness.)

You can put both stretch and shrink in the same length, as in 1ex plus 0.05ex minus 0.02ex.
If \TeX is setting two or more rubber lengths then it allocates the stretch or shrink in proportion.

\begin{verbatim}
ABC\showhbox{300pt}{%
  \blackbar{100pt}\% left
  \hspace{Opt plus 50pt}\blackbar{80pt}\hspace{Opt plus 10pt}\% middle
  \blackbar{100pt}}YYY \% right
\end{verbatim}

The left and right bars take up 100 points, so the middle needs another 100. The middle bar is 80 points so the two \hspace's must stretch 20 points. Because the two are plus 50pt and plus 10pt, \TeX gets 5/6 of the stretch from the first space and 1/6 from the second.

The plus or minus component of a rubber length can contain a fill component, as in 1in plus2fill. This gives the length infinite stretchability or shrinkability so that \TeX could set it to any distance. Here the two figures will be equally spaced across the page.

\begin{verbatim}
\begin{minipage}{\linewidth}
  \hspace{0pt plus 1fill}\includegraphics{godel.png}%
  \hspace{0pt plus 1fill}\includegraphics{einstein.png}%
  \hspace{0pt plus 1fill}
\end{minipage}
\end{verbatim}

\TeX has three levels of infinity for glue components: fil, fill, and filll. The later ones are more infinite than the earlier ones. Ordinarily document authors only use the middle one (see Section 19.3 \hfill, page 185, and see Section 19.15 \vfill, page 195).

Multiplying a rubber length by a number turns it into a rigid length, so that after \setlength{\ylength}{1in plus 0.2in} and \setlength{\zlength}{3\ylength} then the value of \zlength is 3in.

### 14.1 Units of length

\TeX and \LaTeX know about these units both inside and outside of math mode.

- **pt**  
  Point, 1/72.27 inch. The (approximate) conversion to metric units is 1 point = .35146 mm = .035146 cm.

- **pc**  
  Pica, 12 pt

- **in**  
  Inch, 72.27 pt

- **bp**  
  Big point, 1/72 inch. This length is the definition of a point in PostScript and many desktop publishing systems.

- **mm**  
  Millimeter, 2.845 pt

- **cm**  
  Centimeter, 10 mm

- **dd**  
  Didot point, 1.07 pt

- **cc**  
  Cicero, 12 dd

- **sp**  
  Scaled point, 1/65536 pt

Three other units are defined according to the current font, rather than being an absolute dimension.

- **ex**  
  The x-height of the current font ex, traditionally the height of the lowercase letter x, is often used for vertical lengths.
Similarly \textit{em}, traditionally the width of the capital letter M, is often used for horizontal lengths. This is also often the size of the current font, e.g., a nominal 10 pt font will have $1\text{ em} = 10$ pt. \LaTeX{} has several commands to produce horizontal spaces based on the \textit{em} (see Section 19.1 \textit{\&quad}, page 184).

Finally, in math mode, many definitions are expressed in terms of the math unit \textit{mu}, defined by $1\text{ em} = 18$ \textit{mu}, where the em is taken from the current math symbols family. See Section 16.6 \textit{Spacing in math mode}, page 171.

Using these units can help make a definition work better across font changes. For example, a definition of the vertical space between list items given as \texttt{\setlength{\itemsep}{1ex plus 0.05ex minus 0.01ex}} is more likely to still be reasonable if the font is changed than a definition given in points.

### 14.2 \texttt{\setlength}

**Synopsis:**

\begin{verbatim}
\setlength{\len}{amount}
\end{verbatim}

Set the length \texttt{\len} to \textit{amount}. The length name \texttt{\len} has to be a control sequence (see Section 12.1.1 \textit{Control sequences}, page 114), and as such must begin with a backslash, \texttt{\} under normal circumstances. The \textit{amount} can be a rubber length (see Chapter 14 \textit{Lengths}, page 137). It can be positive, negative or zero, and can be in any units that \LaTeX{} understands (see Section 14.1 \textit{Units of length}, page 138).

Below, with \LaTeX{}’s defaults the first paragraph will be indented while the second will not.

\begin{verbatim}
I told the doctor I broke my leg in two places.
\setlength{\parindent}{0em}
He said stop going to those places.
\end{verbatim}

If you did not declare \texttt{\len} with \texttt{\newlength}, for example if you mistype it as in \texttt{\newlength{\specparindent}\setlength{\secparindent}{...}}, then you get an error like ‘\textit{Undefined control sequence}. <\texttt{\argument}> \texttt{\seccindent}'. If you omit the backslash at the start of the length name then you get an error like ‘\textit{Missing number, treated as zero}'.

### 14.3 \texttt{\addtolength}

**Synopsis:**

\begin{verbatim}
\addtolength{\len}{amount}
\end{verbatim}

Increment the length \texttt{\len} by \textit{amount}. The length name \texttt{\len} has to be a control sequence (see Section 12.1.1 \textit{Control sequences}, page 114), and as such must begin with a backslash, \texttt{\} under normal circumstances. The \textit{amount} is a rubber length (see Chapter 14 \textit{Lengths}, page 137). It can be positive, negative or zero, and can be in any units that \LaTeX{} understands (see Section 14.1 \textit{Units of length}, page 138).

Below, if \texttt{\parskip} starts with the value \texttt{0pt plus 1pt}

\begin{verbatim}
Doctor: how is the boy who swallowed the silver dollar?
\end{verbatim}
\addtolength{\parskip}{1pt}

Nurse: no change.
then it has the value 1pt plus 1pt for the second paragraph.

If you did not declare \len with \newlength, for example if you mistype it as in \newlength{\specparindent}\addtolength{\sepcparindent}{\ldots}, then you get an error like ‘Undefined control sequence. \sepcindent’. If the amount uses some length that has not been declared, for instance if for example you mistype the above as \addtolength{\specparindent}{0.6\praindent}, then you get something like ‘Undefined control sequence. \praindent’. If you leave off the backslash at the start of \len, as in \addtolength{\praindent}{1pt}, then you get something like ‘You can’t use \texttt{the letter p} after \advance’.

14.4 \settodepth

Synopsis:
\settodepth{\len}{text}

Set the length \len to the depth of box that \LaTeX\ gets on typesetting the text argument.
The length name \len has to be a control sequence (see Section 12.1.1 [Control sequences], page 114), and as such must begin with a backslash, \ under normal circumstances.

This will print how low the character descenders go.
\newlength{\alphabetdepth}
\settodepth{\alphabetdepth}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetdepth

If you did not declare \len with \newlength, if for example you mistype the above as \settodepth{\alphabetdepth}{abc\ldots}, then you get something like ‘Undefined control sequence. \alphabetdepth’. If you leave the backslash out of \len, as in \settodepth{\alphabetdepth}{\ldots} then you get something like ‘Missing number, treated as zero. \texttt{to be read again} \setbox’.

14.5 \settoheight

Synopsis:
\settoheight{\len}{text}

Sets the length \len to the height of box that \LaTeX\ gets on typesetting the text argument.
The length name \len has to be a control sequence (see Section 12.1.1 [Control sequences], page 114), and as such must begin with a backslash, \ under normal circumstances.

This will print how high the characters go.
\newlength{\alphabetheight}
\settoheight{\alphabetheight}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetheight

If no such length \len has been declared with \newlength, if for example you mistype as \settoheight{\alphabetheight}{abc\ldots}, then you get something like ‘Undefined control sequence. \alphabetheight’. If you leave the backslash out of \len, as in \settoheight{\alphabetheight}{\ldots} then you get something like ‘Missing number, treated as zero. \texttt{to be read again} \setbox’.
14.6 \settowidth

Synopsis:

\settowidth{\len}{text}

Set the length \len to the width of the box that \LaTeX gets on typesetting the text argument. The length name \len has to be a control sequence (see Section 12.1.1 [Control sequences], page 114), and as such must begin with a backslash, \ under normal circumstances.

This prints the width of the lowercase ASCII alphabet.

\newlength{\alphabetwidth}
\settowidth{\alphabetwidth}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetwidth

If no such length \len has been declared with \newlength, if for example you mistype the above as \settowidth{\alphabetwidth}{abc...}, then you get something like ‘Undefined control sequence. <argument> \alphabetwidth’. If you leave the backslash out of \len, as in \settowidth{alphabetwidth}{...} then you get something like ‘Missing number, treated as zero. <to be read again> \setbox’.

14.7 \stretch

Synopsis:

\stretch{number}

Produces a rubber length with zero natural length and number times \fill units of stretchability (see Chapter 14 [Lengths], page 137). The number can be positive or negative. This command is robust (see Section 12.11 [protect], page 123).

It works for both vertical and horizontal spacing. In this horizontal example, \LaTeX produces three tick marks, and the distance between the first and second is half again as long as the distance between the second and third.

\rule{0.4pt}{1ex}\hspace{\stretch{1.5}}\%  
\rule{0.4pt}{1ex}\hspace{\stretch{1}}\%
\rule{0.4pt}{1ex}

In this vertical example, the ‘We dedicate ...’ will have three times as much space under it as above it.

\newenvironment{dedication}{% in document preamble
\clearpage\thispagestyle{empty}\%
\vspace*{\stretch{1}} % stretchable space at top
\it }
{\vspace{\stretch{3}} % space at bot is 3x as at top
\clearpage
}

\begin{dedication} % in document body
We dedicate this book to our wives.
\end{dedication}
14.8 Expressions

Synopsis, one of:

\numexpr expression
\dimexpr expression
\glueexpr expression
\muglue expression

Any place where you may write an integer, or a \TeX\ dimen, or \TeX\ glue, or muglue, you can instead write an expression to compute that type of quantity.

An example is that \texttt{\the\dimexpr\linewidth-4pt\relax} will produce as output the length that is four points less than width of a line (the only purpose of \texttt{\the} is to show the result in the document). Analogously, \texttt{\romannumeral\numexpr6+3\relax} will produce ‘ix’, and \texttt{\the\glueexpr 5pt\ plus 1pt\ * 2\relax} will produce ‘10.0pt\ plus 2.0pt’.

A convenience here over doing calculations by allocating registers and then using \texttt{\advance}, etc., is that the evaluation of expressions does not involve assignments and can therefore be performed in places where assignments are not allowed. The next example computes the width of the \texttt{\parbox}.

\newlength{\offset}\setlength{\offset}{2em}
\begin{center}
\parbox{\dimexpr\linewidth-\offset*3}{With malice toward none
with charity for all with firmness in the right as God gives us to see
the right let us strive on to finish the work we are in to bind up the
nation’s wounds, to care for him who shall have borne the battle and
for his widow and his orphan \texttt{\textasciitilde}\ to do all which may
achieve and cherish a just and lasting peace among ourselves and with
all nations. ---Abraham Lincoln, Second Inaugural Address, from the
memorial}
\end{center}

The expression consists of one or more terms of the same type (integer, dimension, etc.) that are added or subtracted. A term that is a type of number, dimension, etc., consists of a factor of that type, optionally multiplied or divided by factors. A factor of a type is either a quantity of that type or a parenthesized subexpression. The expression produces a result of the given type, so that \texttt{\numexpr} produces an integer, \texttt{\dimexpr} produces a dimension, etc.

In the quotation example above, changing to \texttt{\dimexpr\linewidth-3*\offset} gives the error \texttt{Illegal unit of measure (pt inserted)}. This is because for \texttt{\dimexpr} and \texttt{\glueexpr}, the input consists of a dimension or glue value followed by an optional multiplication factor, and not the other way around. Thus \texttt{\the\dimexpr 1pt*10\relax} is valid and produces ‘10.0pt’, but \texttt{\the\dimexpr 10*1pt\relax} gives the \texttt{Illegal unit} error.

The expressions absorb tokens and carry out appropriate mathematics up to a \texttt{\relax} (which will be absorbed), or up to the first non-valid token. Thus, \texttt{\the\numexpr2+3\relax\px} will print ‘5px’, because \LaTeX\ reads the \texttt{\numexpr2+3}, which is made up of numbers, and then finds the letter \texttt{p}, which cannot be part of a number. It therefore terminates the expression and produces the ‘5’, followed by the regular text ‘px’.

This termination behavior is useful in comparisons. In \texttt{\ifnum\numexpr\parindent*2<10pt Yes\else No\fi}, the less than sign terminates the expression and the result is ‘No’ (in a standard \LaTeX\ article).
Expressions may use the operators +, −, * and / along with parentheses for subexpressions, (...)\ldots. In glue expressions the plus and minus parts do not need parentheses to be affected by a factor. So $\texttt{\textbackslash the\textbackslash glueexpr 5pt plus 1pt * 2 \textbackslash relax}$ results in ‘10pt plus 2pt’.

\TeX\ will coerce other numerical types in the same way as it does when doing register assignment. Thus $\texttt{\textbackslash the\textbackslash numexpr\textbackslash dimexpr 1pt\textbackslash relax\textbackslash relax}$ will result in ‘65536’, which is 1pt converted to scaled points (see \cite{units of length sp}, page 138, \TeX\’s internal unit) and then coerced into an integer. With a $\texttt{\textbackslash glueexpr}$ here, the stretch and shrink would be dropped. Going the other way, a $\texttt{\textbackslash numexpr}$ inside a $\texttt{\textbackslash dimexpr}$ or $\texttt{\textbackslash glueexpr}$ will need appropriate units, as in $\texttt{\textbackslash the\textbackslash dimexpr\textbackslash numexpr 1 + 2\textbackslash relax pt\textbackslash relax}$, which produces ‘3.0pt’.

The details of the arithmetic: each factor is checked to be in the allowed range, numbers must be less than $2^{31}$ in absolute value, and dimensions or glue components must be less than $2^{14}$ points, or mu, or fil, etc. The arithmetic operations are performed individually, except for a scaling operation (a multiplication immediately followed by a division) which is done as one combined operation with a 64-bit product as intermediate value. The result of each operation is again checked to be in the allowed range.

Finally, division and scaling take place with rounding (unlike \TeX\’s $\texttt{\textbackslash divide}$, which truncates). Thus $\texttt{\textbackslash the\textbackslash dimexpr 5pt*(3/2)\textbackslash relax}$ puts ‘10.0pt’ in the document, because it rounds 3/2 to 2, while $\texttt{\textbackslash the\textbackslash dimexpr 5pt*(4/3)\textbackslash relax}$ produces ‘5.0pt’.
15 Making paragraphs

To start a paragraph, just type some text. To end the current paragraph, put an empty line. This is three paragraphs, the separation of which is made by two empty lines.

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

A paragraph separator can be made of a sequence of at least one blank line, at least one of which is not terminated by a comment. A blank line is a line that is empty or made only of blank characters such as space or tab. Comments in source code are started with a % and span up to the end of line. In the following example the two columns are identical:

```
\documentclass[twocolumn]{article}
\begin{document}
First paragraph.
Second paragraph.
\newpage
First paragraph.

% separator lines may contain blank characters.

Second paragraph.
\end{document}
```

Once \LaTeX{} has gathered all of a paragraph’s contents it divides that content into lines in a way that is optimized over the entire paragraph (see Chapter 9 [Line breaking], page 99).

There are places where a new paragraph is not permitted. Don’t put a blank line in math mode (see Chapter 17 [Modes], page 177); here the blank line before the \end{equation}

```
\begin{equation}
2^{-|S|} > |S|
\end{equation}
```

will get you the error ‘Missing $ inserted’. Similarly, the blank line in this \section argument

```
\section{aaa

bbb}
```

gets ‘Runaway argument? {aaa ! Paragraph ended before \section was complete’.
15.1 \par

Synopsis (note that while reading the input \TeX converts any sequence of one or more blank lines to a \par, Chapter 15 [Making paragraphs], page 144):

\par

End the current paragraph. The usual way to separate paragraphs is with a blank line but the \par command is entirely equivalent. This command is robust (see Section 12.11 [\protect], page 123).

This example uses \par rather than a blank line simply for readability.

\newcommand{\syllabusLegalese}{\% \whatCheatingIs\par\whatHappensWhenICatchYou}

In LR mode the \par command does nothing and is ignored. In paragraph mode, the \par command terminates paragraph mode, switching \LaTeX to vertical mode (see Chapter 17 [Modes], page 177).

You cannot use the \par command in a math mode. You also cannot use it in the argument of many commands, such as the sectioning commands, e.g. \section (see Chapter 15 [Making paragraphs], page 144, and Section 12.1 [\newcommand & \renewcommand], page 112).

The \par command is not the same as the \paragraph command. The latter is, like \section or \subsection, a sectioning command used by the \LaTeX document standard classes (see Section 6.5 [\subsubsection & \paragraph & \subparagraph], page 41).

The \par command is not the same as \newline or the line break double backslash, \\. The difference is that \par ends the paragraph, not just the line, and also triggers the addition of the between-paragraph vertical space \parskip (see Section 15.3 [\parindent & \parskip], page 146).

The output from this example

xyz

\setlength{\parindent}{3in}
\setlength{\parskip}{5in}
\noindent test\indent test1\par test2

is: after 'xyz' there is a vertical skip of 5 inches and then 'test' appears, aligned with the left margin. On the same line, there is an empty horizontal space of 3 inches and then 'test1' appears. Finally, there is a vertical space of 5 inches, followed by a fresh paragraph with a paragraph indent of 3 inches, and then \LaTeX puts the text 'test2'.

15.2 \indent & \noindent

Synopsis:

\indent

or

\noindent

Go into horizontal mode (see Chapter 17 [Modes], page 177). The \indent command first outputs an empty box whose width is \parindent. These commands are robust (see Section 12.11 [\protect], page 123).
Ordinarily you create a new paragraph by putting in a blank line. See Section 15.1 \[\text{\par}\], page 145, for the difference between this command and \par. To start a paragraph without an indent, or to continue an interrupted paragraph, use \noindent.

In the middle of a paragraph the \noindent command has no effect, because \LaTeX{} is already in horizontal mode there. The \indent command’s only effect is to output a space.

This example starts a fresh paragraph.

... end of the prior paragraph.

\noindent This paragraph is not indented.

and this continues an interrupted paragraph.

The data

\begin{center}
\begin{tabular}{rl} ... \end{tabular}
\end{center}

\noindent shows this clearly.

To omit indentation in the entire document put $\setlength{\parindent}{0pt}$ in the preamble. If you do that, you may want to also set the length of spaces between paragraphs, $\setlength{\parskip}$ (see Section 15.3 [\parindent & \parskip], page 146).

Default \LaTeX{} styles have the first paragraph after a section that is not indented, as is traditional typesetting in English. To change that, look on CTAN for the package indentfirst.

**15.3 \parindent & \parskip**

Synopsis:

$\setlength{\parindent}{\text{horizontal \_len}}$

$\setlength{\parskip}{\text{vertical \_len}}$

Both are rubber lengths (see Chapter 14 [Lengths], page 137). They affect the indentation of ordinary paragraphs, not paragraphs inside minipages (see Section 8.18 [minipage], page 71), and the vertical space between paragraphs, respectively.

For example, if this is put in the preamble:

$\setlength{\parindent}{0em}$

$\setlength{\parskip}{1ex}$

The document will have paragraphs that are not indented, but instead are vertically separated by about the height of a lowercase ‘x’.

In \LaTeX{} standard class documents, the default value for \parindent in one-column documents is 15pt when the default text size is 10pt, 17pt for 11pt, and 1.5em for 12pt. In two-column documents it is 1em. (These values are set before \LaTeX{} calls $\text{\normalfont}$ so em is derived from the default font, Computer Modern. If you use a different font then to set \parindent to 1em matching that font, put $\text{\AtBeginDocument{\setlength{\parindent}{1em}}}$ in the preamble.)

The default value for \parskip in \LaTeX{}’s standard document classes is 0pt plus1pt.
15.4 Marginal notes

Synopsis, one of:

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

Create a note in the margin. The first line of the note will have the same baseline as the line in the text where the \marginpar occurs.

The margin that \LaTeX uses for the note depends on the current layout (see Section 3.1 [Document class options], page 10) and also on \reversemarginpar (see below). If you are using one-sided layout (document option oneside) then it goes in the right margin. If you are using two-sided layout (document option twoside) then it goes in the outside margin. If you are in two-column layout (document option twocolumn) then it goes in the nearest margin.

If you declare \reversemarginpar then \LaTeX will place subsequent marginal notes in the opposite margin to that given in the prior paragraph. Revert that to the default position with \normalmarginpar.

When you specify the optional argument left then it is used for a note in the left margin, while the mandatory argument right is used for a note in the right margin.

Normally, a note’s first word will not be hyphenated. You can enable hyphenation there by beginning left or right with \hspace{0pt}.

These parameters affect the formatting of the note:

\marginparpush
Minimum vertical space between notes; default ‘7pt’ for ‘12pt’ documents, ‘5pt’ else. See also [page layout parameters marginparpush], page 29.

\marginparspace
Horizontal space between the main text and the note; default ‘11pt’ for ‘10pt’ documents, ‘10pt’ else.

\marginparwidth
Width of the note itself; default for a one-sided ‘10pt’ document is ‘90pt’, ‘83pt’ for ‘11pt’, and ‘68pt’ for ‘12pt’; ‘17pt’ more in each case for a two-sided document. In two column mode, the default is ‘48pt’.

The standard \LaTeX routine for marginal notes does not prevent notes from falling off the bottom of the page.
16 Math formulas

Produce mathematical text by putting \( \text{LaTeX} \) into math mode or display math mode (see Chapter 17 [Modes], page 177). This example shows both.

The wave equation for \( u \) is

\[
\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u
\]

where \( \nabla^2 \) is the spatial Laplacian and \( c \) is constant.

Math mode is for inline mathematics. In the above example it is invoked by the starting \( and finished by the matching ending \). Display math mode is for displayed equations and here is invoked by the displaymath environment. Note that any mathematical text whatever, including mathematical text consisting of just one character, is handled in math mode.

When in math mode or display math mode, \( \text{LaTeX} \) handles many aspects of your input text differently than in other text modes. For example,

\( x+y \) with \( x+y \)

in math mode the letters are in italics and the spacing around the plus sign is different.

There are three ways to make inline formulas, to put \( \text{LaTeX} \) in math mode.

\[
\text{mathematical material}
\]

\[
\text{mathematical material}
\]

\[
\text{begin} \{ \text{math} \} \text{mathematical material} \text{end} \{ \text{math} \}
\]

The first form is preferred and the second is quite common, but the third form is rarely used. You can sometimes use one and sometimes another, as in \( x \) and \( y \). You can use these in paragraph mode or in LR mode (see Chapter 17 [Modes], page 177).

To make displayed formulas, put \( \text{LaTeX} \) into display math mode with either:

\[
\text{begin} \{ \text{displaymath} \}
\text{mathematical material}
\text{end} \{ \text{displaymath} \}
\]

or

\[
\text{begin} \{ \text{equation} \}
\text{mathematical material}
\text{end} \{ \text{equation} \}
\]

(see Section 8.5 [displaymath], page 56, see Section 8.9 [equation], page 59). The only difference is that with the equation environment, \( \text{LaTeX} \) puts a formula number alongside the formula. The construct \( [ \text{math} ] \) is equivalent to \text{begin} \{ \text{displaymath} \} \text{math} \text{end} \{ \text{displaymath} \}. These environments can only be used in paragraph mode (see Chapter 17 [ Modes ], page 177).

The American Mathematical Society has made freely available a set of packages that greatly expand your options for writing mathematics, amsmath and amssymb (also be aware of the mathtools package that is an extension to, and loads, amsmath). New documents that will have mathematical text should use these packages. Descriptions of these packages is outside the scope of this document; see their documentation on CTAN.
16.1 Subscripts & superscripts

Synopsis (in math mode or display math mode), one of:

\[ \text{base}^\text{exp} \]
\[ \text{base}^{\{\text{exp}\}} \]

or, one of:

\[ \text{base}_{\text{exp}} \]
\[ \text{base}_{\{\text{exp}\}} \]

Make \text{exp} appear as a superscript of \text{base} (with the caret character, \^) or a subscript (with underscore, \_).

In this example the 0’s and 1’s are subscripts while the 2’s are superscripts.

\[
(x_0+x_1)^2 \leq (x_0)^2+(x_1)^2
\]

To have the subscript or superscript contain more than one character, surround the expression with curly braces, as in \(e^{-2x}\). This example’s fourth line shows curly braces used to group an expression for the exponent.

\[
\begin{align*}
(3^3)^3 &= 27^3 = 19,683 \\
\text{qquad} \\
3^{(3^3)} &= 3^{27} = 7,625,597,484,987
\end{align*}
\]

\(\text{\LaTeX}\) knows how to handle a superscript on a superscript, or a subscript on a subscript, or supers on subs, or subs on supers. So, expressions such as \(e^{x^2}\) and \(x_{i_0}\) give correct output. Note the use in those expressions of curly braces to give the base a determined exp. If you enter \(3^3^3\), this interpreted as \(3^{3^3}\) and then you get \(\text{\LaTeX}\) error ‘Double superscript’.

\(\text{\LaTeX}\) does the right thing when something has both a subscript and a superscript. In this example the integral has both. They come out in the correct place without any author intervention.

\[
\begin{align*}
\int_{x=a}^b f'(x)\,dx &= f(b)-f(a)
\end{align*}
\]

Note the curly braces around \(x=a\) to make the entire expression a subscript.

To put a superscript or subscript before a symbol, use a construct like \(_t K^2\). The empty curly braces {} give the subscript something to attach to and keeps it from accidentally attaching to a prior symbols.

Using the subscript or superscript character outside of math mode or display math mode, as in the expression \(x^2\), will get you the \(\text{\LaTeX}\) error ‘Missing $ inserted’.

A common reason to want subscripts outside of a mathematics mode is to typeset chemical formulas. There are packages for that, such as \texttt{mhchem}; see CTAN.

16.2 Math symbols

\(\text{\LaTeX}\) provides almost any mathematical or technical symbol that anyone uses. For example, if you include \(\texttt{\$\pi\$}\) in your source, you will get the pi symbol \(\pi\). See the “Comprehensive \(\text{\LaTeX}\) Symbol List” package at \url{https://ctan.org/pkg/comprehensive}. 
Here is a list of commonly-used symbols. It is by no means exhaustive. Each symbol is described with a short phrase, and its symbol class, which determines the spacing around it, is given in parenthesis. Unless said otherwise, the commands for these symbols can be used only in math mode. To redefine a command so that it can be used whatever the current mode, see Section 17.1 \texttt{\textbackslash ensuremath}, page 177.

\|$\parallel$\ Parallel (relation). Synonym: $\texttt{\parallel}$.
\texttt{\textbackslash aleph}$\aleph$\ Aleph, transfinite cardinal (ordinary).
\texttt{\textbackslash alpha}$\alpha$\ Lowercase Greek letter alpha (ordinary).
\texttt{\textbackslash amalg}$\amalg$\ Disjoint union (binary)
\texttt{\textbackslash angle}$\angle$\ Geometric angle (ordinary). Similar: less-than sign $<$ and angle bracket $\langle$.
\texttt{\textbackslash approx}$\approx$\ Almost equal to (relation).
\texttt{\textbackslash ast}$*\ Asterisk operator, convolution, six-pointed (binary). Synonym: $\ast$, which is often a superscript or subscript, as in the Kleene star. Similar: $\star$, which is five-pointed, and is sometimes used as a general binary operation, and sometimes reserved for cross-correlation.
\texttt{\textbackslash asym}$\asymp$\ Asymptotically equivalent (relation).
\texttt{\textbackslash backslash}$\backslash$\ Backslash (ordinary). Similar: set minus $\setminus$, and $\texttt{\textbackslash textbackslash}$ for backslash outside of math mode.
\texttt{\textbackslash beta}$\beta$\ Lowercase Greek letter beta (ordinary).
\texttt{\textbackslash bigcap}$\bigcap$\ Variable-sized, or n-ary, intersection (operator). Similar: binary intersection $\cap$.
\texttt{\textbackslash bigcirc}$\bigcirc$\ Circle, larger (binary). Similar: function composition $\circ$.
\texttt{\textbackslash bigcup}$\bigcup$\ Variable-sized, or n-ary, union (operator). Similar: binary union $\cup$.
\texttt{\textbackslash bigodot}$\bigodot$\ Circle, larger (binary). Similar: function composition $\circ$.
\texttt{\textbackslash bigoplus}$\bigoplus$\ Variable-sized, or n-ary, circled plus operator (operator).
\texttt{\textbackslash bigotimes}$\bigotimes$\ Variable-sized, or n-ary, circled times operator (operator).
\texttt{\textbackslash bigtriangledown}$\bigtriangledown$\ Variable-sized, or n-ary, open triangle pointing down (binary). Synonym: $\varbigtriangledown$.
\texttt{\textbackslash bigtriangleup}$\bigtriangleup$\ Variable-sized, or n-ary, open triangle pointing up (binary). Synonym: $\varbigtriangleup$.
\texttt{\textbackslash bigsqcup}$\bigsqcup$\ Variable-sized, or n-ary, square union (operator).
\biguplus
| Variable-sized, or n-ary, union operator with a plus (operator). (Note that the name has only one p.)
\bigvee
| Variable-sized, or n-ary, logical-or (operator).
\bigwedge
| Variable-sized, or n-ary, logical-and (operator).
\bot
| Up tack, bottom, least element of a partially ordered set, or a contradiction (ordinary). See also \top.
\bowtie
| Natural join of two relations (relation).
\Box
| Modal operator for necessity; square open box (ordinary). Not available in plain \TeX. In \LaTeX you need to load the amssymb package.
\bullet
| Bullet (binary). Similar: multiplication dot \cdot.
\cap
| Intersection of two sets (binary). Similar: variable-sized operator \bigcap.
\cdot
| Multiplication (binary). Similar: Bullet dot \bullet.
\chi
| Lowercase Greek chi (ordinary).
\circ
\clubsuit
| Club card suit (ordinary).
\complement
| Set complement, used as a superscript as in \(S^\complement\) (ordinary). Not available in plain \TeX. In \LaTeX you need to load the amssymb package. Also used: \(S^{\mathsf{c}}\) or \(\bar{S}\).
\cong
| Congruent (relation).
\coprod
| Cop product (operator).
\cup
| Union of two sets (binary). Similar: variable-sized operator \bigcup.
\dagger
| Dagger relation (binary).
\dashv
| Dash with vertical, reversed turnstile (relation). Similar: turnstile \vdash.
\ddagger
| Double dagger relation (binary).
\Delta
| Greek uppercase delta, used for increment (ordinary).
\delta
| Greek lowercase delta (ordinary).
\Diamond
| Large diamond operator (ordinary). Not available in plain \TeX. In \LaTeX you need to load the amssymb package.
\diamond
| Diamond operator (binary). Similar: large diamond \Diamond, circle bullet \bullet.
\diamondsuit
| Diamond card suit (ordinary).
\div \quad Division sign (binary).
\doteq \quad \doteq \quad Approaches the limit (relation). Similar: geometrically equal to \Doteq.
\downarrow \quad \Downarrow \quad Down arrow, converges (relation). Similar: \Downarrow double line down arrow.
\Downarrow \quad \Downarrow \quad \Downarrow \quad Double line down arrow (relation). Similar: \downarrow single line down arrow.
\ell \quad \ell \quad Lowercase cursive letter l (ordinary).
\emptyset \quad \emptyset \quad Empty set symbol (ordinary). The variant form is \varnothing.
\epsilon \quad \epsilon \quad Lowercase lunate epsilon (ordinary). Similar to Greek text letter. More widely used in mathematics is the script small letter epsilon \varepsilon. Related: the set membership relation \in \in.
\equiv \quad \equiv \quad Equivalence (relation).
\eta \quad \eta \quad Lowercase Greek letter (ordinary).
\exists \quad \exists \quad Existential quantifier (ordinary).
\flat \quad \flat \quad Musical flat (ordinary).
\forall \quad \forall \quad Universal quantifier (ordinary).
\frown \quad \frown \quad Downward curving arc (ordinary).
\Gamma \quad \Gamma \quad Uppercase Greek letter (ordinary).
\gamma \quad \gamma \quad Lowercase Greek letter (ordinary).
\ge \quad \ge \quad Greater than or equal to (relation). This is a synonym for \geq.
\geq \quad \ge \quad Greater than or equal to (relation). This is a synonym for \geq.
\gets \quad \gets \quad \gets \quad Is assigned the value (relation). Synonym: \leftarrow.
\gg \quad \gg \quad Much greater than (relation). Similar: much less than \ll.
\hbar \quad \hbar \quad Planck constant over two pi (ordinary).
\heartsuit \quad \heartsuit \quad Heart card suit (ordinary).
\hookrightarrow \quad \hookrightarrow \quad \hookrightarrow \quad Hooked right arrow (relation).
\hookleftarrow \quad \hookrightarrow \quad \hookrightarrow \quad Hooked left arrow (relation).
\iff \quad \iff \quad \iff \quad If and only if (relation). It is \Longleftrightarrow with a \thickmuskip on either side.
\Im \quad \Im \quad Imaginary part (ordinary). See: real part \Re.
\textbf{\imath} Dotless i; used when you are putting an accent on an i (see Section 16.4 [Math accents], page 169).

\textbf{\in} Set element (relation). See also: lowercase lunate epsilon \textbf{\epsilon} and small letter script epsilon \textbf{\varepsilon}.

\textbf{\infty} Infinity (ordinary).

\textbf{\int} Integral (operator).

\textbf{\iota} Lowercase Greek letter (ordinary).

\textbf{\Join} Condensed bowtie symbol (relation). Not available in Plain TeX.

\textbf{\jmath} Dotless j; used when you are putting an accent on a j (see Section 16.4 [Math accents], page 169).

\textbf{\kappa} Lowercase Greek letter (ordinary).

\textbf{\Lambda} Uppercase Greek letter (ordinary).

\textbf{\lambda} Lowercase Greek letter (ordinary).

\textbf{\land} Logical and (binary). Synonym: \textbf{\wedge}. See also logical or \textbf{\lor}.

\textbf{\langle} Left angle, or sequence, bracket (opening). Similar: less-than \textbf{\textless}. Matches \textbf{\rangle}.

\textbf{\lbrace} Left curly brace (opening). Synonym: \textbf{\{}. Matches \textbf{\rbrace}.

\textbf{\lbrack} Left square bracket (opening). Synonym: \textbf{[}. Matches \textbf{\rbrack}.

\textbf{\lceil} Left ceiling bracket, like a square bracket but with the bottom shaved off (opening). Matches \textbf{\rceil}.

\textbf{\leq} Less than or equal to (relation). This is a synonym for \textbf{\leq}. Matches \textbf{\leq}.

\textbf{\leadsto} Squiggly right arrow (relation). To get this symbol outside of math mode you can put \textbf{newcommand+}{\textbf{\Leadsto}} in the preamble and then use \textbf{\Leadsto} instead.

\textbf{\Leftarrow} Is implied by, double-line left arrow (relation). Similar: single-line left arrow \textbf{\leftarrow}.

\textbf{\Leftarrow} Single-line left arrow (relation). Synonym: \textbf{\gets}. Similar: double-line left arrow \textbf{\Leftarrow}.

\textbf{\leftharpoonup} Single-line left harpoon, barb under bar (relation).

\textbf{\leftharpoonup} Single-line left harpoon, barb over bar (relation).

\textbf{\Leftrightarrow} Bi-implication; double-line double-headed arrow (relation). Similar: single-line double headed arrow \textbf{\leftrightharpoons}.
\leftarrow – Single-line double-headed arrow (relation). Similar: double-line double-headed arrow \Leftarrow.

\leq \leq Less than or equal to (relation). This is a synonym for \le.

\lfloor \lfloor Left floor bracket (opening). Matches: \floor.

\lhd \lhd Arrowhead, that is, triangle, pointing left (binary). For the normal subgroup symbol you should load amssymb and use \vartriangleleft (which is a relation and so gives better spacing).

\ll \ll Much less than (relation). Similar: much greater than \gg.

\lnot \lnot Logical negation (ordinary). Synonym: \neg.

\longleftarrow \longleftrightarrow ←→ Long single-line double-headed arrow (relation). Similar: long double-line double-headed arrow \Longleftarrow.

\longleftrightarrow \longleftrightarrow ←→ Long single-line double-headed arrow (relation). Similar: long double-line double-headed arrow \Longleftrightarrow.

\longmapsto \longmapsto ↦→ Long single-line left arrow starting with vertical bar (relation). Similar: shorter version \mapsto.

\longrightarrow \longrightarrow →→ Long single-line right arrow (relation). Similar: long double-line right arrow \Longrightarrow.

\lor \lor Logical or (binary). Synonym: \vee. See also logical and \land.

\mapsto \mapsto ↦→ Single-line left arrow starting with vertical bar (relation). Similar: longer version \longmapsto.

\models \models ⊧ Entails, or satisfies; double turnstile, short double dash (relation). Similar: long double dash \vDash.

\models \models ⊧ Entails, or satisfies; double turnstile, short double dash (relation). Similar: long double dash \vDash.

\mp \mp ⊕ Minus or plus (relation).

\mu \mu µ Lowercase Greek letter (ordinary).

\nabla \nabla ∇ Hamilton’s del, or differential, operator (ordinary).

\natural \natural † Musical natural notation (ordinary).
\ne \neq Not equal (relation). Synonym: \neq.

\nearrow \nearrow North-east arrow (relation).

\neg \neg Logical negation (ordinary). Synonym: \lnot. Sometimes instead used for negation: \sim.

\neq \neq Not equal (relation). Synonym: \ne.

\ni \ni Reflected membership epsilon; has the member (relation). Synonym: \owns. Similar: is a member of \in.

\not / \not Long solidus, or slash, used to overstrike a following operator (relation). Many negated operators are available that don’t require \not, particularly with the amssymb package. For example, \notin is typographically preferable to \not\in.

\notin \notin Not an element of (relation). Similar: not subset of \nsubseteqq.

\nu \nu Lowercase Greek letter (ordinary).

\nwarrow \nwarrow North-west arrow (relation).

\odot \odot Dot inside a circle (binary). Similar: variable-sized operator \bigodot.

\oint \oint Contour integral, integral with circle in the middle (operator).

\Omega \Omega Uppercase Greek letter (ordinary).

\omega \omega Lowercase Greek letter (ordinary). The variant form is \varphi \varphi.

\ominus \ominus Minus sign, or dash, inside a circle (binary).

\oplus \oplus Plus sign inside a circle (binary). Similar: variable-sized operator \bigoplus.

\oslash \oslash Solidus, or slash, inside a circle (binary).

\otimes \otimes Times sign, or cross, inside a circle (binary). Similar: variable-sized operator \bigotimes.

\owns \owns Reflected membership epsilon; has the member (relation). Synonym: \ni. Similar: is a member of \in.

\parallel \parallel Parallel (relation). Synonym: \|.

\partial \partial Partial differential (ordinary).

\perp \perp Perpendicular (relation). Similar: \bot uses the same glyph but the spacing is different because it is in the class ordinary.

\Phi \Phi Uppercase Greek letter (ordinary).

\phi \phi Lowercase Greek letter (ordinary). The variant form is \varphi \varphi.

\Pi \Pi Uppercase Greek letter (ordinary).

\pi \pi Lowercase Greek letter (ordinary). The variant form is \varpi \varpi.

\pm \pm Plus or minus (binary).
\prec \prec Precedes (relation). Similar: less than \textless.
\preceq \preceq Precedes or equals (relation). Similar: less than or equals \textless=.
\prime \prime Prime, or minute in a time expression (ordinary). Typically used as a superscript: $f^\prime$; $f''$ and $f'''$ produce the same result. An advantage of the second is that $f''''$ produces the desired symbol, that is, the same result as $f^\prime\prime\prime\prime$, but uses rather less typing. You can only use \prime in math mode. Using the right single quote ’ in text mode produces a different character (apostrophe).
\prod \prod Product (operator).
\propto \propto Is proportional to (relation)
\Psi \Psi uppercase Greek letter (ordinary).
\psi \psi Lowercase Greek letter (ordinary).
\rangle \rangle Right angle, or sequence, bracket (closing). Similar: greater than \textgreater. Matches: \langle.
\rbrace \lbrace Right curly brace (closing). Synonym: \}. Matches \lbrack.
\rbrack \rbrack Right square bracket (closing). Synonym: \]. Matches \lceil.
\rceil \rceil Right ceiling bracket (closing). Matches \lfloor.
\Re \Re Real part, real numbers, cursive capital R (ordinary). Related: double-line, or blackboard bold, R \mathbb{R}; to access this, load the amsfonts package.
\restriction \restriction Restriction of a function (relation). Synonym: \upharpoonright. Not available in plain \TeX. In \LaTeX you need to load the amssymb package.
\revemptyset \revemptyset Reversed empty set symbol (ordinary). Related: \emptyset. Not available in plain \TeX. In \LaTeX you need to load the stix package.
\rfloor \rfloor Right floor bracket, a right square bracket with the top cut off (closing). Matches \lbrack.
\rhd \rhd Arrowhead, that is, triangle, pointing right (binary). For the normal subgroup symbol you should instead load amssymb and use \vartriangleright (which is a relation and so gives better spacing).
\rho \rho Lowercase Greek letter (ordinary). The variant form is \varkappa ρ.
\Rightarrow \Rightarrow \Rightarrow Implies, right-pointing double line arrow (relation). Similar: right single-line arrow \rightarrow.
\rightarrow \rightarrow Right-pointing single line arrow (relation). Synonym: \to. Similar: right double line arrow \Rightarrow.
\rightharpoondown \rightharpoondown Right-pointing harpoon with barb below the line (relation).
\rightharpoonup \rightarrow Right-pointing harpoon with barb above the line (relation).
\rightleftharpoons \equiv Right harpoon up above left harpoon down (relation).
\searrow \downarrow Arrow pointing southeast (relation).
\setminus \setminus Set difference, reverse solidus or reverse slash, like \setminus (binary). Similar: backslash \textbackslash and also \textbackslash outside of math mode.
\sharp \# Musical sharp (ordinary).
\Sigma \Sigma uppercase Greek letter (ordinary).
\sigma \sigma Lowercase Greek letter (ordinary). The variant form is \varsigma \varsigma.
\sim \sim Similar, in a relation (relation).
\simeq \simeq Similar or equal to, in a relation (relation).
\smallint \int Integral sign that does not change to a larger size in a display (operator).
\smile \smile Upward curving arc, smile (ordinary).
\spadesuit \spadesuit Spade card suit (ordinary).
\sqcap \sqcap Square intersection symbol (binary). Similar: intersection \cap.
\sqcup \sqcup Square union symbol (binary). Similar: union \cup. Related: variable-sized operator \bigsqcup.
\sqsubset \sqsubset Square subset symbol (relation). Similar: subset \subset. Not available in plain \TeX. In \LaTeX you need to load the \texttt{amssymb} package.
\sqsubseteq \sqsubseteq Square subset or equal symbol (binary). Similar: subset or equal to \subseteq.
\sqsupset \sqsupset Square superset symbol (relation). Similar: superset \supset. Not available in plain \TeX. In \LaTeX you need to load the \texttt{amssymb} package.
\sqsupseteq \sqsupseteq Square superset or equal symbol (binary). Similar: superset or equal \supseteq.
\star \star Five-pointed star, sometimes used as a general binary operation but sometimes reserved for cross-correlation (binary). Similar: the synonyms asterisk \ast and \texttt{ast}, which are six-pointed, and more often appear as a superscript or subscript, as with the Kleene star.
\subset \subset Subset (occasionally, is implied by) (relation).
\subseteq \text{ Subset or equal to (relation).}
\succ \text{ Comes after, succeeds (relation). Similar: less than >.}
\succeq \text{ Succeeds or is equal to (relation). Similar: less than or equal to \leq.}
\sum \text{ Summation (operator). Similar: Greek capital sigma \Sigma.}
\supset \text{ Superset (relation).}
\supseteq \text{ Superset or equal to (relation).}
\surd \text{ Radical symbol (ordinary). The \LaTeX command \sqrt{…} typesets the square root of the argument, with a bar that extends to cover the argument.}
\swarrow \text{ Southwest-pointing arrow (relation).}
\tau \text{ Lowercase Greek letter (ordinary).}
\theta \text{ Lowercase Greek letter (ordinary). The variant form is \vartheta \vartheta.}
\times \text{ Primary school multiplication sign (binary). See also \cdot.}
\to \text{ Right-pointing single line arrow (relation). Synonym: \rightarrow.}
\top \text{ Top, greatest element of a partially ordered set (ordinary). See also \bot.}
\triangle \text{ Triangle (ordinary).}
\triangleleft \text{ Not-filled triangle pointing left (binary). Similar: \lhd. For the normal subgroup symbol you should load \texttt{amssymb} and use \texttt{\vartriangleleft} (which is a relation and so gives better spacing).}
\triangleright \text{ Not-filled triangle pointing right (binary). For the normal subgroup symbol you should instead load \texttt{amssymb} and use \texttt{\vartriangleright} (which is a relation and so gives better spacing).}
\unlhd \text{ Left-pointing not-filled underlined arrowhead, that is, triangle, with a line under (binary). For the normal subgroup symbol load \texttt{amssymb} and use \texttt{\vartrianglelefteq} (which is a relation and so gives better spacing).}
\unrhd \text{ Right-pointing not-filled underlined arrowhead, that is, triangle, with a line under (binary). For the normal subgroup symbol load \texttt{amssymb} and use \texttt{\vartrianglerighteq} (which is a relation and so gives better spacing).}
\Uparrow \text{ Double-line upward-pointing arrow (relation). Similar: single-line up-pointing arrow \uparrow.}
\uparrow \text{ Single-line upward-pointing arrow, diverges (relation). Similar: double-line up-pointing arrow \Uparrow.}
\Updownarrow \text{ Double-line upward-and-downward-pointing arrow (relation). Similar: single-line upward-and-downward-pointing arrow \updownarrow.}
\updownarrow


\upharpoonright

\upharpoonright Up harpoon, with barb on right side (relation). Synonym: \restriction. Not available in plain \TeX. In \LaTeX you need to load the \texttt{amssymb} package.

\uplus

\uplus Multiset union, a union symbol with a plus symbol in the middle (binary). Similar: union \cup. Related: variable-sized operator \texttt{biguplus}.

\Upsilon

\Upsilon Upsilon uppercase Greek letter (ordinary).

\upsilon

\upsilon Lowercase Greek letter (ordinary).

\varepsilon

\varepsilon Small letter script epsilon (ordinary). This is more widely used in mathematics than the non-variant lunate epsilon form \texttt{epsilon \varepsilon}. Related: set membership \texttt{in}.

\van nothing

\van nothing Empty set symbol. Similar: \texttt{emptyset}. Related: \texttt{reemptyset}. Not available in plain \TeX. In \LaTeX you need to load the \texttt{amssymb} package.

\varphi

\varphi \varphi Variant on the lowercase Greek letter (ordinary). The non-variant form is \texttt{phi \phi}.

\varpi

\varpi Variant on the lowercase Greek letter (ordinary). The non-variant form is \texttt{pi \pi}.

\varrho

\varrho Variant on the lowercase Greek letter (ordinary). The non-variant form is \texttt{rho \rho}.

\varsigma

\varsigma \varsigma Variant on the lowercase Greek letter (ordinary). The non-variant form is \texttt{sigma \sigma}.

\vartheta

\vartheta \vartheta Variant on the lowercase Greek letter (ordinary). The non-variant form is \texttt{theta \theta}.

\vdash

\vdash \vdash Provable; turnstile, vertical and a dash (relation). Similar: turnstile rotated a half-circle \texttt{dashv}.

\vee

\vee \vee Logical or; a downwards v shape (binary). Related: logical and \texttt{wedge}. Similar: variable-sized operator \texttt{bigvee}.

\Vert

\Vert \Vert Vertical double bar (ordinary). See Section 16.2.5 [Delimiters], page 162, for how to use the \texttt{mathtools} package to create flexibly-sized norm symbols.

\vert

\vert \vert Single line vertical bar (ordinary). For “such that”, as in the definition of a set, use \texttt{mid} because it is a relation. See Section 16.2.5 [Delimiters], page 162, for how to use the \texttt{mathtools} package to create flexibly-sized absolute-value symbols.

\wedge

\wedge \wedge Logical and (binary). Synonym: \texttt{land}. See also logical or \texttt{vee}. Similar: variable-sized operator \texttt{bigwedge}.
\wp \qquad \wp \quad \text{Weierstrass p (ordinary).}

\\wr \qquad \wr \quad \text{Wreath product (binary).}

\Xi \quad \Xi \quad \text{Uppercase Greek letter (ordinary).}

\xi \quad \xi \quad \text{Lowercase Greek letter (ordinary).}

\zeta \quad \zeta \quad \text{Lowercase Greek letter (ordinary).}

The following symbols are most often used in plain text but LATEX provides versions to use in mathematical text.

\mathdollar
\quad \text{Dollar sign in math mode: \$}.

\mathparagraph
\quad \text{Paragraph sign (pilcrow) in math mode, ¶}.

\mathsection
\quad \text{Section sign in math mode: §}.

\mathsterling
\quad \text{Sterling sign in math mode: £}.

\mathunderscore
\quad \text{Underscore in math mode: _}.

16.2.1 Arrows

These are the arrows that come with standard LATEX. The \texttt{latexsym} and \texttt{amsfonts} packages contain many more.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>⇓</td>
<td>\Downarrow</td>
</tr>
<tr>
<td>↓</td>
<td>\downarrow</td>
</tr>
<tr>
<td>←</td>
<td>\hookleftarrow</td>
</tr>
<tr>
<td>→</td>
<td>\hookrightarrow</td>
</tr>
<tr>
<td>←</td>
<td>\leftarrow</td>
</tr>
<tr>
<td>⇐</td>
<td>\Leftarrow</td>
</tr>
<tr>
<td>⇔</td>
<td>\Leftrightarrow</td>
</tr>
<tr>
<td>←−</td>
<td>\longleftarrow</td>
</tr>
<tr>
<td>⇐⇒</td>
<td>\Longleftarrow</td>
</tr>
<tr>
<td>←−−−</td>
<td>\Longleftarrow</td>
</tr>
<tr>
<td>←→−</td>
<td>\longleftarrow</td>
</tr>
<tr>
<td>←⇒−</td>
<td>\Longleftarrow</td>
</tr>
<tr>
<td>→→−</td>
<td>\longrightarrow</td>
</tr>
<tr>
<td>→−</td>
<td>\longrightarrow</td>
</tr>
<tr>
<td>↑</td>
<td>\nearrow</td>
</tr>
<tr>
<td>↓</td>
<td>\nwarrow</td>
</tr>
<tr>
<td>⇒</td>
<td>\Rightarrow</td>
</tr>
<tr>
<td>→</td>
<td>\rightarrow</td>
</tr>
<tr>
<td>←</td>
<td>\Leftarrow</td>
</tr>
<tr>
<td>⇒</td>
<td>\Rightarrow</td>
</tr>
<tr>
<td>−→</td>
<td>\longrightarrow</td>
</tr>
<tr>
<td>−⇒</td>
<td>\longmapsto</td>
</tr>
</tbody>
</table>


An example of the difference between \text{to} and \text{mapsto} is: $(f \colon D \to C)$ given by $(n \mapsto n^2)$.

For commutative diagrams there are a number of packages, including \texttt{tikz-cd} and \texttt{amscd}.

\section*{16.2.2 \texttt{\boldmath} \& \texttt{\unboldmath}}

Synopsis (used in paragraph mode or LR mode):

\begin{verbatim}
\boldmath $(math)$
\end{verbatim}

or

\begin{verbatim}
\unboldmath $(math)$
\end{verbatim}

Declarations to change the letters and symbols in \texttt{math} to be in a bold font, or to countermand that and bring back the regular (non-bold) default, respectively. They must be used when \texttt{not} in math mode or display math mode (see Chapter 17 [Modes], page 177). Both commands are fragile (see Section 12.11 [\texttt{protect}], page 123).

In this example each \texttt{\boldmath} command takes place inside an \texttt{\mbox},

\begin{verbatim}
we have $\mbox{\boldmath $(v)$} = 5 \cdot \mbox{\boldmath $(u)$}$
\end{verbatim}

which means \texttt{\boldmath} is only called in a text mode, here LR mode, and explains why we must switch \LaTeX into math mode to set $v$ and $u$.

If you use either command inside math mode, as with \texttt{Trouble: $(\boldmath x)$}, then you get something like ‘\LaTeX Font Warning: Command \texttt{\boldmath} invalid in math mode’ and ‘\LaTeX Font Warning: Command \texttt{\mathversion} invalid in math mode’.

\subsection*{16.2.2.1 bm: Individual bold math symbols}

Specifying \texttt{\boldmath} is the best method for typesetting a whole math expression in bold. But to typeset individual symbols within an expression in bold, the \texttt{bm} package provided by the \LaTeX Project team is better. Its usage is outside the scope of this document (see its documentation at \url{https://ctan.org/pkg/bm} or in your installation) but the spacing in the output of this small example will show that it is an improvement over \texttt{\boldmath} within an expression:

\begin{verbatim}
\usepackage{bm}  \% in preamble
...
we have $\texttt{\bm{v}} = 5 \cdot \texttt{\bm{u}}$
\end{verbatim}

\subsection*{16.2.2.2 OpenType bold math}

Unfortunately, when using the Unicode engines (\texttt{Xe\LaTeX}, \texttt{Lua\LaTeX}), neither \texttt{\boldmath} nor \texttt{bm} usually work well, because the OpenType math fonts normally used with those engines rarely come with a bold companion, and both \texttt{\boldmath} and \texttt{bm} require this. (The implementation of \texttt{bm} relies on \texttt{\boldmath}, so the requirements are the same.) If you do have a bold math font, though, then \texttt{\boldmath} and \texttt{bm} work fine.
If no such font is available, one alternative is to construct fake bold fonts with the fontspec package’s FakeBold=1 parameter (see its documentation, https://ctan.org/pkg/fontspec). This may be acceptable for drafting or informal distribution, but the results are far from a true bold font.

Another alternative to handling bold for OpenType math fonts is to use the \symbf (bold), \symbfit (bold italic), and related commands from the unicode-math package. These do not change the current font, but rather change the (Unicode) “alphabet” used, which in practice is more widely supported than a separate bold font. Many variations are possible, and so there are subtleties to getting the desired output. As usual, see the package documentation (https://ctan.org/pkg/unicode-math).

16.2.3 Blackboard bold

Synopsis:
\usepackage{amssymb} % in preamble
...
\mathbb{uppercase-letter}

Provide blackboard bold symbols, sometimes also known as doublestruck letters, used to denote number sets such as the natural numbers, the integers, etc.

Here
\(( \forall n \in \mathbb{N}, n^2 \geq 0 \)\)
the \mathbb{N} gives blackboard bold symbol N, representing the natural numbers.

If the argument contains something other than an uppercase letter, you do not get an error but you do get strange results, including unexpected characters.

There are packages that give access to symbols other than just the capital letters; look on CTAN.

16.2.4 Calligraphic

Synopsis:
\mathcal{uppercase-letters}

Use a script-like font.

In this example the graph identifier is output in a cursive font.

Let the graph be \(( \mathcal{G} \)\).

If you use something other than an uppercase letter then you do not get an error but you also do not get math calligraphic output. For instance, \mathcal{g} outputs a close curly brace symbol.

16.2.5 Delimiters

Delimiters are parentheses, braces, or other characters used to mark the start and end of subformulas. This formula has three sets of parentheses delimiting the three subformulas.

\[(z-z_0)^2 = (x-x_0)^2 + (y-y_0)^2\]

The delimiters do not need to match, so you can enter \((0,1)\).

Here are the common delimiters:
The `mathtools` package allows you to create commands for paired delimiters. For instance, if you put \texttt{\DeclarePairedDelimiter\abs\lvert\rvert} in your preamble then you get two commands for single-line vertical bars (they only work in math mode). The starred form, such as \texttt{\abs*{\frac{22}{7}}}, has the height of the vertical bars match the height of the argument. The unstarred form, such as \texttt{\abs{\frac{22}{7}}}, has the bars fixed at a default height. This form accepts an optional argument, as in \texttt{\abs[size command]{\frac{22}{7}}}, where the height of the bars is given in \texttt{size command}, such as \texttt{\Bigg}. Using instead \texttt{\lVert} and \texttt{\rVert} as the symbols will give you a norm symbol with the same behavior.

### 16.2.5.1 \left & \right

**Synopsis:**

```
\left delimeter1 \ldots \right delimeter2
```

Make matching parentheses, braces, or other delimiters. \LaTeX makes the delimiters tall enough to just cover the size of the formula that they enclose.

This makes a unit vector surrounded by parentheses tall enough to cover the entries.

```
\begin{equation}
\begin{array}{c}
1 \\
0 \\
\end{array}
\end{equation}
```

See Section 16.2.5 [Delimiters], page 162, for a list of the common delimiters.

Every \texttt{\left} must have a matching \texttt{\right}. In the above example, leaving out the \texttt{\left(} gets the error message 'Extra \texttt{\right}'. Leaving out the \texttt{\right)} gets 'You can’t use \texttt{\eqno} in math mode'.

However, \texttt{delimeter1} and \texttt{delimeter2} need not match. A common case is that you want an unmatched brace, as below. Use a period, '.', as a \texttt{null delimeter}.
\begin{equation}
f(n)=\left\{\begin{array}{ll}
1 & \text{--if } (n=0) \\
 f(n-1)+3n^2 & \text{--else}
\end{array}\right.
\end{equation}

Note that to get a curly brace as a delimiter you must prefix it with a backslash, \{ (see [Reserved characters], page 6). (The packages amsmath and mathtools allow you to get the above construct through in a cases environment.)

The \texttt{\textbackslash left} ... \texttt{\textbackslash right} pair make a group. One consequence is that the formula enclosed in the \texttt{\textbackslash left} ... \texttt{\textbackslash right} pair cannot have line breaks in the output. This includes both manual line breaks and \LaTeX-generated automatic ones. In this example, \LaTeX breaks the equation to make the formula fit the margins.

\begin{equation}
\text{Lorem ipsum dolor sit amet}
\left((a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+x+y+z)\right)
\end{equation}

But with \texttt{\textbackslash left} and \texttt{\textbackslash right}

\begin{equation}
\text{Lorem ipsum dolor sit amet}
\left(\text{Lorem ipsum dolor sit amet} \right)
\end{equation}

\LaTeX won’t break the line, causing the formula to extend into the margin.

Because \texttt{\textbackslash left} ... \texttt{\textbackslash right} make a group, all the usual grouping rules hold. Here, the value of \texttt{\textbackslash testlength} set inside the equation will be forgotten, and the output is ‘1.2pt’.

\begin{verbatim}
\newlength{\testlength} \setlength{\testlength}{1.2pt}
\begin{equation}
\left( a+b=c \setlength{\testlength}{3.4pt} \right) \the\testlength
\end{equation}
\end{verbatim}

The \texttt{\textbackslash left} ... \texttt{\textbackslash right} pair affect the horizontal spacing of the enclosed formula, in two ways. The first is that in \texttt{\sin(x) = \sin(left(x)\right)} the one after the equals sign has more space around the x. That’s because \texttt{\textbackslash left( ... \textbackslash right)} inserts an inner node while \texttt{( ... )} inserts an opening node. The second way that the pair affect the horizontal spacing is that because they form a group, the enclosed subformula will be typeset at its natural width, with no stretching or shrinking to make the line fit better.

\LaTeX scales the delimiters according to the height and depth of the enclosed formula. Here \LaTeX grows the brackets to extend the full height of the integral.

\begin{equation}
\left[ \int_{x=r_0}^{\infty} -G\frac{Mm}{r^2}\, dr \right]
\end{equation}

Manual sizing is often better. For instance, although below the rule has no depth, \LaTeX will create delimiters that extend far below the rule.

\begin{verbatim}
\begin{equation}
\left( \rule{1pt}{1cm} \right)
\end{equation}
\end{verbatim}

\LaTeX can choose delimiters that are too small, as in \texttt{\left| |x|+|y| \right| \textbackslash \textbackslash)}. It can also choose delimiters that are too large, as here.
A third awkward case is when a long displayed formula is on more than one line and you must match the sizes of the opening and closing delimiter; you can’t use \texttt{\left} on the first line and \texttt{\right} on the last because they must be paired.

To size the delimiters manually, see Section 16.2.5.2 [\texttt{\bigl} & \texttt{\bigr} etc.], page 165.

16.2.5.2 \texttt{\bigl}, \texttt{\bigr}, etc.

Synopsis, one of:

\begin{verbatim}
\bigl delimiter1 ... \bigr delimiter2
\Bigl delimiter1 ... \Bigr delimiter2
\biggl delimiter1 ... \biggr delimiter2
\Biggl delimiter1 ... \Biggr delimiter2
\end{verbatim}

(as with \texttt{\bigl[} ... \texttt{\bigr]}; strictly speaking they need not be paired, see below), or one of:

\begin{verbatim}
\bigm delimiter
\Bigm delimiter
\biggm delimiter
\Biggm delimiter
\end{verbatim}

(as with \texttt{\bigm|}), or one of:

\begin{verbatim}
\bigdelimiter
\Bigdelimiter
\biggdelimiter
\Biggdelimiter
\end{verbatim}

Produce manually-sized delimiters. For delimiters that are automatically sized see Section 16.2.5.1 [\texttt{\left} & \texttt{\right}], page 163.

This produces slightly larger outer vertical bars.

\begin{verbatim}
\big| \mid x \mid + \mid y \mid \bigr|
\end{verbatim}

The commands above are listed in order of increasing size. You can use the smallest size such as \texttt{\bigl...\bigr} in a paragraph without causing \LaTeX{} to spread the lines apart. The larger sizes are meant for displayed equations.

See Section 16.2.5 [Delimiters], page 162, for a list of the common delimiters. In the family of commands with ‘1’ or ‘r’, \texttt{delimiter1} and \texttt{delimiter2} need not match together.

The ‘l’ and ‘r’ commands produce open and close delimiters that insert no horizontal space between a preceding atom and the delimiter, while the commands without ‘l’ and ‘r’ insert some space (because each delimiter is set as an ordinary variable). Compare these two.

\begin{verbatim}
\begin{tabular}{l}
\begin{pmatrix} \sin \frac{1}{2} \end{pmatrix} & | \mid x \mid + \mid y \mid \bigr|
\end{tabular}
\end{verbatim}

The traditional typographic treatment is on the first line. On the second line the output will have some extra space between the \texttt{\sin} and the open parenthesis.
Commands without ‘l’ or ‘r’ do give correct spacing in some circumstances, as with this large vertical line
\begin{equation}
\int_{x=a}^b x^2 \, dx = \frac{1}{3} x^3 \big|_{x=a}^b
\end{equation}
(many authors would replace \frac with the \tfrac command from the amsmath package), and as with this larger slash.
\begin{equation}
\lim_{n \to \infty} \pi(n) \big/ (n/\log n) = 1
\end{equation}
Unlike the \left...\right pair (see Section 16.2.5.1 \[\left & \right\], page 163), the commands here with ‘l’ or ‘r’ do not make a group. Strictly speaking they need not be matched so you can write something like this.
\begin{equation}
\Biggl[ \pi/6 \Biggr]
\end{equation}
The commands with ‘m’ are for relations, which are in the middle of formulas, as here.
\begin{equation}
\biggl\{ a \in B \biggm| a = \sum_{0 \leq i < n} 3i^2 + 4 \biggr\}
\end{equation}

16.2.6 Dots, horizontal or vertical
Ellipses are the three dots (usually three) indicating that a pattern continues.
\begin{array}{cccc}
a_{0,0} & a_{0,1} & a_{0,2} & \ldots \\
a_{1,0} & \ddots \\
\vdots
\end{array}
\begin{array}{cc}
\cdot \cdot \cdot \\
\mathellipsis
\end{array}
\begin{array}{cc}
\ldots
\end{array}
\begin{array}{cc}
\cdots
\end{array}
\begin{array}{cc}
\ddots
\end{array}
\begin{array}{cc}
\vdots
\end{array}
\Bbb{\La}{\TeX} provides these.
\cdots Horizontal ellipsis with the dots raised to the center of the line, as in \cdots. Used as:\( a_0, a_1, \cdots a_{n-1} \).
\ddots Diagonal ellipsis, \ddots. See the above array example for a usage.
\ldots Ellipsis on the baseline, \ldots. Used as:\( x_0, \ldots x_{n-1} \). Another example is the above array example. Synonyms are \mathellipsis and \ldots. A synonym from the amsmath package is \ldots.
You can also use this command outside of mathematical text, as in The gears, brakes, \ldots{} are all broken.
\vdots Vertical ellipsis, \vdots. See the above array example for a usage.
The `amsmath` package has the command \dots to semantically mark up ellipses. This example produces two different-looking outputs for the first two uses of the \dots command.

```latex
\usepackage{amsmath} % in preamble
...
Suppose that \(( p_0, p_1, \ldots, p_{n-1} \) \) lists all of the primes. Observe that \(( p_0 \cdot p_1 \ldots \cdot p_{n-1} +1 \) \) is not a multiple of any \( p_i \).
Conclusion: there are infinitely many primes \(( p_0, p_1, \ldots \) \).
```

In the first line LaTeX looks to the comma following \dots to determine that it should output an ellipsis on the baseline. The second line has a \( \cdot \) following \dots so \LaTeX{} outputs an ellipsis that is on the math axis, vertically centered. However, the third usage has no follow-on character so you have to tell \LaTeX{} what to do. You can use one of the commands: \dotsc if you need the ellipsis appropriate for a comma following, \dotsb if you need the ellipses that fits when the dots are followed by a binary operator or relation symbol, \dotsi for dots with integrals, or \dotso for others.

The \dots command from `amsmath` differs from the \LaTeX{} kernel’s \dots command in another way: it outputs a thin space after the ellipsis. Furthermore, the `unicode-math` package automatically loads `amsmath`, so `amsmath`’s \dots may be active even when you did not explicitly load it, thus changing the output from \dots in both text and math mode.

Yet more about the ellipsis commands: when running under Unicode engines (\texttt{lualatex}, \texttt{xelatex}), \LaTeX{} will use the Unicode ellipsis character (\texttt{U+2026}) in the font if it’s available; under traditional \TeX{} engines (\texttt{pdflatex}, \texttt{latex}), it will typeset three spaced periods. Generally, the Unicode single-character ellipsis has almost no space between the three periods, while the spacing of the non-Unicode ellipsis is looser, more in accordance with traditional typography.

#### 16.2.7 Greek letters

The upper case versions of these Greek letters are only shown when they differ from Roman upper case letters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Command</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>\alpha</td>
<td>Alpha</td>
</tr>
<tr>
<td>β</td>
<td>\beta</td>
<td>Beta</td>
</tr>
<tr>
<td>γ, Ζ</td>
<td>\gamma, \Gamma</td>
<td>Gamma</td>
</tr>
<tr>
<td>δ, Δ</td>
<td>\delta, \Delta</td>
<td>Delta</td>
</tr>
<tr>
<td>ε, ϵ</td>
<td>\varepsilon, \epsilon</td>
<td>Epsilon</td>
</tr>
<tr>
<td>ζ</td>
<td>\zeta</td>
<td>Zeta</td>
</tr>
<tr>
<td>η</td>
<td>\eta</td>
<td>Eta</td>
</tr>
<tr>
<td>θ, ϑ</td>
<td>\theta, \vartheta</td>
<td>Theta</td>
</tr>
<tr>
<td>ι</td>
<td>\iota</td>
<td>Iota</td>
</tr>
<tr>
<td>κ</td>
<td>\kappa</td>
<td>Kappa</td>
</tr>
<tr>
<td>λ, Λ</td>
<td>\lambda, \Lambda</td>
<td>Lambda</td>
</tr>
<tr>
<td>μ</td>
<td>\mu</td>
<td>Mu</td>
</tr>
<tr>
<td>ν</td>
<td>\nu</td>
<td>Nu</td>
</tr>
<tr>
<td>ξ, Ξ</td>
<td>\xi, \Xi</td>
<td>Xi</td>
</tr>
<tr>
<td>π, Π</td>
<td>\pi, \Pi</td>
<td>Pi</td>
</tr>
<tr>
<td>ρ, ρ</td>
<td>\rho, \varrho</td>
<td>Rho</td>
</tr>
<tr>
<td>Greek Letter</td>
<td>LaTeX Command</td>
<td>Roman Name</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>$\sigma$, $\Sigma$</td>
<td>\sigma, \Sigma</td>
<td>Sigma</td>
</tr>
<tr>
<td>$\tau$</td>
<td>\tau</td>
<td>Tau</td>
</tr>
<tr>
<td>$\phi$, $\varphi$, $\Phi$</td>
<td>\phi, \varphi, \Phi</td>
<td>Phi</td>
</tr>
<tr>
<td>$\chi$</td>
<td>\chi</td>
<td>chi</td>
</tr>
<tr>
<td>$\psi$, $\Psi$</td>
<td>\psi, \Psi</td>
<td>Psi</td>
</tr>
<tr>
<td>$\omega$, $\Omega$</td>
<td>\omega, \Omega</td>
<td>Omega</td>
</tr>
</tbody>
</table>

For omicron, if you are using \LaTeX's default Computer Modern font then enter omicron just as 'o' or 'O'. If you like having the name or if your font shows a difference then you can use something like \newcommand\omicron{o}. The package unicode-math has \upomicron for upright omicron and \mitomicron for math italic.

While the set membership relation symbol $\in$ generated by \in is related to epsilon, it is never used for a variable.

### 16.3 Math functions

These commands produce roman function names in math mode with proper spacing.

- $\arccos$ Inverse cosine arccos
- $\arcsin$ Inverse sine arcsin
- $\arctan$ Inverse tangent arctan
- $\arg$ Angle between the real axis and a point in the complex plane arg
- $\bmod$ Binary modulo operator, used as in $(5\bmod 3=2)$ $5 \text{ mod } 3 = 2$
- $\cos$ Cosine cos
- $\cosh$ Hyperbolic cosine cosh
- $\cot$ Cotangent cot
- $\coth$ Hyperbolic cotangent coth
- $\csc$ Cosecant csc
- $\deg$ Degrees deg
- $\det$ Determinant det
- $\dim$ Dimension dim
- $\exp$ Exponential exp
- $\gcd$ Greatest common divisor gcd
- $\hom$ Homomorphism hom
- $\inf$ Infimum inf
- $\ker$ Kernel ker
- $\lg$ Base 2 logarithm lg
- $\lim$ Limit lim
- $\liminf$ Limit inferior lim inf
\limsup  \quad \text{Limit superior lim sup}
\ln  \quad \text{Natural logarithm ln}
\log  \quad \text{Logarithm log}
\max  \quad \text{Maximum max}
\min  \quad \text{Minimum min}
\pmod  \quad \text{Parenthesized modulus, as used in } 5 \equiv 2 \pmod{3} \quad (\text{mod } 3)
\Pr  \quad \text{Probability Pr}
\sec  \quad \text{Secant sec}
\sin  \quad \text{Sine sin}
\sinh  \quad \text{Hyperbolic sine sinh}
\sup  \quad \text{Supremum sup}
\tan  \quad \text{Tangent tan}
\tanh  \quad \text{Hyperbolic tangent tanh}

The amsmath package adds improvements on some of these, and also allows you to define your own. The full documentation is on CTAN, but briefly, you can define an identity operator with \DeclareMathOperator{\identity}{id} that is like the ones above but prints as ‘id’. The starred form \DeclareMathOperator*{\op}{op} sets any superscript or subscript to be above and below, as is traditional with \lim, \sup, or \max.

16.4 Math accents

\LaTeX provides a variety of commands for producing accented letters in math. These are different from accents in normal text (see Section 23.5 [Accents], page 224).
\acute  \quad \text{Math acute accent } \acute{x}
\bar  \quad \text{Math bar-over accent } \bar{x}
\breve  \quad \text{Math breve accent } \breve{x}
\check  \quad \text{Math háček (check) accent } \check{x}
\ddot  \quad \text{Math dieresis accent } \ddot{x}
\dot  \quad \text{Math dot accent } \dot{x}
\grave  \quad \text{Math grave accent } \grave{x}
\hat  \quad \text{Math hat (circumflex) accent } \hat{x}
\mathring  \quad \text{Math ring accent } \mathring{x}
\tilde  \quad \text{Math tilde accent } \tilde{x}
\vec  \quad \text{Math vector symbol } \vec{x}
\widehat  \quad \text{Math wide hat accent } x + y
\widetilde

Math wide tilde accent $x \pm y$

When you are putting an accent on an i or a j, the tradition is to use one without a dot, \imath or jmath (see Section 16.2 [Math symbols], page 149).

16.5 Over- or under math

\LaTeX \ provides commands for putting lines, braces, and arrows over or under math material.

\underline{math}

Underline math. For example: \underline{x+y}. The result looks like: $x + y$. The line is always completely below the text, taking account of descenders, so in $\underline{y}$ the line is lower than in $\underline{x}$. As of approximately 2019, this command and others in this section are robust; before that, they were fragile (see Section 12.11 [\protect], page 123).

The package ulem (https://ctan.org/pkg/ulem) does text mode underlining and allows line breaking as well as a number of other features. See also Section 19.10 [\hrulefill \& \dotfill], page 191, for producing a line for such things as a signature or placeholder.

\overline{math}

Put a horizontal line over math. For example: \overline{x+y}. The result looks like: $\overline{x+y}$. This differs from the accent command \bar (see Section 16.4 [Math accents], page 169).

\underbrace{math}

Put a brace under math. For example: $(1-\underbrace{1/2}+(1/2)-1/3)$. The result looks like: $(1 - \overbrace{1/2} + (1/2) - 1/3)$. You can attach text to the brace as a subscript (\_ or superscript (^) as here:

\begin{displaymath}
1+1/2+\underbrace{1/3+1/4}_{>1/2}+\underbrace{1/5+1/6+1/7+1/8}_{>1/2}\cdots
\end{displaymath}

The superscript appears on top of the expression, and so can look unconnected to the underbrace.

\overbrace{math}

Put a brace over math. For example: $\overbrace{x+x+\cdots+x}^{(k \text{ times})}$. The result looks like: $\overbrace{x + x + \cdots + x}^{(k \text{ times})}$.

\overrightarrow{math}

Put a right arrow over math. For example: $\overrightarrow{x+y}$. The result looks like: $\overrightarrow{x + y}$.

\overleftarrow{math}

Put a left arrow over math. For example: $\overleftarrow{a+b}$. The result looks like: $\overleftarrow{a + b}.$
The package **mathtools** ([https://ctan.org/pkg/mathtools](https://ctan.org/pkg/mathtools)) adds an over- and under-bracket, as well as some improvements on the braces.

### 16.6 Spacing in math mode

When typesetting mathematics, **\LaTeX** puts in spacing according to the normal rules for mathematics texts. If you enter \( y=mx \) then **\LaTeX** ignores the space and in the output the \( m \) is next to the \( x \), as \( y = mx \).

But **\LaTeX**’s rules occasionally need tweaking. For example, in an integral the tradition is to put a small extra space between the \( f(x) \) and the \( dx \), here done with the \( \, \) command:

\[
\int_0^1 f(x)\,dx
\]

**\LaTeX** provides the commands that follow for use in math mode. Many of these spacing definitions are expressed in terms of the math unit \( \text{mu} \). It is defined as 1/18 em, where the em is taken from the current math symbols family (see Section 14.1 [Units of length], page 138). Thus, a \( \text{thickspace} \) is something like 5/18 times the width of a ‘M’.

\[
; \quad \text{Synonym:} \ \text{thickspace}. \ \text{Normally} \ 5.0\text{mu plus} \ 5.0\text{mu}. \ \text{With the} \ \text{amsmath} \ \text{package, or as of the 2020-10-01} \ \text{\LaTeX} \ \text{release, can be used in} \ \text{text mode as well as} \ \text{math mode}; \ \text{otherwise, in math mode only.}
\]

\[
\negthickspace \quad \text{Normally} \ -5.0\text{mu plus} \ 2.0\text{mu minus} \ 4.0\text{mu}. \ \text{With the} \ \text{amsmath} \ \text{package, or as of the 2020-10-01} \ \text{\LaTeX} \ \text{release, can be used in} \ \text{text mode as well as} \ \text{math mode}; \ \text{otherwise, in math mode only.}
\]

\[
\; \quad \text{Synonym:} \ \text{medspace}. \ \text{Normally} \ 4.0\text{mu plus} \ 2.0\text{mu minus} \ 4.0\text{mu}. \ \text{With the} \ \text{amsmath} \ \text{package, or as of the 2020-10-01} \ \text{\LaTeX} \ \text{release, can be used in} \ \text{text mode as well as} \ \text{math mode}; \ \text{before that, in math mode only.}
\]

\[
\negmedspace \quad \text{Normally} \ -4.0\text{mu plus} \ 2.0\text{mu minus} \ 4.0\text{mu}. \ \text{With the} \ \text{amsmath} \ \text{package, or as of the 2020-10-01} \ \text{\LaTeX} \ \text{release, can be used in} \ \text{text mode as well as} \ \text{math mode}; \ \text{before that, in math mode only.}
\]

\[
, \quad \text{Synonym:} \ \text{thinspace}. \ \text{Normally} \ 3\text{mu}, \ \text{which is} \ 1/6 \ \text{em}. \ \text{Can be used in both} \ \text{math mode and text mode (see Section 19.8 [\text{thinspace} & \text{\negthinspace}], page 190).}
\]

This space is widely used, for instance between the function and the infinitesimal in an integral \( \int f(x)\,dx \) and, if an author does this, before punctuation in a displayed equation.

\[
\text{The antiderivative is}
\begin{equation}
3x^{-1/2}+3^{1/2},
\end{equation}
\]

\[
! \quad \text{Synonym:} \ \text{negthinspace}. \ \text{A negative thin space. Normally} \ -3\text{mu}. \ \text{With the} \ \text{amsmath} \ \text{package, or as of the 2020-10-01} \ \text{\LaTeX} \ \text{release, can be used in} \ \text{text mode as well as} \ \text{math mode}; \ \text{otherwise, the} \ \backslash! \ \text{command is math mode only.}
\]
but the \negthinspace command has always also worked in text mode (see Section 19.8 [\thinspace & \negthinspace], page 190).

\quad This is 18 mu, that is, 1 em. This is often used for space surrounding equations or expressions, for instance for the space between two equations inside a \texttt{displaymath} environment. It is available in both text and math mode.

\texttt{quad} A length of 2 quads, that is, 36 mu = 2 em. It is available in both text and math mode.

\subsection{16.6.1 \texttt{smash}}

Synopsis:

\texttt{\smash{subformula}}

Typeset subformula as if its height and depth were zero.

In this example the exponential is so tall that without the \texttt{smash} command \LaTeX{} would separate its line from the line above it, and the uneven line spacing might be unsightly.

To compute the tetration $\smash{2^{2^{2^2}}}$, evaluate from the top down, as $2^{2^4}=2^{16}=65536$.

(Because of the \texttt{smash} the printed expression could run into the line above so you may want to wait until the final version of the document to make such adjustments.)

This pictures the effect of \texttt{smash} by using \texttt{fbox} to surround the box that \LaTeX{} will put on the line. The \texttt{blackbar} command makes a bar extending from 10 points below the baseline to 20 points above.

\texttt{newcommand}{\blackbar}{\rule[-10pt]{5pt}{30pt}}
\fbox{\blackbar}
\fbox{\smash{\blackbar}}

The first box that \LaTeX{} places is 20 points high and 10 points deep. But the second box is treated by \LaTeX{} as having zero height and zero depth, despite that the ink printed on the page still extends well above and below the line.

The \texttt{smash} command appears often in mathematics to adjust the size of an element that surrounds a subformula. Here the first radical extends below the baseline while the second lies just on the baseline.

\begin{equation}
\sqrt{\sum_{0\leq k< n} f(k)}
\sqrt{\vphantom{\sum} \smash{\sum_{0\leq k< n}} f(k)}
\end{equation}

Note the use of \texttt{vphantom} to give the \texttt{sqrt} command an argument with the height of the \texttt{sum} (see Section 16.6.2 [\phantom & \vphantom & \hphantom], page 173).

While most often used in mathematics, the \texttt{smash} command can appear in other contexts. However, it doesn’t change into horizontal mode. So if it starts a paragraph then you should first put a \texttt{leavevmode}, as in the bottom line below.

Text above.

\texttt{smash\{smashed, no indent\} % no paragraph indent}
\leavevmode\smash{smashed, with indent} \% usual paragraph indent

The package mathtools has operators that provide even finer control over smashing a subformula box.

16.6.2 \phantom \& \vphantom \& \hphantom

Synopsis:

\phantom{subformula}

or

\vphantom{subformula}

or

\hphantom{subformula}

The \phantom command creates a box with the same height, depth, and width as subformula, but empty. That is, this command causes \LaTeX to typeset the space but not fill it with the material. Here \LaTeX will put a blank line that is the correct width for the answer, but will not show that answer.

\begin{displaymath}
\int x^2\,dx = \underline{\phantom{(1/3)x^3+C}}
\end{displaymath}

The \vphantom variant produces an invisible box with the same vertical size as subformula, the same height and depth, but having zero width. And \hphantom makes a box with the same width as subformula but with zero height and depth.

In this example, the tower of exponents in the second summand expression is so tall that \LaTeX places this expression further down than its default. Without adjustment, the two summand expressions would be at different levels. The \vphantom in the first expression tells \LaTeX to leave as much vertical room as it does for the tower, so the two expressions come out at the same level.

\begin{displaymath}
\sum_{j\in \{0,\ldots, 10\}\vphantom{3^{3^{3^j}}}} \sum_{i\in \{0,\ldots, 3^{3^{3^j}}\}} i\cdot j
\end{displaymath}

These commands are often used in conjunction with \smash. See Section 16.6.1 [\smash], page 172, which includes another example of \vphantom.

The three phantom commands appear often but note that \LaTeX provides a suite of other commands to work with box sizes that may be more convenient, including \makebox (see Section 20.1 [mbox \& makebox], page 197) as well as \settodepth (see Section 14.4 [settodepth], page 140), \settoheight (see Section 14.5 [settoheight], page 140), and \settowidth (see Section 14.6 [settowidth], page 141). In addition, the mathtools package has many commands that offer fine-grained control over spacing.

All three commands produce an ordinary box, without any special mathematics status. So to do something like attaching a superscript you should give it such a status, for example with the \operatorname command from the package amsmath.

While most often used in mathematics, these three can appear in other contexts. However, they don’t cause \LaTeX to change into horizontal mode. So if one of these starts a paragraph then you should prefix it with \leavevmode.
16.6.3 \mathstrut

Synopsis:

\mathstrut

The analogue of \strut for mathematics. See Section 19.13 [\strut], page 193.

The input $\sqrt{x} + \sqrt{x^i}$ gives output where the second radical is taller than the first. To add extra vertical space without any horizontal space, so that the two have the same height, use $\sqrt{x\mathstrut} + \sqrt{x^i\mathstrut}$.

The \mathstrut command adds the vertical height of an open parenthesis, (, but no horizontal space. It is defined as \vphantom{(}, so see Section 16.6.2 [\phantom & \vphantom & \hphantom], page 173, for more. An advantage over \strut is that \mathstrut adds no depth, which is often the right thing for formulas. Using the height of an open parenthesis is just a convention; for complete control over the amount of space, use \rule with a width of zero. See Section 23.8 [\rule], page 228.

16.7 Math styles

\TeX’s rules for typesetting a formula depend on the context. For example, inside a displayed equation, the input \( \sum_{0 \leq i < n} k^m = \frac{n^{m+1}}{m+1} + \text{lower order terms} \) will give output with the summation index centered below the summation symbol. But if that input is inline then the summation index is off to the right rather than below, so it won’t push the lines apart. Similarly, in a displayed context, the symbols in the numerator and denominator will be larger than for an inline context, and in display math subscripts and superscripts are further apart then they are in inline math.

\TeX uses four math styles.

• Display style is for a formula displayed on a line by itself, such as with \begin{equation} \end{equation}.

• Text style is for an inline formula, as with ‘so we have $\ldots$’.

• Script style is for parts of a formula in a subscript or superscript.

• Scriptscript style is for parts of a formula at a second level (or more) of subscript or superscript.

\TeX determines a default math style but you can override it with a declaration of \displaystyle, or \textstyle, or \scriptstyle, or \scriptscriptstyle.

In this example, the ‘Arithmetic’ line’s fraction will look scrunched.

\begin{tabular}{r|cc}
\textsc{Name} & \textsc{Series} & \textsc{Sum} \\
Arithmetic & $a+(a+b)+(a+2b)+\cdots+(a+(n-1)b)$ & $na+(n-1)n\frac{b}{2}$ \\
Geometric & $a+ab+ab^2+\cdots+ab^{n-1}$ & $a\frac{1-b^n}{1-b}$ \\
\end{tabular}

But because of the \displaystyle declaration, the ‘Geometric’ line’s fraction will be easy to read, with characters the same size as in the rest of the line.
Another example is that, compared to the same input without the declaration, the result of
\[
we\ get\ \$
\pi=2\cdot\int_{x=0}^1 \sqrt{1-x^2}\,dx$
will have an integral sign that is much taller. Note that here the \texttt{\displaystyle} applies to only part of the formula, and it is delimited by being inside curly braces, as ‘\texttt{\{\displaystyle \ldots\}}’.

The last example is a continued fraction.
\[
\begin{equation}
a_0+\frac{1}{\displaystyle a_1+\frac{\mathstrut 1}{\displaystyle a_2+\frac{\mathstrut 1}{\displaystyle a_3}}}
\end{equation}
\]
Without the \texttt{\displaystyle} declarations, the denominators would be set in script style and scriptscript style. (The \texttt{\mathstrut} improves the height of the denominators; see Section 16.6.3 [\mathstrut], page 174.)

16.8 Math miscellany
\LaTeX{} contains a wide variety of mathematics facilities. Here are some that don’t fit into other categories.

16.8.1 Colon character $\colon$ \& \texttt{\colon}
Synopsis, one of:
\[
\colon
\]
In mathematics, the colon character, $\colon$, is a relation.

With side ratios \(3:4\) and \(4:5\), the triangle is right.
Ordinary \LaTeX{} defines \texttt{\colon} to produce the colon character with the spacing appropriate for punctuation, as in set-builder notation \{x\colon 0\leq x<1\}.

But the widely-used \texttt{amsmath} package defines \texttt{\colon} for use in the definition of functions \texttt{f\colon D\to C}. So if you want the colon character as a punctuation then use \texttt{\mathpunct{\colon}}.

16.8.2 \texttt{\*}
Synopsis:
\[
\ast
\]
A multiplication symbol that allows a line break. If there is a break then \LaTeX{} puts a \texttt{\times} symbol, $\times$, before that break.

In \texttt{\( A_1\ast A_2\ast A_3\ast A_4\)}, if there is no line break then \LaTeX{} outputs it as though it were \texttt{\( A_1 A_2 A_3 A_4\)}. If a line break does happen, for example between the two middle ones, then \LaTeX{} sets it like \texttt{\( A_1 A_2 \times \)}, followed by the break, followed by \texttt{\( A_3 A_4\)}. 

16.8.3 \texttt{\frac}

Synopsis:

\texttt{\frac{numerator}{denominator}}

Produces the fraction. Used as: \begin{displaymath} \frac{1}{\sqrt{2\pi\sigma}} \end{displaymath}. In inline math mode it comes out small; see the discussion of \texttt{\displaystyle} (see Chapter 16 [Math formulas], page 148).

16.8.4 \texttt{\sqrt}

Synopsis, one of:

\texttt{\sqrt{arg}}

\texttt{\sqrt[root-number]{arg}}

The square root, or optionally other roots, of \textit{arg}. The optional argument \textit{root-number} gives the root, i.e., enter the cube root of \textit{x+y} as \texttt{\sqrt[3]{x+y}}. It comes out like this: \sqrt[3]{x+y}. The size of the radical grows with that of \textit{arg} (as the height of the radical grows, the angle on the leftmost part gets steeper, until for a tall enough \textit{arg}, it is vertical).

\LaTeX{} has a separate \texttt{\surd} symbol for making a square root without \textit{arg} (see Section 16.2 [Math symbols], page 149).

16.8.5 \texttt{\stackrel}

Synopsis:

\texttt{\stackrel{text}{relation}}

Put \textit{text} above \textit{relation}. To put a function name above an arrow enter \texttt{\stackrel{f}{\longrightarrow}}. The result looks like this: $\stackrel{f}{\longrightarrow}$. 
17 Modes

As \LaTeX{} processes your document, at any point it is in one of six modes. They fall into three categories of two each, the horizontal modes, the math modes, and the vertical modes. Some commands only work in one mode or another (in particular, many commands only work in one of the math modes), and error messages will refer to these.

- **Paragraph mode** (in plain \TeX{} this is called *horizontal mode*) is what \LaTeX{} is in when processing ordinary text. It breaks the input text into lines and finds the positions of line breaks, so that in vertical mode page breaks can be done. This is the mode \LaTeX{} is in most of the time.

  *LR mode* (for left-to-right mode; in plain \TeX{} this is called *restricted horizontal mode*) is in effect when \LaTeX{} starts making a box with an \verb|\mbox| command. As in paragraph mode, \LaTeX{}'s output is a string of words with spaces between them. Unlike in paragraph mode, in LR mode \LaTeX{} never starts a new line, it just keeps going from left to right. (Although \LaTeX{} will not complain that the LR box is too long, when it is finished and next tries to put that box into a line, it might well complain that the finished LR box won’t fit there.)

- **Math mode** is when \LaTeX{} is generating an inline mathematical formula.

  *Display math mode* is when \LaTeX{} is generating a displayed mathematical formula. (Displayed formulas differ somewhat from inline ones. One example is that the placement of the subscript on \verb|\int| differs in the two situations.)

- **Vertical mode** is when \LaTeX{} is building the list of lines and other material making the output page, which comprises insertion of page breaks. This is the mode \LaTeX{} is in when it starts a document.

  *Internal vertical mode* is in effect when \LaTeX{} starts making a \verb|\vbox|. It has not such thing as page breaks, and as such is the vertical analogue of LR mode.

For instance, if you begin a \LaTeX{} article with ‘Let \( x \) be …’ then these are the modes: first \LaTeX{} starts every document in vertical mode, then it reads the ‘L’ and switches to paragraph mode, then the next switch happens at the ‘\( \)’ where \LaTeX{} changes to math mode, and then when it leaves the formula it pops back to paragraph mode.

Paragraph mode has two subcases. If you use a \verb|\parbox| command or a \verb|\minipage| then \LaTeX{} is put into paragraph mode. But it will not put a page break here. Inside one of these boxes, called a *parbox*, \LaTeX{} is in *inner paragraph mode*. Its more usual situation, where it can put page breaks, is *outer paragraph mode* (see Chapter 10 [Page breaking], page 104).

17.1 \verb|\ensuremath|

Synopsis:

\inputenc{\ensuremath}\{formula\}

Ensure that *formula* is typeset in math mode.

For instance, you can redefine commands that ordinarily can be used only in math mode, so that they can be used both in math and in plain text.

\begin{verbatim}
\newcommand{\dx}{\ensuremath{dx}}
\end{verbatim}

In $\int f(x)\, \dx$, the \verb|\dx| is an infinitesimal.
Caution: the \texttt{ensuremath} command is useful but not a panacea.

\begin{verbatim}
\newcommand{\alf}{\ensuremath{\alpha}}
You get an alpha in text mode: \alf.
But compare the correct spacing in $\alf+\alf$ with that in \alf+\alf.
\end{verbatim}

Best is to typeset math things in a math mode.
18 Page styles

The style of a page determines where \LaTeX\ places the components of that page, such as headers and footers, and the text body. This includes pages in the main part of the document but also includes special pages such as the title page of a book, a page from an index, or the first page of an article.

The package \texttt{fancyhdr} is commonly used for constructing page styles. See its documentation.

18.1 \texttt{\textbackslash make\_title}

Synopsis:

\texttt{\textbackslash make\_title}

Generate a title. In the standard classes the title appears on a separate page, except in the \texttt{article} class where it is at the top of the first page. (See Section 3.1 [Document class options], page 10, for information about the \texttt{titlepage} document class option.)

This example shows \texttt{\textbackslash make\_title} appearing in its usual place, immediately after \texttt{\textbackslash begin\{document\}}.

\begin{verbatim}
\documentclass\{article\}
\title\{Constructing a Nuclear Reactor Using Only Coconuts\}
\author\{Jonas Grumby\thanks\{With the support of a Ginger Grant from the Roy Hinkley Society.\} \\skipper, \textit{Minnow}\}
\and\Willy Gilligan\thanks\{Thanks to the Mary Ann Summers foundation and to Thurston and Lovey Howell.\} \\mate, \textit{Minnow}\}
\date\{1964-Sep-26\}
\begin\{document\}
\maketitle
Just sit right back and you'll hear a tale, a tale of a fateful trip. That started from this tropic port, aboard this tiny ship. The mate was a mighty sailin' man, the Skipper brave and sure. Five passengers set sail that day for a three hour tour. A three hour tour.

... You tell \LaTeX\ the information used to produce the title by making the following declarations. These must come before the \texttt{\textbackslash make\_title}, either in the preamble or in the document body.

\texttt{\textbackslash author\{name1 \and name2 \and ...\}}

Required. Declare the document author or authors. The argument is a list of authors separated by \texttt{\and} commands. To separate lines within a single author's entry, for instance to give the author's institution or address, use a double backslash, \texttt{\\}. If you omit the \texttt{\author} declaration then you get 'LaTeX Warning: No \author given'.
\date{text}

Optional. Declare text to be the document’s date. The text doesn’t need to be in a date format; it can be any text at all. If you omit \date then \TeX uses the current date (see Section 23.9 \today, page 228). To have no date, instead use \date{}.

\thanks{text}

Optional. Produce a footnote. You can use it in the author information for acknowledgements as illustrated above, but you can also use it in the title, or anywhere that a footnote mark makes sense. It can be any text at all so you can use it for any purpose, such as to print an email address.

\title{text}

Required. Declare text to be the title of the document. Get line breaks inside text with a double backslash, \\\. If you omit the \title declaration then the \maketitle command yields error ‘\LaTeX Error: No \title given’.

To make your own title page, see Section 8.26 \titlepage, page 95. You can either create this as a one-off or you can include it as part of a renewed \maketitle command. Many publishers will provide a class to use in place of \article that formats the title according to their house requirements.

18.2 \pagenumbering

Synopsis:

\pagenumbering{number-style}

Specifies the style of page numbers, and resets the page number. The numbering style is reflected on the page, and also in the table of contents and other page references. This declaration has global scope so its effect is not stopped by an end of group such as a closing brace or an end of environment.

By default, \TeX numbers pages starting at 1, using Arabic numerals.

The argument number-style is one of the following (see also Section 13.1 [alph Alph arabic \arabic roman \Roman \fnsymbol], page 133).

arabic Arabic numerals: 1, 2, …

roman lowercase Roman numerals: i, ii, …

Roman uppercase Roman numerals: I, II, …

alph lowercase letters: a, b, … If you have more than 26 pages then you get ‘\LaTeX Error: Counter too large’.

Alph uppercase letters: A, B, … If you have more than 26 pages then you get ‘\LaTeX Error: Counter too large’.

gobble no page number is output, though the number is still reset. References to that page also are blank.

This setting does not work with the popular package \hyperref, so to omit page numbers you may want to instead use \pagestyle{empty} or \thispagestyle{empty}.
If you want to typeset the page number in some other way, or change where the page number appears on the page, see Section 18.3 [\pagestyle], page 181, (in short: use the fancyhdr package). The list above of \LaTeX{}'s built-in numbering styles cannot be extended.

Traditionally, if a document has front matter—preface, table of contents, etc.—then it is numbered with lowercase Roman numerals. The main matter of a document uses arabic. \LaTeX{} implements this, by providing explicit commands for the different parts (see Section 6.7 [\frontmatter & \mainmatter & \backmatter], page 43).

As an explicit example, before the ‘Main’ section the pages are numbered ‘a’, etc. Starting on the page containing the \pagenumbering call in that section, the pages are numbered ‘1’, etc.

\begin{document}\pagenumbering{alph}

\section{Main}\pagenumbering{arabic}

If you want to change the value of the page number, then you manipulate the page counter (see Chapter 13 [Counters], page 133).

18.3 \pagestyle

Synopsis:

\pagestyle{style}

Declaration that specifies how the page headers and footers are typeset, from the current page onwards.

A discussion with an example is below. First, however: the package fancyhdr is now the standard way to manipulate headers and footers. New documents that need to do anything other than one of the standard options below should use this package. See its documentation (https://ctan.org/pkg/fancyhdr).

Values for style:

plain The header is empty. The footer contains only a page number, centered.
empty The header and footer are both empty.
headings Put running headers and footers on each page. The document style specifies what goes in there; see the discussion below.

myheadings Custom headers, specified via the \markboth or the \markright commands.

Some discussion of the motivation for \LaTeX{}'s mechanism will help you work with the options headings or myheadings. The document source below produces an article, two-sided, with the pagestyle headings. On this document’s left hand pages, \LaTeX{} wants (in addition to the page number) the title of the current section. On its right hand pages \LaTeX{} wants the title of the current subsection. When it makes up a page, \LaTeX{} gets this information from the commands \leftmark and \rightmark. So it is up to \section and \subsection to store that information there.

\documentclass[twoside]{article}
\pagestyle{headings}
\begin{document}
... \section{Section 1} ...
\subsection{Subsection 1.1} ...
\section{Section 2} ...
\subsection{Subsection 2.1} ...
\subsection{Subsection 2.2} ...

Suppose that the second section falls on a left page. Although when the page starts it is in the first section, \LaTeX{} will put ‘Section 2’ in the left page header. As to the right header, if no subsection starts before the end of the right page then \LaTeX{} blanks the right hand header. If a subsection does appear before the right page finishes then there are two cases. If at least one subsection starts on the right hand page then \LaTeX{} will put in the right header the title of the first subsection starting on that right page. If at least one of 2.1, 2.2, ..., starts on the left page but none starts on the right then \LaTeX{} puts in the right hand header the title of the last subsection to start, that is, the one in effect during the right hand page.

To accomplish this, in a two-sided article, \LaTeX{} has \texttt{section} issue a command \texttt{markboth}, setting \texttt{leftmark} to ‘Section 2’ and setting \texttt{rightmark} to an empty content. And, \LaTeX{} has \texttt{subsection} issue a command \texttt{markright}, setting \texttt{rightmark} to ‘Subsection 2.1’, etc.

Here are the descriptions of \texttt{markboth} and \texttt{markright}:

\texttt{markboth\{left-head\}\{right-head\}}

Sets both the right hand and left hand heading information for either a page style of \texttt{headings} or \texttt{myheadings}. A left hand page heading \texttt{left-head} is generated by the last \texttt{markboth} command before the end of the page. A right hand page heading \texttt{right-head} is generated by the first \texttt{markboth} or \texttt{markright} that comes on the page if there is one, otherwise by the last one that came before that page.

\texttt{markright\{right-head\}}

Sets the right hand page heading, leaving the left unchanged.

18.4 \texttt{thispagestyle}

Synopsis:

\texttt{thispagestyle\{style\}}

Works in the same way as the \texttt{pagestyle} (see Section 18.3 [\texttt{pagestyle}], page 181), except that it changes to \texttt{style} for the current page only. This declaration has global scope, so its effect is not delimited by braces or environments.

Often the first page of a chapter or section has a different style. For example, this \LaTeX{} book document has the first page of the first chapter in \texttt{plain} style, as is the default (see Chapter 18 [Page styles], page 179).

\documentclass{book}
\pagestyle{headings}
\begin{document}
\chapter{First chapter}
\chapter{Second chapter}\thispagestyle{empty}

The plain style has a page number on it, centered in the footer. To make the page entirely empty, the command \thispagestyle{empty} immediately follows the second \chapter.

18.5 \thepage

If you want to change the appearance of page numbers only in the page headers, for example by adding an ornament, typesetting in small caps, etc., then the fancyhdr package, as mentioned in a previous section, is the best approach.

On the other hand, you may want to change how page numbers are denoted everywhere, including the table of contents and cross-references, as well as the page headers. In this case, you should redefine \thepage, which is the command \LaTeX{} uses for the representation of page numbers.

However, \thepage{} should do any typesetting or other complicated maneuvers, but merely expand to the intended page number representation. The results of a complicated redefinition of \thepage{} are not predictable, but \LaTeX{}’s report of page numbers in diagnostic messages, at least, will become unusable.

There is some discussion of this issue at https://tex.stackexchange.com/questions/687258.
19 Spaces

\LaTeX{} has many ways to produce white space, or filled space. Some of these are best suited to mathematical text; for these see Section 16.6 [Spacing in math mode], page 171.

19.1 \enspace & \quad & \qquad

Synopsis, one of:
\begin{itemize}
  \item \enspace
  \item \quad
  \item \qquad
\end{itemize}

Insert a horizontal space of 1/2 em, 1 em, or 2 em. The em is a length defined by a font designer, often thought of as being the width of a capital M. One advantage of describing space in ems is that it can be more portable across documents than an absolute measurement such as points (see [Lengths/em], page 139).

This puts a suitable gap between two graphics.
\begin{center}
\includegraphics{womensmile.png}\qquad\includegraphics{mensmile.png}
\end{center}

See Section 16.6 [Spacing in math mode], page 171, for \quad and \qquad. These are lengths from centuries of typesetting and so may be a better choice in many circumstances than arbitrary lengths, such as you get with \hspace.

19.2 \hspace

Synopsis, one of:
\begin{itemize}
  \item \hspace{\textit{length}}
  \item \hspace*{\textit{length}}
\end{itemize}

Insert the amount \textit{length} of horizontal space. The \textit{length} can be positive, negative, or zero; adding a negative amount of space is like backspacing. It is a rubber length, that is, it may contain a plus or minus component, or both (see Chapter 14 [Lengths], page 137). Because the space is stretchable and shrinkable, it is sometimes called glue.

This makes a line with ‘Name:’ an inch from the right margin.
\begin{verbatim}
\noindent\makebox[\linewidth][r]{Name:\hspace*{1in}}
\end{verbatim}

The *-form inserts horizontal space that is non-discardable. More precisely, when \TeX{} breaks a paragraph into lines any white space—glues and kerns—that come at a line break are discarded. The *-form avoids that (technically, it adds a non-discardable invisible item in front of the space).

In this example
\begin{verbatim}
\parbox{0.8\linewidth}{%
Fill in each blank: Four \hspace*{1in} and seven years ago our fathers brought forth on this continent, a new \hspace*{1in}, conceived in \hspace*{1in}, and dedicated to the proposition that all men are created \hspace*{1in}.}
\end{verbatim}
the 1 inch blank following ‘conceived in’ falls at the start of a line. If you erase the * then \LaTeX discards the blank.

Here, the \hspace separates the three graphics.

\begin{center}
\includegraphics{lion.png}\hspace{1cm minus 0.25cm}\includegraphics{tiger.png}\hspace{1cm minus 0.25cm}\includegraphics{bear.png}
\end{center}

Because the argument to each \hspace has minus 0.25cm, each can shrink a little if the three figures are too wide. But each space won’t shrink more than 0.25 cm (see Chapter 14 [Lengths], page 137).

19.3 \hfill

Synopsis:

\hfill

Produce a rubber length which has no natural space but that can stretch horizontally as far as needed (see Chapter 14 [Lengths], page 137).

This creates a one-line paragraph with ‘Name:’ on the left side of the page and ‘Quiz One’ on the right.

\noindent Name: \hfill Quiz One

The \hfill command is equivalent to \hspace{\fill} and so the space can be discarded at line breaks. To avoid that instead use \hspace*{\fill} (see Section 19.2 [\hspace], page 184).

Here the graphs are evenly spaced in the middle of the figure.

\begin{figure}
\hspace*{\fill} \vcenteredhbox{\includegraphics{graph0.png}} \hfill \vcenteredhbox{\includegraphics{graph1.png}} \hspace*{\fill}
\caption{Comparison of two graphs} \label{fig:twographs}
\end{figure}

Note the \hspace*’s where the space could otherwise be dropped.

19.4 \hss

Synopsis:

\hss

Produce a horizontal space that is infinitely shrinkable as well as infinitely stretchable (this command is a \TeX primitive). \LaTeX authors should reach first for the \makebox command to get the effects of \hss (see Section 20.1 [\mbox & \makebox], page 197).
Here, the first line’s `\hss` makes the Z stick out to the right, overwriting the Y. In the second line the Z sticks out to the left, overwriting the X.

```
X \hbox to 0pt{Z\hss} Y
X \hbox to 0pt{\hss Z} Y
```

Without the `\hss` you get something like ‘Overfull \hbox (6.1111pt too wide) detected at line 20’.

### 19.5 `\spacefactor`

**Synopsis:**

```
\spacefactor=integer
```

Influence \LaTeX{}’s stretching and shrinking of glue. Few user-level documents need to use this.

While \LaTeX{} is laying out the material, it may stretch or shrink the gaps between words. This space is not a character; it is called the *interword glue*; see Section 19.2 `[\hskip]`, page 184). The `\spacefactor` parameter (a \TeX{} primitive) allows you to, for instance, have the space after a period stretch more than the space after a word-ending letter.

After \LaTeX{} places each character, or rule or other box, it sets a parameter called the *space factor*. If the next thing in the input is a space then this parameter affects how much stretching or shrinking can happen. A space factor that is larger than the normal value means that the glue can stretch more and shrink less. Normally, the space factor is 1000. This value is in effect following most characters, and any non-character box or math formula. But it is 3000 after a period, exclamation mark, or question mark, 2000 after a colon, 1500 after a semicolon, 1250 after a comma, and 0 after a right parenthesis or bracket, or closing double quote or single quote. Finally, it is 999 after a capital letter.

If the space factor $f$ is 1000 then the glue gap will be the font’s normal space value (for Computer Modern Roman 10 point this is 3.3333 pt). Otherwise, if the space factor $f$ is greater than 2000 then \TeX{} adds the font’s extra space value (for Computer Modern Roman 10 point this is 1.11111 pt), and then the font’s normal stretch value is multiplied by $f/1000$ and the normal shrink value is multiplied by $1000/f$ (for Computer Modern Roman 10 point these are 1.66666 and 1.11111 pt).

For example, consider the period ending ‘A man’s best friend is his dog.’. After it, \TeX{} puts in a fixed extra space, and also allows the glue to stretch 3 times as much and shrink 1/3 as much, as the glue after friend or any of the other words, since they are not followed by punctuation.

The rules for space factors are even more complex because they play additional roles. In practice, there are two consequences. First, if a period or other punctuation is followed by a right parenthesis or bracket, or right single or double quote then the spacing effect of that period carries through those characters (that is, the following glue will have increased stretch and shrink). Second, if punctuation comes after a capital letter then the normal effect of the period is does not happen, so you get an ordinary space. This second case also affects abbreviations that do not end in a capital letter (see Section 19.5.1 `[\@]`, page 187).

You can only use `\spacefactor` in paragraph mode or LR mode (see Chapter 17 [Modes], page 177). You can see the current value with `\the\spacefactor` or `\showthe\spacefactor`. 
Finally, not especially related to \texttt{\spacefactor} itself: if you get errors like ‘\texttt{You can't use `\spacefactor' in vertical mode}', or ‘\texttt{You can't use `\spacefactor' in math mode}', or ‘\texttt{Improper `\spacefactor}’ then you have probably tried to redefine an internal command. See Section 12.3 \texttt{[makeatletter & \makeatother]}, page 115.

19.5.1 \texttt{@}

Synopsis:

\texttt{capital-letter@}.

Treat a period (or other punctuation) as sentence-ending, where \LaTeX{} would otherwise think it is part of an abbreviation. \LaTeX{} thinks that a period ends an abbreviation if the period comes after a capital letter, and otherwise thinks the period ends the sentence.

This example shows the two cases to remember.

\begin{quote}
The songs \texttt{\textit{Red Guitar}}, etc.\, are by Loudon Wainwright\textsc{~III}@. The first period ends the abbreviation ‘\texttt{etc.}’ but not the sentence. The backslash-space, \texttt{\ }, produces a mid-sentence space. The second period ends the sentence, despite it being preceded by a capital letter. We tell \LaTeX{} that it ends the sentence by putting \texttt{@} before it.

So: if you have a capital letter followed by a period that ends the sentence, then put \texttt{@} before the period. This holds even if there is an intervening right parenthesis or bracket, or right single or double quote, because the spacing effect of that period carries through those characters. For example, this

\begin{quote}
Use the \texttt{\textit{Instructional Practices Guide}}, (a book by the MAA)\texttt{@}.
\end{quote}

will have correct inter-sentence spacing after the period.

The \texttt{@} command is only for text modes. If you use it outside of a text mode then you get the error ‘\texttt{You can't use `\spacefactor' in vertical mode}’ (see Chapter 17 \texttt{[Modes]}, page 177).

All the above applies equally to question marks and exclamation points as periods, since all are sentence-ending punctuation, and \LaTeX{} increases the space after each in the same way, when they end a sentence. \LaTeX{} also increases spacing after colon, semicolon, and comma characters (see Section 19.5 \texttt{[spacefactor]}, page 186).

In addition: the converse case is a period (or other punctuation) that does not end a sentence. For that case, follow the period with a backslash-space, (\texttt{\ }), or a tie, (\texttt{~}), or \texttt{@}. Examples are \texttt{Nat.\ Acad.\ Science}, and \texttt{Mr.\ Bean}, and \texttt{(manure, etc.\@)} for sale (note in the last one that the \texttt{@} comes after the period but before the closing parenthesis).

19.5.2 \texttt{\frenchspacing} \& \texttt{\nonfrenchspacing}

Synopsis, one of:

\texttt{\frenchspacing} \nonfrenchspacing

\texttt{\frenchspacing} causes \LaTeX{} to make spacing after all punctuation, including periods, be the same as the space between words in the middle of a sentence. \texttt{\nonfrenchspacing} switches back to the default handling in which spacing after most punctuation stretches or shrinks differently than a word space (see Section 19.5 \texttt{[spacefactor]}, page 186).
In American English, the typesetting tradition is to adjust, typically increasing, the space after punctuation more than the space between words that are in the middle of a sentence. Declaring `\frenchspacing` (the command is inherited from plain \TeX) switches to the tradition that all spaces are treated equally.

If your \LaTeX document specifies the language being used, for example with the `babel` package, the necessary settings should be taken care of for you.

**19.5.3 `\normalsfcodes`**

**Synopsis:**

`\normalsfcodes`

Reset the \LaTeX space factors to the default values (see Section 19.5 `[\spacefactor]`, page 186).

**19.6 Backslash-space, `\`**

This section refers to the command consisting of two characters, a backslash followed by a space. **Synopsis:**

`\`


When you type one or more blanks between words, \LaTeX produces whitespace that is different than an explicit space. This illustrates:

\begin{tabular}{rl}
One blank:& makes some space \\
Three blanks:& in a row \\
Three spaces:& in a row \\
\end{tabular}

On the first line \LaTeX puts some space after the colon. On the second line \LaTeX collapses the three blanks to output one whitespace, so you end with the same space after the colon as in the first line. \LaTeX would similarly collapse them to a single whitespace if one, two or all of the three blanks were replaced by a tab, or by a newline. However, the bottom line asks for three spaces so the white area is wider. That is, the backslash-space command creates a fixed amount of horizontal space. (Note that you can define a horizontal space of any width at all with `\hspace`; see Section 19.2 `[\hspace]`, page 184.)

The backslash-space command has two main uses. It is often used after control sequences to keep them from gobbling the blank that follows, as after `\TeX` in `\TeX\` (or `\LaTeX\`). (But using curly braces has the advantage of still working whether the next character is a blank or any other non-letter, as in `\TeX{}` (or `\LaTeX{}`) in which `{}` can be added after `\LaTeX` as well as after `\TeX\`). The other common use is that it marks a period as ending an abbreviation instead of ending a sentence, as in `Prof. Smith` or `Jones et al. (1993)` (see Section 19.5.1 `[\@]`, page 187).

Under normal circumstances, `\TAB` and `\NEWLINE` are equivalent to backslash-space, `\`.

In order to allow source code indentation, under normal circumstances, \TeX ignores leading blanks in a line. So the following prints `one word`:

one
where the white space between ‘one’ and ‘word’ is produced by the newline after ‘one’, not by the space before ‘word’.

19.7 ~, \nobreakspace

Synopsis:

\texttt{before\textasciitilde after}

The tie character, ~, produces a space between \texttt{before} and \texttt{after} at which the line will not be broken. By default the white space has length 3.33333 pt plus 1.66666 pt minus 1.11111 pt (see Chapter 14 [Lengths], page 137). The command \texttt{\nobreakspace} and the Unicode input character U+00A0 (also in many 8-bit encodings) are synonyms.

Note that the word ‘tie’ has this meaning in the T\LaTeX/Texinfo community; this differs from the typographic term “tie”, which is a diacritic in the shape of an arc, called a “tie-after” accent in The T\LaTeX{}book.

Here \LaTeX{} will not break the line between the final two words:

\texttt{Thanks to Prof.~Lerman.}

In addition, despite the period, \LaTeX{} does not use the end-of-sentence spacing (see Section 19.5.1 [\texttt{\@}].

Ties prevent a line break where that could cause confusion. They also still allow hyphenation (of either of the tied words), so they are generally preferable to putting consecutive words in an \texttt{\mbox} (see Section 20.1 [\texttt{\mbox} & \texttt{\makebox}], page 197).

Exactly where ties should be used is something of a matter of taste, sometimes alarmingly dogmatic taste, among readers. Nevertheless, here are some usage models, many of them from The T\LaTeX{}book.

- Between an enumerator label and number, such as in references: Chapter~12, or Theorem\texttt{\textasciitilde \ref{th:Wilsons}}, or Figure\texttt{\textasciitilde \ref{fig:KGraph}}.
- When cases are enumerated inline: \texttt{(b)\textasciitilde Show that $f(x)$ is (1)~continuous, and (2)~bounded.}
- Between a number and its unit: \texttt{$745.7.8\textasciitilde$watts} (the \texttt{\textsc{siunitx}} package has a special facility for this) or \texttt{144\textasciitilde eggs}. This includes between a month and day number in a date: October\textasciitilde 12 or 12\textasciitilde Oct. In general, in any expressions where numbers and abbreviations or symbols are separated by a space: AD\textasciitilde 565, or 2:50\textasciitilde pm, or Boeing\textasciitilde 747, or 268\textasciitilde Plains Road, or \texttt{\$1.4\textasciitilde billion}. Other common choices here are a thin space (see Section 19.8 [\texttt{\textsc{\textit{\textasciitilde}} \textasciitilde}] and \texttt{\textasciitilde \textsc{\textasciitilde}}). Between mathematical symbols in apposition with nouns: dimension\textasciitilde$d$ or function\textasciitilde$f(x)$ (but compare \texttt{length $l$ or more}). When a symbol is a tightly bound object of a preposition: of\textasciitilde$x$, or from $0$ to\textasciitilde$1$, or in common with\textasciitilde$m$.
- Between mathematical phrases are rendered in words: equals\textasciitilde$n$ or less than\textasciitilde$\epsilon$, or given\textasciitilde$X$, or modulo\textasciitilde$p$\textasciitilde$e$ for all large\textasciitilde$n$ (but compare is\textasciitilde$15$ with is $15\times$ times the height). Between mathematical symbols in apposition with nouns: dimension\textasciitilde$d$ or function\textasciitilde$f(x)$ (but compare with length\textasciitilde$1$ or more). When a symbol is a tightly bound object of a preposition: of\textasciitilde$x$, or from $0$ to\textasciitilde$1$, or in common with\textasciitilde$m$.
- Between symbols in series: \texttt{$1$\textasciitilde$2$, $3$\textasciitilde$4$}, or\textasciitilde$2$ or \texttt{$1$, $2$, \ldots, $n$}.
- Between a person’s given names and between multiple surnames: Donald\textasciitilde E.~Knuth, or Luis\textasciitilde I.~Trabb\textasciitilde Pardo, or Charles\textasciitilde XII—but you must give \LaTeX{} places to break the line so you might do Charles Louis Xavier\textasciitilde Joseph de\textasciitilde la Vall\textasciitilde ‘ee\textasciitilde Poussin.}
19.8 \texttt{\textbackslash thinspace} & \texttt{\textbackslash negthinspace}

Synopsis, one of:

\begin{verbatim}
\textbackslash thinspace
\textbackslash negthinspace
\end{verbatim}

These produce unbreakable and unstretchable spaces of 1/6 em and -1/6 em, respectively. These are the text mode equivalents of \ and \! (see [Spacing in math mode/\textbackslash thinspace], page 171).

You can use \, as a synonym for \textbackslash thinspace in text mode.

One common use of \textbackslash thinspace is as the space between nested quotes:

\begin{verbatim}
Killick replied, "I heard the Captain say, 'Ahoy there.'\textbackslash thinspace"
\end{verbatim}

Another use is that some style guides call for a \textbackslash thinspace between an ellipsis and a sentence ending period (other style guides, think the three dots and/or four dots are plenty). Another style-specific use is between initials, as in D.\textbackslash thinspace E.\ Knuth.

\LaTeX{} provides a variety of similar spacing commands for math mode (see Section 16.6 [Spacing in math mode], page 171). With the \texttt{amsmath} package, or as of the 2020-10-01 \LaTeX{} release, they can be used in text mode as well as math mode, including \! for \texttt{\textbackslash negthinspace}; but otherwise, they are available only in math mode.

19.9 \textbackslash /

Synopsis:

\begin{verbatim}
 before-character\textbackslash after-character
\end{verbatim}

Insert an italic correction, a small space defined by the font designer for each character (possibly zero), to avoid the character colliding with whatever follows. When you use \textbackslash /, \LaTeX{} takes the correction from the font metric file, scales it by any scaling that has been applied to the font, and then inserts that much horizontal space.

Here, were it not for the \textbackslash /, the \texttt{before-character} italic f would hit the \texttt{after-character} roman H

\begin{verbatim}
\newcommand{\companylogo}{\{\textit{f}\}/\H}
\end{verbatim}

because the italic letter f leans far to the right.

If \texttt{after-character} is a period or comma then don’t insert an italic correction since those punctuation symbols are so low to the baseline already. However, with semicolons or colons, as well as with normal letters, the italic correction can help. It is typically used between a switch from italic or slanted fonts to an upright font.

When you use commands such as \texttt{\textbf{emph}} and \texttt{\textit{textit}} and \texttt{\textsl{texts1}} to change fonts, \LaTeX{} automatically inserts the italic correction when needed (see Section 4.2 [Font styles], page 20). However, declarations such as \texttt{\em} and \texttt{\itshape} and \texttt{\slshape} do not automatically insert italic corrections.

Upright characters can also have an italic correction. An example where this is needed is the name \texttt{pdf/\LaTeX}. However, most upright characters have a zero italic correction. Some font creators do not include italic correction values even for italic fonts.

Technically, \LaTeX{} uses another font-specific value, the so-called slant parameter (namely \texttt{\fontdimen1}), to determine whether to possibly insert an italic correction, rather than tying the action to particular font commands.
There is no concept of italic correction in math mode; math spacing is done in a different way.

19.10 \hrulefill & \dotfill

Synopsis, one of:
\hrulefill
\dotfill

Produce an infinite horizontal rubber length (see Chapter 14 [Lengths], page 137) that \LaTeX fills with a rule (that is, a line) or with dots, instead of white space.

This outputs a line 2 inches long.

Name: \makebox[2in]{\hrulefill}

This example, when placed between blank lines, creates a paragraph that is left and right justified and where the middle is filled with evenly spaced dots.

\noindent John Aubrey, RN \dotfill{} Melbury Lodge

To make the rule or dots go to the line's end use \null at the start or end.

To change the rule's thickness, copy the definition and adjust it, as here

\renewcommand{\hrulefill}{%
  \leavevmode\leaders\hrule height 1pt\hfill\kern0pt }

which changes the default thickness of 0.4 pt to 1 pt. Similarly, adjust the dot spacing as with

\renewcommand{\dotfill}{%
  \leavevmode\cleaders\hbox to 1.00em{\hss .\hss }\hfill\kern0pt }

which changes the default length of 0.33 em to 1.00 em.

This example produces a line for a signature.

\begin{minipage}{4cm}
\centering
\hrulefill\\
Signed
\end{minipage}

The line is 4 cm long.

19.11 \bigskip & \medskip & \smallskip

Synopsis, one of:
\bigskip
\medskip
\smallskip

Produce an amount of vertical space, large or medium-sized or small. These commands are fragile (see Section 12.11 [\protect], page 123).

Here the skip suggests the passage of time (from The Golden Ocean by O'Brian).

Mr Saumarez would have something rude to say to him, no doubt: he was at home again, and it was delightful.
Each command is associated with a length defined in the document class file.

\bigskip
``A hundred and fifty-seven miles and one third, in twenty-four hours,''
said Peter.

Because each command is a \vspace, if you use it in mid-paragraph then it will insert
its vertical space between the line in which you use it and the next line, not necessarily at
the place that you use it. So these are best between paragraphs.

The commands \bigbreak, \medbreak, and \smallbreak are similar but also suggest
to \LaTeX that this is a good place to put a page break (see Section 19.12 [\bigbreak &
\medbreak & \smallbreak], page 192.

19.12 \bigbreak & \medbreak & \smallbreak

Synopsis, one of:

\bigbreak
\medbreak
\smallbreak

Produce a vertical space that is big or medium-sized or small, and suggest to \LaTeX that
this is a good place to break the page. (The associated penalties are respectively −200,
−100, and −50.)

See Section 19.11 [\bigskip & \medskip & \smallskip], page 191, for more. These
commands produce the same vertical space but differ in that they also remove a preceding
vertical space if it is less than what they would insert (as with \addvspace). In addition,
they terminate a paragraph where they are used: this example

abc\bigbreak def ghi

jkl mno pqr

will output three paragraphs, the first ending in ‘abc’ and the second starting, after an extra
vertical space and a paragraph indent, with ‘def’.
19.13 \strut

Synopsis:

\strut

Ensure that the current line has height at least 0.7\baselineskip and depth at least 0.3\baselineskip. Essentially, \LaTeX inserts into the line a rectangle having zero width, \rule[-0.3\baselineskip]{0pt}{\baselineskip} (see Section 23.8 \rule, page 228). The \baselineskip changes with the current font or fontsize.

In this example the \strut keeps the box inside the frame from having zero height.

\setlength{\fboxsep}{0pt}\framebox[2in]{\strut}

This example has four lists. In the first there is a much bigger gap between items 2 and 3 than there is between items 1 and 2. The second list fixes that with a \strut at the end of its first item’s second line.

\setlength{\fboxsep}{0pt}
\noindent\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \parbox[t]{15pt}{test \ test}
\item test
\item test
\end{enumerate}
\end{minipage}\
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \parbox[t]{15pt}{test \ test\strut}
\item test
\item test
\end{enumerate}
\end{minipage}\
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \fbox{\parbox[t]{15pt}{test \ test}}
\item \fbox{test}
\item \fbox{test}
\end{enumerate}
\end{minipage}\
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \fbox{\parbox[t]{15pt}{test \ test\strut}}
\item \fbox{test}
\item \fbox{test}
\end{enumerate}
\end{minipage}

The final two lists use \fbox to show what’s happening. The first item \parbox of the third list goes only to the bottom of its second ‘test’, which happens not have any characters that descend below the baseline. The fourth list adds the strut that gives the needed extra below-baseline space.
The \texttt{strut} command is often useful in graphics, such as in \texttt{TikZ} or \texttt{Asymptote}. For instance, you may have a command such as \texttt{\textbf{graphnode\{node-name\}}} that fits a circle around \texttt{node-name}. However, unless you are careful the \texttt{node-name}'s 'x' and 'y' will produce different-diameter circles because the characters are different sizes. A careful \texttt{\textbf{graphnode\{node-name\}}} might insert a \texttt{\textbf{strut}}, then \texttt{\textbf{node-name}}, and then draw the circle.

The general approach of using a zero width \texttt{\textbf{rule}} is useful in many circumstances. In this table, the zero-width rule keeps the top of the first integral from hitting the \texttt{\hline}. Similarly, the second rule keeps the second integral from hitting the first.

\begin{tabular}{rl}
\textsc{Integral} & \textsc{Value} \\
$\int_0^x t\, dt$ & $x^2/2$ \rule{0em}{2.5ex} \\
$\int_0^x t^2\, dt$ & $x^3/3$ \rule{0em}{2.5ex}
\end{tabular}
(Although the line-ending double backslash command has an available optional argument to change the corresponding baseline skip, that won’t solve this issue. Changing the first double backslash to something like \texttt{\[2.5ex\]} will put more room between the header line and the \texttt{\hline} rule, and the integral would still hit the rule.)

\section*{19.14 \texttt{vspace}}

Synopsis, one of:
\begin{itemize}
\item \texttt{\vspace\{\textit{length}\}}
\item \texttt{\vspace*\{\textit{length}\}}
\end{itemize}

Add the vertical space \textit{length}. The \textit{length} can be positive, negative, or zero. It is a rubber length—it may contain a plus or minus component (see Chapter 14 [Lengths], page 137).

This puts space between the two paragraphs.

And I slept.
\texttt{\vspace\{1ex plus 0.5ex\}}
The new day dawned cold.

(See Section 19.11 [\texttt{\bigskip} \& \texttt{\medskip} \& \texttt{\smallskip}], page 191, for common inter-paragraph spaces.)

The *-form inserts vertical space that is non-discardable. More precisely, \LaTeX discards vertical space at a page break and the *-form causes the space to stay. This example leaves space between the two questions.

\begin{itemize}
\item Question: Find the integral of $5x^4+5$.
\item Question: Find the derivative of $x^5+5x+9$.
\end{itemize}

That space will be present even if the page break happens to fall between the questions.

If you use \texttt{\vspace} in the middle of a paragraph (i.e., in horizontal mode) then the space is inserted after the line containing the \texttt{\vspace} command; it does not start a new paragraph at the \texttt{\vspace} command.
In this example the two questions will be evenly spaced vertically on the page, with at least one inch of space below each.

\begin{document}
1) Who put the bomp in the bomp bah bomp bah bomp?
\vspace{1in plus 1fill}

2) Who put the ram in the rama lama ding dong?
\vspace{1in plus 1fill}
\end{document}

19.15 \texttt{\vfill}

Synopsis:

\texttt{\vfill}

End the current paragraph and insert a vertical rubber length that is infinite, so it can stretch or shrink as far as needed (see Chapter 14 [Lengths], page 137).

It is often used in the same way as \texttt{\vspace{\fill}}, except that \texttt{\vfill} ends the current paragraph whereas \texttt{\vspace{\fill}} adds the infinite vertical space below its line, irrespective of the paragraph structure. In both cases that space will disappear at a page boundary; to circumvent this see the starred option in Section 19.14 \texttt{\vspace}, page 194.

In this example the page is filled, so the top and bottom lines contain the text ‘Lost Dog!’ and the second ‘Lost Dog!’ is exactly halfway between them.

\begin{document}
Lost Dog!
\vfill
Lost Dog! % perfectly in the middle
\vfill
Lost Dog!
\end{document}

19.16 \texttt{\addvspace}

Synopsis:

\texttt{\addvspace{vert-length}}

Add a vertical space of \textit{vert-length}. However, if there are two or more \texttt{\addvspace}'s in a sequence then together they only add the space needed to make the natural length equal to the maximum of the \textit{vert-length}'s in that sequence. This command is fragile (see Section 12.11 \texttt{\protect}, page 123). The \textit{vert-length} is a rubber length (see Chapter 14 [Lengths], page 137).

This example illustrates. The picture draws a scale over which to rules are placed. In a standard \LaTeX{} article the length \texttt{\baselineskip} is 12 pt. As shown by the scale, the two rules are 22 pt apart: the sum of the \texttt{\baselineskip} and the 10 pt from the first \texttt{\addvspace}.

\begin{verbatim}
\documentclass{article}
\usepackage{color}
\end{verbatim}
Now uncomment the second `\addvspace`. It does not make the gap 20 pt longer; instead the gap is the sum of `\baselineskip` and 20 pt. So `\addvspace` in a sense does the opposite of its name—it makes sure that multiple vertical spaces do not accumulate, but instead that only the largest one is used.

\LaTeX\ uses this command to adjust the vertical space above or below an environment that starts a new paragraph. For instance, a `theorem` environment begins and ends with `\addvspace` so that two consecutive `theorem`'s are separated by one vertical space, not two.

A error ‘Something’s wrong—perhaps a missing `\item` pointing to an `\addvspace` means that you were not in vertical mode when you hit this command. One way to change that is to precede `\addvspace` with a `\par` command (see Section 15.1 `\par`, page 145), as in the above example.
20 Boxes

At its core, \LaTeX{} puts things in boxes and then puts the boxes on a page. So these commands are central.

There are many packages on CTAN that are useful for manipulating boxes. One useful adjunct to the commands here is \texttt{adjustbox}.

\subsection*{20.1 \texttt{\texttt{mbox}} & \texttt{\texttt{makebox}}}

Synopsis, one of:
\begin{verbatim}
\mbox{text}
\makebox{text}
\makebox[width]{text}
\makebox[width][position]{text}
\end{verbatim}

Create a box, a container for material. The \textit{text} is typeset in LR mode (see Chapter 17 [Modes], page 177) so it is not broken into lines. The \texttt{mbox} command is robust, while \texttt{makebox} is fragile (see Section 12.11 [protect], page 123).

Because \textit{text} is not broken into lines, you can use \texttt{mbox} to prevent hyphenation. In this example, \LaTeX{} will not hyphenate the tank name, ‘T-34’.

The soviet tank \texttt{\mbox{T-34}} is a symbol of victory against nazism.

The first two command invocations shown, \texttt{mbox} and \texttt{makebox}, are roughly the same. They create a box just wide enough to contain the \textit{text}. (They are like plain \TeX{}’s \texttt{hbox}.)

In the third version the optional argument \textit{width} specifies the width of the box. Note that the space occupied by the text need not equal the width of the box. For one thing, \textit{text} can be too small; this creates a full-line box:
\begin{verbatim}
\makebox[\linewidth]{Chapter Exam}
\end{verbatim}

with ‘\textit{Chapter Exam}’ centered. But \textit{text} can also be too wide for \textit{width}. See the example below of zero-width boxes.

In the \textit{width} argument you can use the following lengths that refer to the dimension of the box that \LaTeX{} gets on typesetting \textit{text}: \texttt{depth}, \texttt{height}, \texttt{width}, \texttt{totalheight} (this is the box’s height plus its depth). For example, to make a box with the text stretched to double the natural size you can say this:
\begin{verbatim}
\makebox[2\width]{Get a stretcher}
\end{verbatim}

For the fourth command synopsis version the optional argument \textit{position} gives position of the text within the box. It may take the following values:
\begin{verbatim}
c  The \textit{text} is centered (default).
l  The \textit{text} is flush left.
r  Flush right.
s  Stretch the interword space in \textit{text} across the entire \textit{width}. The \textit{text} must contain stretchable space for this to work. For instance, this could head a press release: \texttt{\noindent\makebox[\textwidth][s]{\large IMMEDIATERELEASE\hfil}}
\end{verbatim}
A common use of \texttt{\makebox} is to make zero-width text boxes. This puts the value of the quiz questions to the left of those questions.

\begin{verbatim}
\newcommand{\pts}[1]{\makebox[0em][r]{#1 points}\hspace*{1em}}
\pts{10}What is the air-speed velocity of an unladen swallow?
\pts{90}An African or European swallow?
\end{verbatim}

The right edge of the output ‘10 points’ (note the ending space after ‘points’) will be just before the ‘What’. You can use \texttt{\makebox} similarly when making graphics, such as in \texttt{TikZ} or \texttt{Asymptote}, where you put the edge of the text at a known location, regardless of the length of that text.

For boxes with frames see Section 20.2 [\texttt{\fbox} & \texttt{\framebox}], page 198. For colors see Section 22.3.3 [Colored boxes], page 218.

There is a related version of \texttt{\makebox} that is used within the \texttt{picture} environment, where the length is given in terms of \texttt{\unitlength} (see Section 8.19.13 [\texttt{\makebox (picture)}], page 80).

As \texttt{text} is typeset in LR mode, neither a double backslash \texttt{\\} nor \texttt{\par} will give you a new line; for instance \texttt{\makebox{abc def \ ghi}} outputs ‘abc defghi’ while \texttt{\makebox{abc def \par ghi}} outputs ‘abc def ghi’, both on a single line. To get multiple lines see Section 20.3 [\texttt{\parbox}], page 199, and Section 8.18 [\texttt{\minipage}], page 71.

### 20.2 \texttt{\fbox} & \texttt{\framebox}

Synopses, one of:

\begin{verbatim}
\fbox{text}
\framebox{text}
\framebox[width]{text}
\framebox[width][position]{text}
\end{verbatim}

Create a box with an enclosing frame, four rules surrounding the text. These commands are the same as \texttt{\mbox} and \texttt{\makebox} except for the frame (see Section 20.1 [\texttt{\mbox} & \texttt{\makebox}], page 197). The \texttt{\fbox} command is robust, the \texttt{\framebox} command is fragile (see Section 12.11 [\texttt{\protect}], page 123).

\begin{verbatim}
\fbox{Warning! No work shown, no credit given.}
\end{verbatim}

\LaTeX{} puts the text into a box, the text cannot be hyphenated. Around that box, separated from it by a small gap, are four rules making a frame.

The first two command invocations, \texttt{\fbox{...}} and \texttt{\framebox{...}}, are roughly the same. As to the third and fourth invocations, the optional arguments allow you to specify the box width as \texttt{width} and the position of the text inside that box as \texttt{position}. See Section 20.1 [\texttt{\mbox} & \texttt{\makebox}], page 197, for the full description but here is an example creating an empty box that is 1/4 in wide.

\begin{verbatim}
\setlength{\fboxsep}{0pt}\framebox[0.25in]{\strut}
\end{verbatim}

The \texttt{\strut} ensures a total height of \texttt{\baselineskip} (see Section 19.13 [\texttt{\strut}], page 193).

These parameters determine the frame layout.
\fboxrule
The thickness of the rules around the enclosed box. The default is 0.2 pt. Change it with a command such as \setlength{\fboxrule}{0.8pt} (see Section 14.2 [setlength], page 139).

\fboxsep
The distance from the frame to the enclosed box. The default is 3 pt. Change it with a command such as \setlength{\fboxsep}{0pt} (see Section 14.2 [setlength], page 139). Setting it to 0 pt is useful sometimes: this will put a frame around the picture with no white border.

\begin{verbatim}
{\setlength{\fboxsep}{0pt}\
 framebox{\
 includegraphics[width=0.5\textwidth]{prudence.jpg}}}
\end{verbatim}

The extra curly braces keep the effect of the \setlength local.

As with \mbox and \makebox, \LaTeX{} will not break lines in text. But this example has \LaTeX{} break lines to make a paragraph, and then frame the result.

\begin{verbatim}
\begin{minipage}{0.6\linewidth}
 My dear, here we must run as fast as we can, just to stay in place.
 And if you wish to go anywhere you must run twice as fast as that.
\end{minipage}
\end{verbatim}

The \texttt{\setlength} local. See Section 22.3.3 [Colored boxes], page 218, for colors other than black and white.

The \texttt{picture} environment has a version of the \texttt{\framebox} command where the units depend on \texttt{picture}'s \texttt{\unitlength} (see Section 8.19.14 [\framebox (picture)], page 80).

\section*{20.3 \parbox}
Synopses, one of:
\begin{verbatim}
\parbox{width}{contents}
\parbox[position]{width}{contents}
\parbox[position]{height}{width}{contents}
\parbox[position]{height}{inner-pos}{width}{contents}
\end{verbatim}

Produce a box of text that is \texttt{width} wide. Use this command to make a box of small pieces of text, of a single paragraph. This command is fragile (see Section 12.11 [\protect], page 123).

\begin{verbatim}
\begin{picture}(0,0)
...
\put(1,2){\parbox{1.75in}{\texttt{\raggedright Because the graph is a line on this semilog paper, the relationship is exponential.}}}
\end{picture}
\end{verbatim}

The \texttt{contents} are processed in a text mode (see Chapter 17 [Modes], page 177) so \LaTeX{} will break lines to make a paragraph. But it won’t make multiple paragraphs; for that, use a \texttt{minipage} environment (see Section 8.18 [\texttt{minipage}], page 71).

The options for \texttt{\parbox} (except for \texttt{contents}) are the same as those for \texttt{minipage}. For convenience a summary of the options is here but see Section 8.18 [\texttt{minipage}], page 71, for a complete description.
There are two required arguments. The \textit{width} is a rigid length (see Chapter 14 [Lengths], page 137). It sets the width of the box into which \LaTeX typesets \textit{contents}. The \textit{contents} is the text that is placed in that box. It should not have any paragraph-making components.

There are three optional arguments, \textit{position}, \textit{height}, and \textit{inner-pos}. The \textit{position} gives the vertical alignment of the \texttt{parbox} with respect to the surrounding material. The supported values are \texttt{c} or \texttt{m} to make the vertical center of the \texttt{parbox} lines up with the center of the adjacent text line (this is the default), or \texttt{t} to match the top line of the \texttt{parbox} with the baseline of the surrounding material, or \texttt{b} to match the bottom line.

The optional argument \textit{height} overrides the natural height of the box.

The optional argument \textit{inner-pos} controls the placement of \textit{content} inside the \texttt{parbox}. Its default is the value of \textit{position}. Its possible values are: \texttt{t} to put the \textit{content} at the top of the box, \texttt{c} to put it in the vertical center, \texttt{b} to put it at the bottom of the box, and \texttt{s} to stretch it out vertically (for this, the text must contain vertically stretchable space).

\section*{20.4 \texttt{\raisebox}}

Synopsis, one of:
\begin{itemize}
\item \texttt{\raisebox{distance}{text}}
\item \texttt{\raisebox{distance}[height]{text}}
\item \texttt{\raisebox{distance}[height][depth]{text}}
\end{itemize}

Raise or lower \textit{text}. This command is fragile (see Section 12.11 \texttt{\protect}, page 123).

This example makes a command for denoting the restriction of a function by lowering the vertical bar symbol.

\begin{verbatim}
\newcommand*{\restricted}[1]{\raisebox{-.5ex}{$|$}_{#1}}
$f\restricted{A}$
\end{verbatim}

The first mandatory argument \textit{distance} specifies how far to raise the second mandatory argument \textit{text}. This is a rigid length (see Chapter 14 [Lengths], page 137). If it is negative then it lowers \textit{text}. The \textit{text} is processed in LR mode so it cannot contain line breaks (see Chapter 17 [Modes], page 177).

The optional arguments \textit{height} and \textit{depth} are dimensions. If they are specified, they override the natural height and depth of the box \LaTeX gets by typesetting \textit{text}.

In the arguments \textit{distance}, \textit{height}, and \textit{depth} you can use the following lengths that refer to the dimension of the box that \LaTeX gets on typesetting \textit{text}: \texttt{\depth}, \texttt{\height}, \texttt{\width}, \texttt{\totalheight} (this is the box’s height plus its depth).

This will align two graphics on their top (see Chapter 21 [Graphics], page 204).

\begin{verbatim}
\usepackage{graphicx,calc} % in preamble

\begin{center}
\raisebox{1ex-\height}{% 
\includegraphics[width=0.4\linewidth]{lion.png}}
\quad
\raisebox{1ex-\height}{% 
\includegraphics[width=0.4\linewidth]{meta.png}}
\end{center}
\end{verbatim}

The first \texttt{\height} is the height of \texttt{lion.png} while the second is the height of \texttt{meta.png}.
20.5 \texttt{\textbackslash sbox} & \texttt{\textbackslash savebox}

Synopsis, one of:

- \texttt{\sbox{box-cmd}\{text\}}
- \texttt{\savebox{box-cmd}\{text\}}
- \texttt{\savebox{box-cmd}[width]\{text\}}
- \texttt{\savebox{box-cmd}[width][pos]\{text\}}

Typeset text just as with \texttt{\makebox} (see Section 20.1 \[mbox \& \makebox], page 197) except that \LaTeX{} does not output it but instead saves it in a box register referred to by a variable named \texttt{box-cmd}. The variable name \texttt{box-cmd} begins with a backslash, \texttt{	extbackslash}. You must have previously allocated the box register \texttt{box-cmd} with \texttt{\newsavebox} (see Section 12.7 \[\newsavebox], page 118). The \texttt{\sbox} command is robust while \texttt{\savebox} is fragile (see Section 12.11 \[\protect], page 123).

This creates and uses a box register.

\begin{verbatim}
\newsavebox{\fullname}
\sbox{\fullname}{John Jacob Jingleheimer Schmidt}

\usebox{\fullname}! His name is my name, too!
Whenever we go out, the people always shout!
There goes \usebox{\fullname}! Ya da da da da da da.
\end{verbatim}

One advantage of using and reusing a box register over a \texttt{\newcommand} macro variable is efficiency, that \LaTeX{} need not repeatedly retypeset the contents. See the example below.

The first two command invocations shown above, \texttt{\sbox{box-cmd}\{text\}} and \texttt{\savebox{box-cmd}\{text\}}, are roughly the same. As to the third and fourth, the optional arguments allow you to specify the box width as \texttt{width}, and the position of the text inside that box as \texttt{position}. See Section 20.1 \[mbox \& \makebox], page 197, for the full description.

In the \texttt{\sbox} and \texttt{\savebox} commands the \texttt{text} is typeset in LR mode so it does not have line breaks (see Chapter 17 \[Modes], page 177). If you use these then \LaTeX{} doesn’t give you an error but it ignores what you want: if you enter \texttt{\sbox{\newreg}\{test \par test\}} and \texttt{\usebox{\newreg}} then you get ‘testtest’, while if you enter \texttt{\sbox{\newreg}\{test \par test\}} and \texttt{\usebox{\newreg}} then you get ‘test test’, but no error or warning. To fix this use a \texttt{\parbox} or \texttt{\minipage} as here.

\begin{verbatim}
\newsavebox{\areg}
\savebox{\areg}{\%}
\begin{minipage}{\linewidth}
\begin{enumerate}
\item First item
\item Second item
\end{enumerate}
\end{minipage}
\end{verbatim}

As an example of the efficiency of reusing a register’s contents, this puts the same picture on each page of the document by putting it in the header. \LaTeX{} only typesets it once.

\begin{verbatim}
\usepackage{graphicx} \% all this in the preamble
\end{verbatim}
Chapter 20: Boxes

20.6 \texttt{lrbox}

Synopsis:

```
\begin{lrbox}{box-cmd}
  text
\end{lrbox}
```

This is the environment form of the \texttt{sbox} and \texttt{savebox} commands, and is equivalent to them. See Section 20.5 [sbox \& savebox], page 201, for the full description.

The \texttt{text} inside the environment is saved in the box register referred to by variable \texttt{box-cmd}. The variable name \texttt{box-cmd} must begin with a backslash, \texttt{\}. You must allocate this box register in advance with \texttt{newsavebox} (see Section 12.7 [newsavebox], page 118). In this example the environment is convenient for entering the \texttt{tabular}:

```
\newsavebox{\jhreg}
\begin{lrbox}{\jhreg}
  \begin{tabular}{c}
    \includegraphics[height=1in]{jh.png} \\ \\
    Jim Hef{}feron
  \end{tabular}
\end{lrbox}
...
\usebox{\jhreg}
```

20.7 \texttt{usebox}

Synopsis:

```
\usebox{box-cmd}
```

Produce the box most recently saved in the box register \texttt{box-cmd} by the commands \texttt{sbox} or \texttt{savebox}, or the \texttt{lrbox} environment. For more information and examples, see

The \texttt{picture} environment is good for fine-tuning the placement.

If the register \texttt{\noreg} has not already been defined then you get something like ‘\texttt{Undefined control sequence. <argument> \noreg}’. 

...
Section 20.5 [\sbox & \savebox], page 201. (Note that the variable name \textit{box-cmd} starts with a backslash, \.) This command is robust (see Section 12.11 [\protect], page 123).
21 Graphics

You can use graphics such as PNG or PDF files in your \LaTeX{} document. You need an additional package, which comes standard with \LaTeX{}. This example is the short how-to.

\begin{verbatim}
\include{graphicx} \% goes in the preamble
...
\includegraphics[width=0.5\linewidth]{plot.pdf}
\end{verbatim}

To use the commands described here your document preamble must contain either \usepackage{graphicx} or \usepackage{graphics}. Most of the time, graphicx is the better choice.

Graphics come in two main types, raster and vector. \LaTeX{} can use both. In raster graphics the file contains an entry for each location in an array, describing what color it is. An example is a photograph in JPG format. In vector graphics, the file contains a list of instructions such as ‘draw a circle with this radius and that center’. An example is a line drawing produced by the Asymptote program, in PDF format. Generally vector graphics are more useful because you can rescale their size without pixelation or other problems, and because they often have a smaller size.

There are systems particularly well-suited to make graphics for a \LaTeX{} document. For example, these allow you to use the same fonts as in your document. \LaTeX{} comes with a picture environment (see Section 8.19 [picture], page 73) that has simple capabilities. Besides that, there are other ways to include the graphic-making commands in the document. Two such systems are the PSTricks and TikZ packages. There are also systems external to \LaTeX{}, that generate a graphic that you include using the commands of this chapter. Two that use a programming language are Asymptote and MetaPost. One that uses a graphical interface is Xfig. Full description of these systems is outside the scope of this document; see their documentation on CTAN.

21.1 graphics package options

Synopsis (must be in the document preamble):

\begin{verbatim}
\usepackage[comma-separated option list]{graphics}
\end{verbatim}

or

\begin{verbatim}
\usepackage[comma-separated option list]{graphicx}
\end{verbatim}

The graphicx package has a format for optional arguments to the \includegraphics command that is convenient (it is the key-value format), so it is the better choice for new documents. When you load the graphics or graphicx package with \usepackage there are two kinds of available options.

The first is that \LaTeX{} does not contain information about different output systems but instead depends on information stored in a \textit{printer driver} file. Normally you should not specify the driver option in the document, and instead rely on your system’s default. One advantage of this is that it makes the document portable across systems.

For completeness here is a list of the drivers. The currently relevant ones are: dvipdfmx, dvips, dvisvgm, luatex, pdftex, xetex. The two xdvi and oztex are essentially aliases for dvips (and xdvi is monochrome). Ones that should not be used for new systems are: dvipdf, dvipdfm, dviwin, dvipsone, emtex, pctxps, pctxwin, pctxxhp, pctx32,
truetex, tcidvi, vtex (and dviwindo is an alias for dvipsone). These are stored in files with a .def extension, such as pdftex.def.

The second kind of options are below.

demo Instead of an image file, \LaTeX{} puts in a 150 pt by 100 pt rectangle (unless another size is specified in the \texttt{includegraphics} command).
draft For each graphic file, it is not shown but instead its file name is printed in a box of the correct size. In order to determine the size, the file must be present.
final (Default) Override any previous draft option, so that the document shows the contents of the graphic files.
hiderotate Do not show rotated text. (This allows for the possibility that a previewer does not have the capability to rotate text.)
hidescale Do not show scaled text. (This allows for the possibility that a previewer does not have the capability to scale.)
hiresbb In a PS or EPS file the graphic size may be specified in two ways. The \texttt{%%BoundingBox} lines describe the graphic size using integer multiples of a PostScript point, that is, integer multiples of 1/72 inch. A later addition to the PostScript language allows decimal multiples, such as 1.23, in \texttt{%%HiResBoundingBox} lines. This option has \LaTeX{} to read the size from the latter.

21.2 graphics package configuration

These commands configure the way \LaTeX{} searches the file system for the graphic.

The behavior of file system search code is necessarily platform dependent. In this document we cover GNU/Linux, Macintosh, and Windows, as those systems are typically configured. For other situations consult the documentation in grfguide.pdf, or the \LaTeX{} source, or your \TeX{} distribution's documentation.

21.2.1 \texttt{\graphicspath}

Synopsis:

\texttt{\graphicspath{list of directories inside curly braces}}

Declare a list of directories to search for graphics files. This allows you to later say something like \texttt{\includegraphics{lion.png}} instead of having to give its path.

\LaTeX{} always looks for graphic files first in the current directory (and the output directory, if specified; see \[output directory\], page 260). The declaration below tells the system to then look in the subdirectory pix, and then \texttt{../pix}.

\texttt{\usepackage{graphics} \% or graphics; put in preamble}

\texttt{...}

\texttt{\graphicspath{ \{pix/\} \{../pix/\} }}

The \texttt{\graphicspath} declaration is optional. If you don’t include it then \LaTeX{}’s default is to search all of the places that it usually looks for a file (it uses \LaTeX{}’s \texttt{\input@path}). In particular, in this case one of the places it looks is the current directory.
Enclose each directory name in curly braces; for example, above it says `{pix}`. Do this even if there is only one directory. Each directory name must end in a forward slash, `/`. This is true even on Windows, where good practice is to use forward slashes for all the directory separators since it makes the document portable to other platforms. If you have spaces in your directory name then use double quotes, as with `{"my docs/"}`. Getting one of these rules wrong will cause \LaTeX to report Error: File `filename' not found.

Basically, the algorithm is that with this example, after looking in the current directory,
\begin{verbatim}
\graphicspath{ {pix/} {../pix/} }
\end{verbatim}
for each of the listed directories, \LaTeX concatenates it with the filename and searches for the result, checking for `pix/lion.png` and then `../pix/lion.png`. This algorithm means that the `\graphicspath` command does not recursively search subdirectories: if you issue `\graphicspath{a/}` and the graphic is in `a/b/lion.png` then \LaTeX will not find it. It also means that you can use absolute paths such as `\graphicspath{(/home/jim/logos/)` or `\graphicspath{(C:/Users/Albert/Pictures/)}`. However, using these means that the document is not portable. (You could preserve portability by adjusting your \TeX system settings configuration file parameter `TEXINPUTS`; see the documentation of your system.)

You can use `\graphicspath` anywhere in the document. You can use it more than once. Show its value with `\makeatletter\typeout{\Ginput@path}\makeatother`.

The directories are taken with respect to the base file. That is, suppose that you are working on a document based on `book/book.tex` and it contains `\include{chapters/chap1}`. If in `chap1.tex` you put `\graphicspath{plots/}` then \LaTeX will not search for graphics in `book/chapters/plots`, but instead in `book/plots`.

\subsection{\texttt{\textbackslash DeclareGraphicsExtensions}}

Synopses:
\begin{verbatim}
\DeclareGraphicsExtensions{comma-separated list of file extensions}
\end{verbatim}

Declare the filename extensions to try. This allows you to specify the order in which to choose graphic formats when you include graphic files by giving the filename without the extension, as in `\includegraphics{functionplot}`.

In this example, \LaTeX will find files in the PNG format before PDF files.
\begin{verbatim}
\DeclareGraphicsExtensions{.png,.PNG,.pdf,.PDF}
\end{verbatim}

\begin{verbatim}
\includegraphics{lion} % will find lion.png before lion.pdf
\end{verbatim}

Because the filename `lion` does not have a period, \LaTeX uses the extension list. For each directory in the graphics path (see Section 21.2.1 `[\graphicspath`, page 205), \LaTeX will try the extensions in the order given. If it does not find such a file after trying all the directories and extensions then it reports `! \LaTeX Error: File `lion' not found`. Note that you must include the periods at the start of the extensions.

Because GNU/Linux and Macintosh filenames are case sensitive, the list of file extensions is case sensitive on those platforms. The Windows platform is not case sensitive.

You are not required to include `\DeclareGraphicsExtensions` in your document; the printer driver has a sensible default. For example, the most recent `pdftex.def` has this extension list.
\section*{21.2.3 \texttt{\textbackslash DeclareGraphicsRule}}

Synopsis:
\begin{verbatim}
\DeclareGraphicsRule{extension}{type}{size-file extension}{command}

Declare how to handle graphic files whose names end in \texttt{extension}.

This example declares that all files with names of the form \texttt{filename-without-dot.mps} will be treated as output from MetaPost, meaning that the printer driver will use its MetaPost-handling code to input the file.
\end{verbatim}

\begin{verbatim}
\DeclareGraphicsRule{.mps}{mps}{.mps}{}
\end{verbatim}

This \begin{verbatim} \DeclareGraphicsRule{*}{mps}{*}{} \end{verbatim} tells \LaTeX{} that it should handle as MetaPost output any file with an extension not covered by another rule, so it covers \texttt{filename.1}, \texttt{filename.2}, etc.

This describes the four arguments.

\textit{extension} \hspace{1em} The file extension to which this rule applies. The extension is anything after and including the first dot in the filename. Use the Kleene star, *, to denote the default behavior for all undeclared extensions.

\textit{type} \hspace{1em} The type of file involved. This type is a string that must be defined in the printer driver. For instance, files with extensions \texttt{.ps}, \texttt{.eps}, or \texttt{.ps.gz} may all be classed as type \texttt{eps}. All files of the same type will be input with the same internal command by the printer driver. For example, the file types that \texttt{pdftex} recognizes are: \texttt{jpg}, \texttt{jb2}, \texttt{mps}, \texttt{pdf}, \texttt{png}, \texttt{tif}.

\textit{size-file extension} \hspace{1em} The extension of the file to be read to determine the size of the graphic, if there is such a file. It may be the same as \textit{extension} but it may be different.

As an example, consider a PostScript graphic. To make it smaller, it might be compressed into a \texttt{.ps.gz} file. Compressed files are not easily read by \LaTeX{} so you can put the bounding box information in a separate file. If \textit{size-file extension} is empty then you must specify size information in the arguments of \texttt{\includegraphics}.

If the driver file has a procedure for reading size files for \textit{type} then that will be used, otherwise it will use the procedure for reading \texttt{.eps} files. (Thus you may specify the size of bitmap files in a file with a PostScript style \TeX{} line if no other format is available.)

\textit{command} \hspace{1em} A command that will be applied to the file. This is often left empty. This command must start with a single backward quote. Thus, \begin{verbatim} \DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}{\texttt{gunzip -c #1}} \end{verbatim} specifies that any file with the extension \texttt{.eps.gz} should be treated as an \texttt{eps} file, with
the BoundingBox information stored in the file with extension \texttt{.eps.bb}, and that the command \texttt{gunzip -c} will run on your platform to decompresses the file.

Such a command is specific to your platform. In addition, your \LaTeX{} system must allow you to run external commands; as a security measure modern systems restrict running commands unless you explicitly allow it. See the documentation for your \LaTeX{} distribution.

### 21.3 Commands for graphics

These are the commands available with the \texttt{graphics} and \texttt{graphicx} packages.

#### 21.3.1 \texttt{\includegraphics}

**Synopses for \texttt{graphics} package:**

\begin{verbatim}
\includegraphics{filename}
\includegraphics[urx,ury]{filename}
\includegraphics[llx,lly][urx,ury]{filename}
\includegraphics*[filename]
\includegraphics*[urx,ury]{filename}
\includegraphics*[llx,lly][urx,ury]{filename}
\end{verbatim}

**Synopses for \texttt{graphicx} package:**

\begin{verbatim}
\includegraphics{filename}
\includegraphics[key-value list]{filename}
\includegraphics*[filename]
\includegraphics*[key-value list]{filename}
\end{verbatim}

Include a graphics file. The starred form \texttt{\includegraphics*} will clip the graphic to the size specified, while for the unstarred form any part of the graphic that is outside the box of the specified size will over-print the surrounding area.

This

\begin{verbatim}
\usepackage{graphicx} % in preamble
...
\begin{center}
  \includegraphics{plot.pdf}
\end{center}
\end{verbatim}

will incorporate into the document the graphic in \texttt{plot.pdf}, centered and at its nominal size. You can also give a path to the file, as with \texttt{\includegraphics[graphics/plot.pdf]}. To specify a list of locations to search for the file, see Section 21.2.1 \texttt{[graphicspath]}, page 205.

If your filename includes spaces then put it in double quotes. An example is \texttt{\includegraphics{"sister picture.jpg"}}.

The \texttt{\includegraphics[filename]} command decides on the type of graphic by splitting \texttt{filename} on the first dot. You can instead use \texttt{filename} with no dot, as in \texttt{\includegraphics[turing]}, and then \LaTeX{} tries a sequence of extensions such as \texttt{.png} and \texttt{.pdf} until it finds a file with that extension (see Section 21.2.2 \texttt{[DeclareGraphicsExtensions]}, page 206).
If your file name contains dots before the extension then you can hide them with curly braces, as in `\includegraphics{plot.2018.03.12.a}.pdf`. Or, if you use the `graphicx` package then you can use the options `type` and `ext`; see below. This and other filename issues are also handled with the package `grffile`.

This example puts a graphic in a `figure` environment so `\LaTeX` can move it to the next page if fitting it on the current page is awkward (see Section 8.10 [figure], page 60).

\begin{figure}
\centering
\includegraphics[width=3cm]{lungxray.jpg}
\caption{The evidence is overwhelming: don't smoke.} \label{fig:xray}
\end{figure}

This places a graphic that will not float, so it is sure to appear at this point in the document even if makes `\LaTeX` stretch the text or resort to blank areas on the page. It will be centered and will have a caption.

\usepackage{caption} % in preamble
...\begin{center}
\includegraphics{pix/nix.png}
\captionof{figure}{The spirit of the night} \label{pix:nix} % optional
\end{center}

This example puts a box with a graphic side by side with one having text, with the two vertically centered.

\newcommand*{\vcenteredhbox}[1]{\begin{tabular}{@{}c@{}}#1\end{tabular}}
...\begin{center}
\vcenteredhbox{\includegraphics[width=0.4\textwidth]{plot}}
\hspace{1em}
\vcenteredhbox{\begin{minipage}{0.4\textwidth}
\begin{displaymath}
f(x)=x\cdot \sin (1/x)
\end{displaymath}
\end{minipage}}
\end{center}

If you use the `graphics` package then the only options involve the size of the graphic (but see Section 21.3.2 [\rotatebox], page 213, and Section 21.3.3 [\scalebox], page 214). When one optional argument is present then it is `[urx,ury]` and it gives the coordinates of the top right corner of the image, as a pair of \TeX\, dimensions (see Section 14.1 [Units of length], page 138). If the units are omitted they default to bp. In this case, the lower left corner of the image is assumed to be at (0,0). If two optional arguments are present then the leading one is `[llx,lly]`, specifying the coordinates of the image's lower left. Thus, `\includegraphics[1in,0.618in]{...}` calls for the graphic to be placed so it is 1 inch wide and 0.618 inches tall and so its origin is at (0,0).

The `graphicx` package gives you many more options. Specify them in a key-value form, as here.

\begin{center}
The options are read left-to-right. So the first graphic above is made one inch wide and then rotated, while the second is rotated and then made one inch wide. Thus, unless the graphic is perfectly square, the two will end with different widths and heights.

There are many options. The primary ones are listed first.

Note that a graphic is placed by \LaTeX into a box, which is traditionally referred to as its bounding box (distinct from the PostScript BoundingBox described below). The graphic’s printed area may go beyond this box, or sit inside this box, but when \LaTeX makes up a page it puts together boxes and this is the box allocated for the graphic.

width The graphic will be shown so its bounding box is this width. An example is \includegraphics[width=1in]{plot}. You can use the standard \TeX dimensions (see Section 14.1 [Units of length], page 138) and also convenient is \linewidth, or in a two-column document, \columnwidth (see Section 5.5 [Page layout parameters], page 28). An example is that by using the calc package you can make the graphic be 1 cm narrower than the width of the text with \includegraphics[width=\linewidth-1.0cm]{hefferon.jpg}.

height The graphic will be shown so its bounding box is this height. You can use the standard \TeX dimensions (see Section 14.1 [Units of length], page 138), and also convenient are \pageheight and \textheight (see Section 5.5 [Page layout parameters], page 28). For instance, the command \includegraphics[height=0.25\textheight]{godel} will make the graphic a quarter of the height of the text area.

totalheight The graphic will be shown so its bounding box has this height plus depth. This differs from the height if the graphic was rotated. For instance, if it has been rotated by -90 then it will have zero height but a large depth.

keepaspectratio If set to true, or just specified as here
\includegraphics[,...,keepaspectratio,...]{...}
and you give as options both width and height (or totalheight), then \LaTeX will make the graphic as large as possible without distortion. That is, \LaTeX will ensure that neither is the graphic wider than width nor taller than height (or totalheight).

scale Factor by which to scale the graphic. To make a graphic twice its nominal size, enter \includegraphics[\scale=2.0]{...}. This number may be any value; a number between 0 and 1 will shrink the graphic and a negative number will reflect it.

angle Rotate the graphic. The angle is taken in degrees and counterclockwise. The graphic is rotated about its origin; see that option. For a complete description of how rotated material is typeset, see Section 21.3.2 [\rotatebox], page 213.
origin
The point of the graphic about which the rotation happens. Possible values are any string containing one or two of: l for left, r for right, b for bottom, c for center, t for top, and B for baseline. Thus, entering the command \includegraphics[angle=180,origin=c]{moon} will turn the picture upside down about that picture's center, while the command \includegraphics[angle=180,origin=lB]{LeBateau} will turn its picture upside down about its left baseline. (The character c gives the horizontal center in bc or tc, but gives the vertical center in lc or rc.) The default is lB.

To rotate about an arbitrary point, see Section 21.3.2 [\rotatebox], page 213.

These are lesser-used options.

viewport
Pick out a subregion of the graphic to show. Takes four arguments, separated by spaces and given in \TeX{} dimensions, as with \includegraphics[\ldots, viewport=0in 0in 1in 0.618in\ldots]{\ldots}. When the unit is omitted, the dimensions default to big points, \text{bp}. They are taken relative to the origin specified by the bounding box. See also the trim option.

trim
Gives parts of the graphic to not show. Takes four arguments, separated by spaces, that are given in \TeX{} dimensions, as with \includegraphics[\ldots, trim=0in 0.1in 0.2in 0.3in, \ldots]{\ldots}. These give the amounts of the graphic not to show, that is, \LaTeX{} will crop the picture by 0 inches on the left, 0.1 inches on the bottom, 0.2 inches on the right, and 0.3 inches on the top. See also the viewport option.

clip
If set to true, or just specified as here

\includegraphics[\ldots, clip,\ldots]{\ldots}

then the graphic is cropped to the bounding box. This is the same as using the starred form of the command, \includegraphics*[\ldots]{\ldots}.

page
Give the page number of a multi-page PDF file. The default is \text{page}=1.

pagebox
Specifies which bounding box to use for PDF files from among mediabox, cropbox, bleedbox, trimbox, or artbox. PDF files do not have the BoundingBox that PostScript files have, but may specify up to four predefined rectangles. The MediaBox gives the boundaries of the physical medium. The CropBox is the region to which the contents of the page are to be clipped when displayed. The BleedBox is the region to which the contents of the page should be clipped in production. The TrimBox is the intended dimensions of the finished page. The ArtBox is the extent of the page's meaningful content. The driver will set the image size based on CropBox if present, otherwise it will not use one of the others, with a driver-defined order of preference. MediaBox is always present.

interpolate
Enable or disable interpolation of raster images by the viewer. Can be set with interpolate=true or just specified as here.

\includegraphics[\ldots, interpolate,\ldots]{\ldots}

quiet
Do not write information to the log. You can set it with quiet=true or just specified it with \includegraphics[\ldots, quiet,\ldots]{\ldots},
draft  If you set it with `draft=true` or just specify it with
\includegraphics[,...,draft,...]{...}
then the graphic will not appear in the document, possibly saving color printer
ink. Instead, \LaTeX will put an empty box of the correct size with the filename
printed in it.

These options address the bounding box for Encapsulated PostScript graphic files, which
have a size specified with a line `%%BoundingBox` that appears in the file. It has four values,
giving the lower \(x\) coordinate, lower \(y\) coordinate, upper \(x\) coordinate, and upper \(y\) coordinate.
The units are PostScript points, equivalent to \TeX’s big points, 1/72 inch. For example,
if an .eps file has the line `%%BoundingBox 10 20 40 80` then its natural size is 30/72 inch
wide by 60/72 inch tall.

bb  Specify the bounding box of the displayed region. The argument is four dimen-
sions separated by spaces, as with \includegraphics[... , bb= 0in 0in 1in
0.618in]{...}. Usually \includegraphics reads the BoundingBox numbers
from the EPS file automatically, so this option is only useful if the bounding
box is missing from that file or if you want to change it.

bblx, bbly, bburx, bbury  Set the bounding box. These four are obsolete, but are retained for compatibility
with old packages.

natwidth, natheight  An alternative for bb. Setting
\includegraphics[... , natwidth=1in,natheight=0.618in,...]{...}
is the same as setting bb=0 0 1in 0.618in.

hiresbb  If set to `true`, or just specified as with
\includegraphics[... , hiresbb,...]{...}
then \LaTeX will look for `%%HiResBoundingBox` lines instead of `%%BoundingBox`
lines. (The `BoundingBox` lines use only natural numbers while the
HiResBoundingBox lines use decimals; both use units equivalent to \TeX’s big
points, 1/72 inch.) To override a prior setting of `true`, you can set it to `false`.

These following options allow a user to override \LaTeX’s method of choos-
ing the graphic type based on the filename extension. An example is that
\includegraphics[type=png,ext=.xyz,read=.xyz]{lion} will read the file lion.xyz
as though it were lion.png. For more on these, see Section 21.2.3 [\DeclareGraphicsRule],
page 207.

type  Specify the graphics type.

ext  Specify the graphics extension. Only use this in conjunction with the option
type.

read  Specify the file extension of the read file. Only use this in conjunction with the
option type.

command  Specify a command to be applied to this file. Only use this in conjunction with
the option type. See Section 28.1 [Command line options], page 259, for a
discussion of enabling the `\write18` functionality to run external commands.
21.3.2 \rotatebox

Synopsis if you use the \texttt{graphics} package:

\rotatebox{angle}{material}

Synopses if you use the \texttt{graphicx} package:

\rotatebox{angle}{material}
\rotatebox[\texttt{key-value list}]{angle}{material}

Put \texttt{material} in a box and rotate it \texttt{angle} degrees counterclockwise.

This example rotates the table column heads forty-five degrees.

\begin{tabular}{ll}
\rotatebox{45}{Character} & \rotatebox{45}{NATO phonetic} \\
A & AL-FAH \\
B & BRAH-VOH
\end{tabular}

The \texttt{material} can be anything that goes in a box, including a graphic.

\rotatebox[\texttt{origin=c}]{45}{\includegraphics[width=1in]{lion}}

To place the rotated material, the first step is that \LaTeX{} sets \texttt{material} in a box, with a reference point on the left baseline. The second step is the rotation, by default about the reference point. The third step is that \LaTeX{} computes a box to bound the rotated material. Fourth, \LaTeX{} moves this box horizontally so that the left edge of this new bounding box coincides with the left edge of the box from the first step (they need not coincide vertically). This new bounding box, in its new position, is what \LaTeX{} uses as the box when typesetting this material.

If you use the \texttt{graphics} package then the rotation is about the reference point of the box. If you use the \texttt{graphicx} package then these are the options that can go in the \texttt{key-value list}, but note that you can get the same effect without needing this package, except for the \texttt{x} and \texttt{y} options (see Section 21.3.1 \texttt{includegraphics}, page 208).

**origin**  

The point of the \texttt{material}'s box about which the rotation happens. Possible value is any string containing one or two of: \texttt{l} for left, \texttt{r} for right, \texttt{b} for bottom, \texttt{c} for center, \texttt{t} for top, and \texttt{B} for baseline. Thus, the first line here

\rotatebox[\texttt{origin=c}]{180}{moon}
\rotatebox[\texttt{origin=1B}]{180}{LeBateau}

will turn the picture upside down from the center while the second will turn its picture upside down about its left baseline. (The character \texttt{c} gives the horizontal center in \texttt{bc} or \texttt{tc} but gives the vertical center in \texttt{lc} or \texttt{rc}, and gives both in \texttt{c}.) The default is \texttt{1B}.

**x, y**  

Specify an arbitrary point of rotation with \rotatebox[\texttt{x=LaTeX\ width},\texttt{y=LaTeX\ dimension}]{...} (see Section 14.1 [Units of length], page 138). These give the offset from the box's reference point.

**units**  

This key allows you to change the default of degrees counterclockwise. Setting \texttt{units=-360} changes the direction to degrees clockwise and setting \texttt{units=6.283185} changes to radians counterclockwise.
21.3.3 \scalebox

Synopses:
\scalebox{horizontal factor}{material}
\scalebox{horizontal factor}[vertical factor]{material}
\reflectbox{material}

Scale the material.
This example halves the size, both horizontally and vertically, of the first text and doubles the size of the second.
\scalebox{0.5}{DRINK ME} and \scalebox{2.0}{Eat Me}

If you do not specify the optional vertical factor then it defaults to the same value as the horizontal factor.

You can use this command to resize a graphic, as here.
\scalebox{0.5}{\includegraphics{lion}}

If you use the graphicx package then you can accomplish the same thing with optional arguments to \includegraphics (see Section 21.3.1 [includegraphics], page 208).

The \reflectbox command abbreviates \scalebox{-1}{1}{material}. Thus, Able was I\reflectbox{Able was I} will show the phrase ‘Able was I’ immediately followed by its mirror reflection against a vertical axis.

21.3.4 \resizebox

Synopses:
\resizebox{horizontal length}{vertical length}{material}
\resizebox*{horizontal length}{vertical length}{material}

Given a size, such as 3cm, transform material to make it that size. If either horizontal length or vertical length is an exclamation point ! then the other argument is used to determine a scale factor for both directions.

This example makes the graphic be a half inch wide and scales it vertically by the same factor to keep it from being distorted.
\resizebox{0.5in}{!}{\includegraphics{lion}}

The unstarred form \resizebox takes vertical length to be the box’s height while the starred form \resizebox* takes it to be height+depth. For instance, make the text have a height+depth of a quarter-inch with \resizebox*{!}{0.25in}{\parbox{3.5in}{This box has both height and depth.}}.

You can use \depth, \height, \totalheight, and \width to refer to the original size of the box. Thus, make the text two inches wide but keep the original height with \resizebox{2in}{\height}{Two inches}. 
22 Color

You can add color to text, rules, etc. You can also have color in a box or on an entire page and write text on top of it.

Color support comes as an additional package. So put \usepackage{color} in your document preamble to use the commands described here.

Many other packages also supplement \LaTeX's color abilities. Particularly worth mentioning is xcolor, which is widely used and significantly extends the capabilities described here, including adding ‘HTML’ and ‘Hsb’ color models.

22.1 color package options

Synopsis (must be in the document preamble):

\usepackage[comma-separated option list]{color}

When you load the color package there are two kinds of available options.

The first specifies the printer driver. \LaTeX doesn’t contain information about different output systems but instead depends on information stored in a file. Normally you should not specify the driver option in the document, and instead rely on your system’s default. One advantage of this is that it makes the document portable across systems. For completeness we include a list of the drivers. The currently relevant ones are: dvipdfmx, dvips, dvisvgm, luartex, pdftex, xetex. The two xdvi and oztex are essentially aliases for dvips (and xdvi is monochrome). Ones that should not be used for new systems are: dvipdf, dvipdfm, dviwin, dvipsone, emtex, pctxps, pctxwin, pctxhp, pctx32, truetex, tcidvi, vtex (and dviwindo is an alias for dvipsone).

The second kind of options, beyond the drivers, are below.

monochrome

Disable the color commands, so that they do not generate errors but do not generate color either.

dvipsnames

Make available a list of 68 color names that are often used, particularly in legacy documents. These color names were originally provided by the dvips driver, giving the option name.

nodvipsnames

Do not load that list of color names, saving \LaTeX a tiny amount of memory space.

22.2 Color models

A color model is a way of representing colors. \LaTeX’s capabilities depend on the printer driver. However, the pdftex, xetex, and luartex printer drivers are today by far the most commonly used. The models below work for those drivers. All but one of these is also supported by essentially all other printer drivers used today.

Note that color combination can be additive or subtractive. Additive mixes colors of light, so that for instance combining full intensities of red, green, and blue produces white.
Subtractive mixes pigments, such as with inks, so that combining full intensity of cyan, magenta, and yellow makes black.

cmyk A comma-separated list with four real numbers between 0 and 1, inclusive. The first number is the intensity of cyan, the second is magenta, and the others are yellow and black. A number value of 0 means minimal intensity, while a 1 is for full intensity. This model is often used in color printing. It is a subtractive model.

gray A single real number between 0 and 1, inclusive. The colors are shades of grey. The number 0 produces black while 1 gives white.

rgb A comma-separated list with three real numbers between 0 and 1, inclusive. The first number is the intensity of the red component, the second is green, and the third the blue. A number value of 0 means that none of that component is added in, while a 1 means full intensity. This is an additive model.

RGB (pdftex, xetex, luatex drivers) A comma-separated list with three integers between 0 and 255, inclusive. This model is a convenience for using rgb since outside of \LaTeX colors are often described in a red-green-blue model using numbers in this range. The values entered here are converted to the rgb model by dividing by 255.

named Colors are accessed by name, such as ‘PrussianBlue’. The list of names depends on the driver, but all support the names ‘black’, ‘blue’, ‘cyan’, ‘green’, ‘magenta’, ‘red’, ‘white’, and ‘yellow’ (See the dvipsnames option in Section 22.1 [Color package options], page 215).

### 22.3 Commands for color

These are the commands available with the color package.

#### 22.3.1 Define colors

Synopsis:

```
\definecolor{name}{model}{specification}
```

Give the name *name* to the color. For example, after this

```
\definecolor{silver}{rgb}{0.75,0.75,0.74}
```

you can use that color name with `Hi ho, \textcolor{silver}{Silver}!`.

This example gives the color a more abstract name, so it could change and not be misleading.

```
\definecolor{logocolor}{RGB}{145,92,131} \% RGB needs pdflatex
\newcommand{\logo}{\textcolor{logocolor}{Bob’s Big Bagels}}
```

Often a document’s colors are defined in the preamble, or in the class or style, rather than in the document body.

#### 22.3.2 Colored text

Synopsis:

```
\textcolor{name}{...}
```
\textcolor[color model]{color specification}{...}

or
\color{name}
\color[color model]{color specification}

The affected text gets the color. This line
\textcolor{magenta}{My name is Ozymandias, King of Kings;}
Look on my works, ye Mighty, and despair!

causes the first half to be in magenta while the rest is in black. You can use a color declared with \definecolor in exactly the same way that we just used the builtin color 'magenta'.
\definecolor{MidlifeCrisisRed}{rgb}{1.0,0.11,0.0}
I'm thinking about getting a \textcolor{MidlifeCrisisRed}{sports car}.

The two \textcolor and \color differ in that the first is a command form, enclosing the text to be colored as an argument. Often this form is more convenient, or at least more explicit. The second form is a declaration, as in The moon is made of \color{green}{green} cheese, so it is in effect until the end of the current group or environment. This is sometimes useful when writing macros or as below where it colors everything inside the \center environment, including the vertical and horizontal lines.
\begin{center} \color{blue}
\begin{tabular}{l|r}
UL & UR \\
\hline
LL & LR
\end{tabular}
\end{center}

You can use color in equations. A document might have this definition in the preamble
\definecolor{highlightcolor}{RGB}{225,15,0}
and then contain this equation.
\begin{equation}
\int_a^b \textcolor{highlightcolor}{f'(x)}\,dx=f(b)-f(a)
\end{equation}

Typically the colors used in a document are declared in a class or style but sometimes you want a one-off. Those are the second forms in the synopses.

Colors of \textcolor[rgb]{0.33,0.14,0.47}{Purple} and \color[rgb]{0.72,0.60,0.37}{Gold} for the team.

The format of \textcolor{gray}{grayed out}. Colors inside colors do not combine. Thus
\textcolor{green}{kind of \textcolor{blue}{blue}} has a final word that is blue, not a combination of blue and green.
22.3.3 Colored boxes

Synopses:

\colorbox{name}{...}
\colorbox[model name]{box background color}{...}

or

\fcolorbox[frame color]{box background color}{...}
\fcolorbox[model name][frame color]{box background color}{...}

Make a box with the stated background color. The \fcolorbox command puts a frame around the box. For instance this

Name:\colorbox{cyan}{\makebox[5cm][l]{\strut}}

makes a cyan-colored box that is five centimeters long and gets its depth and height from the \strut (so the depth is \text{-0.3\baselineskip} and the height is \text{\baselineskip}). This puts white text on a blue background.

\colorbox{blue}{\textcolor{white}{Welcome to the machine.}}

The \fcolorbox commands use the same parameters as \fbox (see Section 20.2 [fbox & framebox], page 198), \fboxrule and \fboxsep, to set the thickness of the rule and the boundary between the box interior and the surrounding rule. \LaTeX\’s defaults are 0.4pt and 3pt, respectively.

This example changes the thickness of the border to 0.8 points. Note that it is surrounded by curly braces so that the change ends at the end of the second line.

{\setlength{\fboxrule}{0.8pt}}
\fcolorbox{black}{red}{Under no circumstances turn this knob.}

22.3.4 Colored pages

Synopses:

\pagecolor{name}
\pagecolor[color model]{color specification}
\nopagecolor

The first two set the background of the page, and all subsequent pages, to the color. For an explanation of the specification in the second form see Section 22.3.2 [Colored text], page 216. The third returns the background to normal, which is a transparent background. (If that is not supported use \pagecolor{white}, although that will make a white background rather than the default transparent background.)

...\pagecolor{cyan}...
\nopagecolor
23 Special insertions

LaTeX provides commands for inserting characters that have a special meaning do not correspond to simple characters you can type.

23.1 Printing special characters

LaTeX sets aside a few characters for special purposes; they are called reserved characters or special characters. Here they are:

\begin{verbatim}
# $ % & { } _ ~ ^ \
\end{verbatim}

The meaning of all the special characters is given elsewhere in this manual (see [Reserved characters], page 6).

If you want a reserved character to be printed as itself, in the text body font, for all but the final three characters in that list simply put a \ in front of the character. Thus, typing \$1.23 will produce $1.23 in your output.

As to the last three characters, to get a tilde in the text body font use $\{\}$ (omitting the curly braces would result in the next character receiving a tilde accent). Similarly, to get a text body font circumflex use \textasciicircum{}\{\}. To get a backslash in the font of the text body, enter \textbackslash{}.

To produce the reserved characters in a typewriter font, use \verb!! as below (the \newline in the example is there only to split the lines in the output).

\begin{verbatim}
\begin{center}
\# $ \% \& \{ \} \_ ~ ^ \\textbackslash \newline
\verb!# $ % & { } _ ~ ^ \!
\end{center}
\end{verbatim}

23.2 Upper and lower case

Synopsis:

\begin{verbatim}
\uppercase{text}
\lowercase{text}
\MakeUppercase{text}
\MakeLowercase{text}
\end{verbatim}

Change the case of characters. The \TeX primitive commands \uppercase and \lowercase are set up by default to work only with the 26 letters a–z and A–Z. The \LaTeX commands \MakeUppercase and \MakeLowercase commands also change characters accessed by commands such as \ae or \aa. The commands \MakeUppercase and \MakeLowercase are robust but they have moving arguments (see Section 12.11 \protect, page 123).

These commands do not change the case of letters used in the name of a command within text. But they do change the case of every other Latin letter inside the argument text. Thus, \MakeUppercase{Let $y=f(x)$} produces ‘LET $Y=F(X)$’. Another example is that the name of an environment will be changed, so that \MakeUppercase{\begin{tabular} ... \end{tabular}} will produce an error because the first half is changed to \begin{tabular}. \end{tabular}
\textTeX uses the same fixed table for changing case throughout a document. The table used is designed for the font encoding T1; this works well with the standard \TeX fonts for all Latin alphabets but will cause problems when using other alphabets.

To change the case of text that results from a macro inside \text you need to do expansion. Here the \texttt\{\texttt\Schoolname\} produces ‘COLLEGE OF MATHEMATICS’.

\begin{verbatim}
\newcommand{\schoolname}{College of Mathematics}
\newcommand{\Schoolname}{\expandafter\MakeUppercase
\expandafter{\schoolname}}
\end{verbatim}

The \texttt\{\texttt\textcase\} package brings some of the missing feature of the standard \textTeX commands \texttt\{\texttt\MakeUppercase\} and \texttt\{\texttt\MakeLowerCase\}.

To uppercase only the first letter of words, you can use the package \texttt\{\texttt\mfirstuc\}.

Handling all the casing rules specified by Unicode, e.g., for non-Latin scripts, is a much bigger job than anything envisioned in the original \TeX and \textTeX. It has been implemented in the \texttt\{\texttt\exp13\} package as of 2020. The article “Case changing: From \TeX primitives to the Unicode algorithm”, (Joseph Wright, \textit{TUGboat} 41:1, \url{https://tug.org/TUGboat/tb41-1/tb127wright-case.pdf}), gives a good overview of the topic, past and present.

### 23.3 Symbols by font position

You can access any character of the current font using its number with the \texttt\{\texttt\symbol\} command. For example, the visible space character used in the \texttt\{\texttt\verb*\} command has the code decimal 32 in the standard Computer Modern typewriter font, so it can be typed as \texttt\{\symbol{32}\}.

You can also specify numbers in octal (base 8) by using a ' prefix, or hexadecimal (base 16) with a " prefix, so the visible space at 32 decimal could also be written as \texttt\{\symbol{'40}\} or \texttt\{\symbol{"20}\}.

### 23.4 Text symbols

\textTeX provides commands to generate a number of non-letter symbols in running text. Some of these, especially the more obscure ones, are not available in OT1. As of the \textTeX February 2020 release, all symbols are available by default; before that, it was necessary to use the \texttt\{\texttt\textcomp\} package for some (technically, those in the TS1 font encoding).

\begin{verbatim}
\copyright
\textcopyright
\dag
\ddag
\LaTeX
\LaTeXe
\end{verbatim}

\textcopyright The copyright symbol.

\dag The dagger symbol (in text).

\ddag The double dagger symbol (in text).

\LaTeX The \textTeX logo.

\LaTeXe The \textTeX2e logo.
\guillemetleft («)  
\guillemetright (»)  
\guillemotleft («)  
\guillemotright (»)  
\guilsinglleft (‹)  
\guilsinglright (›)  

«, », <, >  Double and single angle quotation marks, commonly used in French. The commands \guillemotleft and \guillemotright are synonyms for \guillemet...; these are misspellings inherited from Adobe. (Guillemots are seabirds; guillemets are French quotes.)

\ldots
\textellipsis  
\ldots and \dots also work in math mode (see Section 16.2.6 [Dots], page 166). See that math mode ellipsis description for additional general information.

\lq  ' Left (opening) quote.

\p  
\textparagraph  ¶  Paragraph sign (pilcrow).

\pounds  £  English pounds sterling.

\quotedblbase („)  
\quotesinglbase (‘)  
„ and ‘, Double and single quotation marks on the baseline.

\rq  ’ Right (closing) quote.

\S  §  Section sign.

\TeX  The \TeX logo.

\textasciicircum  ^  ASCII circumflex.

\textasciitilde  ~  ASCII tilde.

\textasteriskcentered  *  Centered asterisk.

\textbackslash  
\ Backslash. However, \texttt{\textbackslash} produces a roman (not typewriter) backslash by default; for a typewriter backslash, it is necessary to use the T1 (or other non-default) font encoding, as in:

\usepackage[T1]{fontenc}
\textbar | Vertical bar.
\textbardbl || Double vertical bar.
\textbigcircle Big circle symbol.
\textbraceleft 
\{ Left brace. See remarks at \textbackslash above about making \texttt{\textbraceleft} produce a typewriter brace.
\textbraceright 
\} Right brace. See remarks at \textbackslash above about making \texttt{\textbraceright} produce a typewriter brace.
\textbullet • Bullet.
\textcircled{letter} Circle around letter.
\textcompwordmark
\textcapitalcompwordmark
\textascendercompwordmark Used to separate letters that would normally ligature. For example, f\textcompwordmark i produces ‘fi’ without a ligature. This is most useful in non-English languages. The \textcapitalcompwordmark form has the cap height of the font while the \textascendercompwordmark form has the ascender height.
\textdagger † Dagger.
\textdaggerdbl ‡ Double dagger.
\textdollar (or \$) $ Dollar sign.
\textemdash (or ---) — Em-dash. Used for punctuation, usually similar to commas or parentheses, as in ‘The playoffs---if you’re lucky enough to make the playoffs---are more like a sprint.’ Conventions for spacing around em-dashes vary widely.
\textendash (or --) – En-dash. Used for ranges, as in ‘see pages 12--14’.
\texteuro The Euro currency symbol: €. For an alternative glyph design, try the eurosym package; also, most fonts nowadays come with their own Euro symbol (Unicode U+20AC).
\textexclamdown (or `!`)  
¡ Upside down exclamation point.

\textfiguredash  
Dash used between numerals, Unicode U+2012. Defined in the June 2021 release of \LaTeX. When used in \pdftex, approximated by an en-dash; with a Unicode engine, either typesets the glyph if available in the current font, or writes the usual “Missing character” warning to the log file.

\textgreater  
> Greater than symbol.

\texthorizontalbar  
Horizontal bar character, Unicode U+2015. Defined in the June 2021 release of \LaTeX. Behavior as with \textfiguredash above; the \pdftex approximation is an em-dash.

\textless  
< Less than symbol.

\textleftarrow  
Left arrow.

\textnonbreakinghyphen  
Non-breaking hyphen character, Unicode U+2011. Defined in the June 2021 release of \LaTeX. Behavior as with \textfiguredash above; the \pdftex approximation is a regular ASCII hyphen (with breaks disallowed after).

\textordfeminine  
\textordmasculine  
a, o Feminine and masculine ordinal symbols.

\textperiodcentered  
. Centered period.

\textquestiondown (or `?`)  
¿ Upside down question mark.

\textquotedblleft (or ```)  
“ Double left quote.

\textquotedblright (or `'')  
” Double right quote.

\textquoteleft (or ```)  
‘ Single left quote.

\textquottoright (or `')  
’ Single right quote.

\textquotesingle  
Straight single quote. (From TS1 encoding.)

\textquotestraightbase  
\textquotestraightdblbase  
Single and double straight quotes on the baseline.
\textregistered
\textregistered Registered symbol.

\textrightarrow
\textrightarrow Right arrow.

\textthreequartersemidash
“Three-quarters” em-dash, between en-dash and em-dash.

\texttrademark
\textnormal{TM} Trademark symbol.

\texttwelveudash
“Two-thirds” em-dash, between en-dash and em-dash.

\textunderscore
Underscore.

\textvisiblespace
Visible space symbol.

23.5 Accents
\LaTeX{} has wide support for many of the world’s scripts and languages, provided through the core \texttt{babel} package, which supports pdf\LaTeX{}, Xe\LaTeX{} and Lua\LaTeX{}. The \texttt{polyglossia} package provides similar support with the latter two engines.

This section does not cover that support. It only lists the core \LaTeX{} commands for creating accented characters. The \texttt{\textbackslash{capital}...} commands shown here produce alternative forms for use with capital letters. These are not available with OT1.

Below, to make them easier to find, the accents are all illustrated with lowercase ‘o’.

Note that \texttt{\textbackslash{}i} produces a dotless i, and \texttt{\textbackslash{}j} produces a dotless j. These are often used in place of their dotted counterparts when they are accented.

\texttt{\textbackslash{"{}}}
\texttt{\textbackslash{capitaldieresis}}
ö Umlaut (dieresis).

\texttt{\textbackslash{'}}
\texttt{\textbackslash{capitalacute}}
ó Acute accent.

\texttt{\textbackslash{.}}
\texttt{\textbackslash{dot}}
ó Dot accent.

\texttt{\textbackslash{=}}
\texttt{\textbackslash{capitalmacron}}
ö Macron (overbar) accent.

\texttt{\textbackslash{\textasciitilde}}
\texttt{\textbackslash{capitalcircumflex}}
ó Circumflex (hat) accent.

\texttt{\textbackslash{\textasciitilde}}
\texttt{\textbackslash{capitalgrave}}
ö Grave accent.
\~
\texttt{\textbackslash capitaltilde}
\begin{verbatim}
\textbackslash capitaltilde
\end{verbatim}
\nñ Tilde accent.

\b
\begin{verbatim}
\textbackslash b
\end{verbatim}
\nø Bar accent underneath.

Related to this, \texttt{\textbackslash underbar{text}} produces a bar under \textit{text}. The argument is always processed in LR mode (see Chapter 17 [Modes], page 177). The bar is always a fixed position under the baseline, thus crossing through descenders. See also \texttt{\textbackslash underline} in [Over- and Underlining], page 170.

\c
\begin{verbatim}
\textbackslash capitalcedilla
\end{verbatim}
\nč Cedilla accent underneath.

\d
\begin{verbatim}
\textbackslash capitaldotaccent
\end{verbatim}
\nø Dot accent underneath.

\H
\begin{verbatim}
\textbackslash capitalhungarumlaut
\end{verbatim}
\nöz Long Hungarian umlaut accent.

\k
\begin{verbatim}
\textbackslash capitalogonek
\end{verbatim}
\nø Ogonek. Not available in the OT1 encoding.

\r
\begin{verbatim}
\textbackslash capitalring
\end{verbatim}
\nő Ring accent.

\t
\begin{verbatim}
\textbackslash capitaltie
\texttt{\textbackslash newtie}
\texttt{\textbackslash capitalnewtie}
\end{verbatim}
\nőő Tie-after accent (used for transliterating from Cyrillic, such as in the ALA-LC romanization). It expects that the argument has two characters. The \texttt{\textbackslash newtie} form is centered in its box.

\u
\begin{verbatim}
\textbackslash capitalbreve
\end{verbatim}
\nő Breve accent.

\v
\begin{verbatim}
\textbackslash capitalcaron
\end{verbatim}
\nő Háček (check, caron) accent.

\textbf{23.5.1 \textbackslash accent}

Synopsis:
\begin{verbatim}
\texttt{\textbackslash accent number character}
\end{verbatim}

A \TeX{} primitive command used to generate accented characters from accent marks and letters. The accent mark is selected by \textit{number}, a numeric argument, followed by a space and then a \textit{character} argument to construct the accented character in the current font.
These are accented ‘e’ characters.
\accent18 e
\accent20 e
\accent21 e
\accent22 e
\accent23 e

The first is a grave, the second a caron, the third a breve, the fourth a macron, and the fifth a ring above.

The position of the accent is determined by the font designer and so the outcome of \accent use may differ between fonts. In \LaTeX it is desirable to have glyphs for accented characters rather than building them using \accent. Using glyphs that already contain the accented characters (as in T1 encoding) allows correct hyphenation whereas \accent disables hyphenation (specifically with OT1 font encoding where accented glyphs are absent).

There can be an optional font change between number and character. Note also that this command sets the \spacefactor to 1000 (see Section 19.5 [\spacefactor], page 186).

An unavoidable characteristic of some Cyrillic letters and the majority of accented Cyrillic letters is that they must be assembled from multiple elements (accents, modifiers, etc.) while \accent provides for a single accent mark and a single letter combination. There are also cases where accents must appear between letters that \accent does not support. Still other cases exist where the letters I and J have dots above their lowercase counterparts that conflict with dotted accent marks. The use of \accent in these cases will not work as it cannot analyze upper/lower case.

### 23.6 Additional Latin letters

Here are the basic \LaTeX commands for inserting letters beyond A–Z that extend the Latin alphabet, used primarily in languages other than English.

\aa
\AA â and Â.

\ae
\AE æ and Æ.

\dh
\DH Icelandic letter eth: ð and Ð. Not available with OT1 encoding, you need the fontenc package to select an alternate font encoding, such as T1.

\dj
\DJ Crossed d and D, a.k.a. capital and small letter d with stroke. Not available with OT1 encoding, you need the fontenc package to select an alternate font encoding, such as T1.

\i j
\IJ ij and IJ (except somewhat closer together than appears here).

\l
\L l and L.

\ng
\NG Lappish letter eng, also used in phonetics.
\o  \O  \ø  \Ø  \oe  \OE  \œ  \Œ  \ss  \SS  \ß  \SS.

\th  \TH  Icelandic letter thorn: þ and Þ. Not available with OT1 encoding, you need the fontenc package to select an alternate font encoding, such as T1.

23.7 inputenc package

Synopsis:
\usepackage[encoding-name]{inputenc}

Declare the input file's text encoding to be encoding-name. (For basic background, see Section 2.4.1 [Input encodings], page 5). The default, if this package is not loaded, is UTF-8. Technically, specifying the encoding name is optional, but in practice it is not useful to omit it.

The inputenc package tells \LaTeX what encoding is used. For instance, the following command explicitly says that the input file is UTF-8 (note the lack of a dash).
\usepackage[utf8]{inputenc}

The most common values for encoding-name are: ascii, latin1, latin2, latin3, latin4, latin5, latin9, latin10, utf8.

Caution: use inputenc only with the pdf\TeX engine (see Section 2.3 [\TeX engines], page 3); with xelatex or lualatex, declaring a non-UTF-8 encoding with inputenc, such as latin1, will get the error inputenc is not designed for xetex or luatex.

An inputenc package error such as \texttt{Invalid UTF-8 byte "96 means that some of the material in the input file does not follow the encoding scheme. Often these errors come from copying material from a document that uses a different encoding than the input file. The simplest solution is often to replace the non-UTF-8 character with a UTF-8 or \LaTeX equivalent.

If you need to process a non-UTF-8 document with Lua\TeX, you can use the luainputenc package (https://ctan.org/pkg/luainputenc). With Xe\TeX, the \texttt{\XeTeXinputencoding} and \texttt{\XeTeXdefaultencoding} primitives can be used (for an explanation and examples, see https://tex.stackexchange.com/questions/324948).

It’s also possible to re-encode a document from an 8-bit encoding to UTF-8 outside of \TeX, using system utilities. For example, \texttt{recode latin1..utf8} or \texttt{iconv -f latin1 -t utf8}.

In a few documents, such as a collection of journal articles from a variety of authors, changing the encoding in mid-document may be necessary. You can use the command \inputencoding{encoding-name} for this.
23.8 \rule

Synopsis, one of:
\[\text{\rule\{width\}\{thickness\}}\]
\[\text{\rule\{width\}\{thickness\}}\]
This produces a rule, a filled-in rectangle.

This example produces a rectangular blob, sometimes called a Halmos symbol, or just “qed”, often used to mark the end of a proof:
\[\text{\newcommand\{\qedsymbol\}\{\rule\{0.4em\}\{2ex\}}\]
The \texttt{amsthm} package includes this command, with a somewhat different-looking symbol.

The \texttt{amsthm} package includes this command, with a somewhat different-looking symbol.

The mandatory arguments give the horizontal \textit{width} and vertical \textit{thickness} of the rectangle. They are rigid lengths (see Chapter 14 [Lengths], page 137). The optional argument \textit{raise} is also a rigid length, and tells \LaTeX{} how much to raise the rule above the baseline, or lower it if the length is negative.

This produces a line, a rectangle that is wide but not tall.
\[\text{\noindent\rule\{\textwidth\}\{0.4pt\}}\]
The line is the width of the page and 0.4 points tall. This line thickness is common in \LaTeX{}.

A rule that has zero width, or zero thickness, will not show up in the output, but can cause \LaTeX{} to change the output around it. See Section 19.13 [\strut], page 193, for examples.

23.9 \today

Synopsis:
\[\text{\today}\]
This produces today’s date in the format ‘\textit{month dd, yyyy’}. An example of a date in that format is ‘July 4, 1976’.

Multilingual packages such as \texttt{babel} or \texttt{polyglossia}, or classes such as \texttt{lettre}, will localize \today. For example, the following will output ‘4 juillet 1976’:
\texttt{\year=1976 \month=7 \day=4}
\texttt{\documentclass\{minimal\}}
\texttt{\usepackage\{french\}}
\texttt{\begin\{document\}}
\texttt{\today}
\texttt{\end\{document\}}

\today uses the counters \texttt{\day}, \texttt{\month}, and \texttt{\year} (see Section 13.8 [\texttt{\day \& \month \& \year}], page 136).

A number of package on CTAN work with dates. One is \texttt{datetime} package which can produce a wide variety of date formats, including ISO standards.

The date is not updated as the \LaTeX{} process runs, so in principle the date could be incorrect by the time the program finishes.
24 Splitting the input

\LaTeX{} lets you split a large document into several smaller ones. This can simplify editing or allow multiple authors to work on the document. It can also speed processing.

Regardless of how many separate files you use, there is always one root file, on which \LaTeX{} compilation starts. This shows such a file with five included files.

\begin{verbatim}
\documentclass{book}
\includeonly{ % comment out lines below to omit compiling
  pref,
  chap1,
  chap2,
  append,
  bib
}
\begin{document}
\frontmatter
\include{pref}
\mainmatter
\include{chap1}
\include{chap2}
\appendix
\include{append}
\backmatter
\include{bib}
\end{document}
\end{verbatim}

This will bring in material from pref.tex, chap1.tex, chap2.tex, append.tex, and bib.tex. If you compile this file, and then comment out all of the lines inside \includeonly{...} except for chap1, and compile again, then \LaTeX{} will only process the material in the first chapter. Thus, your output will appear more quickly and be shorter to print. However, the advantage of the \includeonly{} command is that \LaTeX{} will retain the page numbers and all of the cross reference information from the other parts of the document so these will appear in your output correctly.

See Section A.4 [Larger book template], page 265, for another example of \includeonly.

24.1 \endinput

Synopsis:

\begin{verbatim}
\endinput
\end{verbatim}

When you \include{filename}, inside filename.tex the material after \endinput will not be included. This command is optional; if filename.tex has no \endinput then \LaTeX{} will read all of the file.

For example, suppose that a document’s root file has \input{chap1} and this is chap1.tex.

\begin{verbatim}
\chapter{One}
This material will appear in the document.
\end{verbatim}
This will not appear.

This can be useful for putting documentation or comments at the end of a file, or for avoiding junk characters that can be added if the file is transmitted in the body of an email. It is also useful for debugging: one strategy to localize errors is to put \endinput halfway through the included file and see if the error disappears. Now, knowing which half contains the error, moving \endinput to halfway through that area further narrows down the location. This process rapidly finds the offending line.

After reading \endinput, \LaTeX{} continues to read to the end of the line, so something can follow this command and be read nonetheless. This allows you, for instance, to close an \if... with a \fi.

### 24.2 \include & \includeonly

**Synopsis:**

\begin{verbatim}
\includeonly{ % in document preamble
    ... 
    filename, 
    ... 
}\include{filename} % in document body
\end{verbatim}

Bring material from the external file filename.tex into a \LaTeX{} document.

The \include command does three things: it executes \clearpage (see Section 10.1 [\clearpage & \cleardoublepage], page 104), then it inputs the material from filename.tex into the document, and then it does another \clearpage. This command can only appear in the document body.

The \includeonly command controls which files will be read by \LaTeX{} under subsequent \include commands. Its list of filenames is comma-separated. It must appear in the preamble or even earlier, e.g., the command line; it can’t appear in the document body.

This example root document, constitution.tex, brings in three files, preamble.tex, articles.tex, and amendments.tex.

```
\documentclass{book}
\includeonly{ 
    preamble, 
    articles, 
    amendments 
}\begin{document} 
\include{preamble} 
\include{articles} 
\include{amendments} 
\end{document}
```

The file preamble.tex contains no special code; you have just excerpted the chapter from constitution.tex and put it in a separate file just for editing convenience.
\chapter{Preamble}

We the People of the United States,
in Order to form a more perfect Union, ...

Running \LaTeX{} on \texttt{constitution.tex} makes the material from the three files appear in
the document but also generates the auxiliary files \texttt{preamble.aux}, \texttt{articles.aux}, and
\texttt{amendments.aux}. These contain information such as page numbers and cross-references (see
Chapter 7 [Cross references], page 48). If you now comment out \texttt{\includeonly}'s lines with
\texttt{preamble} and \texttt{amendments} and run \LaTeX{} again then the resulting document shows only
the material from \texttt{articles.tex}, not the material from \texttt{preamble.tex} or \texttt{amendments.tex}.
Nonetheless, all of the auxiliary information from the omitted files is still there, including
the starting page number of the chapter.

If the document preamble does not have \texttt{\includeonly} then \LaTeX{} will include all the
files you call for with \texttt{\include} commands.

The \texttt{\include} command makes a new page. To avoid that, see Section 24.3 [\texttt{input}],
page 232, (which, however, does not retain the auxiliary information).

See Section A.4 [Larger book template], page 265, for another example using \texttt{\include} and
\texttt{\includeonly}. That example also uses \texttt{\input} for some material that will not necessarily
start on a new page.

File names can involve paths.

\begin{verbatim}
\documentclass{book}
\includeonly{
   chapters/chap1,
}
\begin{document}
\include{chapters/chap1}
\end{document}
\end{verbatim}

To make your document portable across distributions and platforms you should avoid
spaces in the file names. The tradition is to instead use dashes or underscores. Nevertheless,
for the name ‘\texttt{amo\ amas\ amat}’, this works under \TeX{} Live on GNU/Linux:

\begin{verbatim}
\documentclass{book}
\includeonly{
   "amo\space amas\space amat"
}
\begin{document}
\include{"amo\space amas\space amat"}
\end{document}
\end{verbatim}

and this works under MiK\TeX{} on Windows:

\begin{verbatim}
\documentclass{book}
\includeonly{
   {"amo amas amat"}
}
\begin{document}
\include{"amo amas amat"}
\end{document}
\end{verbatim}
You cannot use \include inside a file that is being included or you get ‘\LaTeX Error: \include cannot be nested.’ The \include command cannot appear in the document preamble; you will get ‘\LaTeX Error: Missing $\begin{document}$.’

If a file that you \include does not exist, for instance if you \include{athiesm} but you meant \include{atheism}, then $\LaTeX$ does not give you an error but will warn you ‘No file athiesm.tex.’ (It will also create athiesm.aux.)

If you \include the root file in itself then you first get ‘\LaTeX Error: Can be used only in preamble.’ Later runs get ‘\TeX capacity exceeded, sorry [text input levels=15]’. To fix this, you must remove the inclusion \include{root} but also delete the file root.aux and rerun $\LaTeX$.

24.3 \input

Synopsis:
\input{filename}

$\LaTeX$ processes the file as if its contents were inserted in the current file. For a more sophisticated inclusion mechanism see Section 24.2 [\include & \includeonly], page 230.

If \texttt{filename} does not end in ‘.tex’ then $\LaTeX$ first tries the filename with that extension; this is the usual case. If \texttt{filename} ends with ‘.tex’ then $\LaTeX$ looks for the filename as it is.

For example, this
\input{macros}

will cause $\LaTeX$ to first look for macros.tex. If it finds that file then it processes its contents as thought they had been copy-pasted in. If there is no file of the name macros.tex then $\LaTeX$ tries the name macros, without an extension. (This may vary by distribution.)

To make your document portable across distributions and platforms you should avoid spaces in the file names. The tradition is to instead use dashes or underscores. Nevertheless, for the name ‘amo amas amat’, this works under $\TeX$ Live on GNU/Linux:
\input{"amo amas amat"}

and this works under MiK$\TeX$ on Windows:
\input{"amo amas amat"}
25 Front/back matter

25.1 Table of contents, list of figures, list of tables

Synopsis, one of:

\tableofcontents
\listoffigures
\listoftables

Produce a table of contents, or list of figures, or list of tables. Put the command in the input file where you want the table or list to go. You do not type the entries; for example, typically the table of contents entries are automatically generated from the sectioning commands \chapter, etc.

This example illustrates the first command, \tableofcontents. \LaTeX{} will produce a table of contents on the book’s first page.

\documentclass{book}
% \setcounter{tocdepth}{1}
\begin{document}
\tableofcontents\newpage
...
\chapter{...}
...
\section{...}
...
\subsection{...}
...
\end{document}

Uncommenting the second line would cause that table to contain chapter and section listings but not subsection listings, because the \section command has level 1. See Chapter 6 [Sectioning], page 36, for level numbers of the sectioning units. For more on the \texttt{tocdepth} see [Sectioning/tocdepth], page 37.

Another example of the use of \tableofcontents is in Section A.4 [Larger book template], page 265.

If you want a page break after the table of contents, write a \texttt{newpage} command after the \tableofcontents command, as above.

To make the table of contents, \LaTeX{} stores the information in an auxiliary file named \texttt{root-file.toc} (see Chapter 24 [Splitting the input], page 229). For example, this \LaTeX{} file \texttt{test.tex}

\documentclass{article}
\begin{document}
\tableofcontents\newpage
\section{First section}
\subsection{First subsection}
...
\end{document}
writes these lines to test.toc.
\contentsline {section}{First section}{1}
\contentsline {subsection}{First subsection}{1.1}

Each line contains a single command, \contentsline (see Section 25.1.4 [\contentsline], page 237). The first argument, the section or subsection, is the sectioning unit. The second argument has two components. The hook \numberline determines how the sectioning number, 1 or 1.1, appears in the table of contents (see Section 25.1.6 [\numberline], page 238). The remainder of the second argument of \contentsline, ‘First section’ or ‘First subsection’, is the sectioning title text. Finally, the third argument, ‘2’, is the page number on which this sectioning unit starts.

To typeset these lines, the document class provides \section-unit commands such as \section{text}{pagenumber} and \subsection{text}{pagenumber}. These commands often use the \dottedtocline command (see Section 25.1.1 [\dottedtocline], page 235).

A consequence of \LaTeX’s strategy of using auxiliary files is that to get the correct information in the document you must run \LaTeX twice, once to store the information and the second time to retrieve it. In the ordinary course of writing a document authors run \LaTeX a number of times, but you may notice that the first time that you compile a new document, the table of contents page will be empty except for its ‘Contents’ header. Just run \LaTeX again.

The commands \listoffigures and \listoftables produce a list of figures and a list of tables. Their information is stored in files with extension .lof and .lot. They work the same way as \tableofcontents but the latter is more common, so we use it for most examples.

You can manually add material to the table of contents, the list of figures, and the list of tables. For instance, add a line about a section to the table of contents with \addcontentsline{toc}{section}{text}. (see Section 25.1.2 [\addcontentsline], page 235). Add arbitrary material, that is, non-line material, with \addtocontents, as with the command \addtocontents{lof}{\vspace{2ex}}, which adds vertical space to the list of figures (see Section 25.1.3 [\addtocontents], page 236).

Lines in the table of contents, the list of figures, and the list of tables, have four parts. First is an indent. Next is a box into which sectioning numbers are placed, and then the third box holds the title text, such as ‘First section’. Finally there is a box up against the right margin, inside of which \LaTeX puts the page number box. For the indent and the width of the number box, see Section 25.1.1 [\dottedtocline], page 235. The right margin box has width \@tocrmarg and the page number is flush right in that space, inside a box of width \@pnumwidth. By default \@tocrmarg is 2.55em and \@pnumwidth is 1.55em. Change these as with \renewcommand{\@tocrmarg}{3.5em}.

CTAN has many packages for the table of contents and lists of figures and tables (see Section 2.8 [CTAN], page 8). The package tocloft is convenient for adjusting some aspects of the default such as spacing. And, tocbibbind will automatically add the bibliography, index, etc. to the table of contents.

To change the header for the table of contents page, do something like these commands before you call \tableofcontents, etc.
\renewcommand{\contentsname}{Table of Contents}
\renewcommand{\listfigurename}{Plots}  
\renewcommand{\listtablename}{Specifications}

Internationalization packages such as \texttt{babel} or \texttt{polyglossia} will change these headers depending on the chosen base language.

\subsection{\texttt{@dottedtocline}}

Synopsis:
\begin{verbatim}
@dottedtocline{section-level-num}{indent}{numwidth}{text}{pagenumber}
\end{verbatim}

Used internally by \LaTeX{} to format an entry line in the table of contents, list of figures, or list of tables. Authors do not directly enter \texttt{@dottedtocline} commands.

This command is typically used by \texttt{\l@section}, \texttt{\l@subsection}, etc., to format the content lines. For example, the \texttt{article.cls} file contains these definitions:
\begin{verbatim}
\newcommand*{\l@section}{@dottedtocline{1}{1.5em}{2.3em}}
\newcommand*{\l@subsection}{@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*{\l@subsubsection}{@dottedtocline{3}{7.0em}{4.1em}}
\end{verbatim}

In this example, \texttt{@dottedtocline} appears to have been given only three arguments. But tracing the internal code shows that it picks up the final \texttt{text} and \texttt{pagenumber} arguments in the synopsis from a call to \texttt{\contentsline} (see Section 25.1.4 \texttt{\contentsline}, page 237).

Between the box for the title text of a section and the right margin box, these \texttt{@dottedtocline} commands insert leaders, that is, evenly-spaced dots. The dot-to-dot space is given by the command \texttt{@dotsep}. By default it is 4.5 (it is in math units, aka. \texttt{mu}, which are \(1/18\) em). Change it using \texttt{\renewcommand}, as in \texttt{\renewcommand{\@dotsep}{3.5}}.

In the standard \texttt{book} class, \LaTeX{} does not use dotted leaders for the Part and Chapter table entries, and in the standard \texttt{article} class it does not use dotted leaders for Section entries.

\subsection{\texttt{\addcontentsline}}

Synopsis:
\begin{verbatim}
\addcontentsline{ext}{unit}{text}
\end{verbatim}

Add an entry to the auxiliary file with extension \texttt{ext}.

The following will result in an ‘Appendices’ line in the table of contents.
\begin{verbatim}
\addcontentsline{toc}{section}{\protect\textbf{Appendices}}
\end{verbatim}

It will appear at the same indentation level as the sections, will be in boldface, and will be assigned the page number associated with the point where the command appears in the input file.

The \texttt{\addcontentsline} command writes information to the file \texttt{root-name.ext}, where \texttt{root-name} is the file name of the root file (see Chapter 24 [Splitting the input], page 229). It writes that information as the text of the command \texttt{\contentsline{unit}{text}{num}}, where \texttt{num} is the current value of counter \texttt{unit} (see Section 25.1.4 \texttt{\contentsline}, page 237). The most common case is the table of contents and there \texttt{num} is the page number of the first page of \texttt{unit}.

This command is invoked by the sectioning commands \texttt{\chapter}, etc. (see Chapter 6 [Sectioning], page 36), and also by \texttt{\caption} inside a float environment (see Section 5.7.
But it is also directly used by authors. For example, an author writing a book whose style is to have an unnumbered preface may use the starred `\chapter*`. But that command leaves out table of contents information, which can be entered manually, as here.

```
\chapter*{Preface}
\addcontentsline{toc}{chapter}{\protect\numberline{}Preface}
```

In the root-name.toc file \LaTeX{} will put the line `\contentsline {chapter}{\numberline {3}Preface}{3}; note that the page number ‘3’ is automatically generated by the system, not entered manually.

All of the arguments for `\addcontentsline` are required.

- **ext**: Typically one of the strings `toc` for the table of contents, `lof` for the list of figures, or `lot` for the list of tables. The filename extension of the information file.
- **unit**: A string that depends on the value of the `ext` argument, typically one of:
  - `toc`: For the table of contents, this is the name of a sectional unit: `part`, `chapter`, `section`, `subsection`, etc.
  - `lof`: For the list of figures: `figure`.
  - `lot`: For the list of tables: `table`.
- **text**: The text of the entry. You must `\protect` any fragile commands (see Section 12.11 \[\protect\], page 123) used in it.

The `\addcontentsline` command has an interaction with `\include` (see Section 24.2 \[\include & \includeonly\], page 230). If you use them at the same level, as with `\addcontentsline{...}{...}{...}\include{...}` then lines in the table of contents can come out in the wrong order. The solution is to move `\addcontentsline` into the file being included.

If you use a `unit` that \LaTeX{} does not recognize, as with the typo here

```
\addcontentsline{toc}{section}{\protect\textbf{Appendices}}
```

then you don’t get an error but the formatting in the table of contents will not make sense.

### 25.1.3 `\addtocontents`

**Synopsis:**

```
\addtocontents{ext}{text}
```

Add `text`, which may be text or formatting commands, directly to the auxiliary file with extension `ext`. This is most commonly used for the table of contents so that is the discussion here, but it also applies to the list of figures and list of tables.

This will put some vertical space in the table of contents after the ‘Contents’ header.

```
\tableofcontents\newpage
\addtocontents{toc}{\protect\vspace*{3ex}}
```

This puts the word ‘Page’, in boldface, above the column of page numbers and after the header.
\addtocontents{toc}{\hfill\textbf{Page}\par}
\chapter{...}
This adds a line announcing work by a new author.
\addtocontents{toc}{%
\protect\vspace{2ex}
\textbf{Chapters by N. Other Author}\par}

The difference between \addtocontents and \addcontentsline is that the latter is strictly for lines, such as with a line giving the page number for the start of a new subset of the chapters. As the above examples show, \addtocontents is for material such as spacing.

The \addtocontents command has two arguments. Both are required.

ext Typically one of: toc for the table of contents, lof for the list of figures, or lot for the list of tables. The extension of the file holding the information.

text The text, and possibly commands, to be written.

The sectioning commands such as \chapter use the \addcontentsline command to store information. This command creates lines in the .toc auxiliary file containing the \contentsline command (see Section 25.1.2 [addcontentsline], page 235). In contrast, the command \addtocontents puts material directly in that file.

The \addtocontents command has an interaction with \include (see Section 24.2 [include & includeonly], page 230). If you use them at the same level, as with \addtocontents{...}{...}\include{...} then lines in the table of contents can come out in the wrong order. The solution is to move \addtocontents into the file being included.

25.1.4 \contentsline

Synopsis:
\contentsline{unit}{text}{pagenumber}

Used internally by \LaTeX to typeset an entry of the table of contents, list of figures, or list of tables (see Section 25.1 [Table of contents etc.], page 233). Authors do not directly enter \contentsline commands.

Usually adding material to these lists is done automatically by the commands \chapter, \section, etc. for the table of contents, or by the \caption command inside of a \figure or \table environment (see Section 8.10 [figure], page 60, and see Section 8.22 [table], page 84). Thus, where the root file is thesis.tex, and contains the declaration \tableofcontents, the command \chapter{Chapter One} produces something like this in the file thesis.toc.

\contentsline {chapter}{Chapter One}{3}

If the file contains the declaration \listoffigures then a figure environment involving \caption{Test} will produce something like this in thesis.lof.

\contentsline {figure}{
umberline {1.1}{\ignorespaces Test}}{6}

To manually add material, use \addcontentsline{filetype}{unit}{text}, where filetype is toc, lof, or lot (see Section 25.1.2 [addcontentsline], page 235).

For manipulating how the \contentsline material is typeset, see the tocloft package.

Note that the hyperref package changes the definition of \contentsline (and \addcontentsline) to add more arguments, to make hyperlinks. This is the source of
the error *Argument of \contentsline has an extra *} when one adds/remove the use of package *hyperref* and a compilation was already run. Fix this error by deleting the .toc or .lof or .lot file, and running \TeX{} again.

### 25.1.5 \nofiles

**Synopsis:**

\nofiles

Prevent \TeX{} from writing any auxiliary files. The only output will be the .log and .pdf (or .dvi) files. This command must go in the preamble.

Because of the \nofiles command this example will not produce a .toc file.

```latex
\documentclass{book}
\nofiles
\begin{document}
\tableofcontents
\chapter{...}
...
```

\TeX{} will not erase any existing auxiliary files, so if you insert the \nofiles command after you have run the file and gotten a .toc then the table of contents page will continue to show the old information.

### 25.1.6 \numberline

**Synopsis:**

\numberline{number}

Typeset its argument flush left in a box. This is used in a \contentsline command to typeset the section number (see Section 25.1.4 \contentsline, page 237).

For example, this line in a .toc file causes the 1.1 to be typeset flush left.

```latex
\contentsline {subsection}{\numberline {1.1}Motivation}{2}
```

By default, \TeX{} typesets the section numbers in a box of length \@tempdima. That length is set by the commands \l@section, \l@subsection, etc. Put section numbers inside a natural-width box with \renewcommand{\numberline}{1}\{#1\} before \tableofcontents.

This command is fragile so you may need to precede it with \protect (see Section 12.11 \protect, page 123). An example is the use of \protect in this command,

```latex
\addcontentsline{toc}{section}{\protect\numberline{}Summary}
```

to get the \numberline into the \contentsline command in the .toc file: \contentsline {section}{\numberline {1}\{}Summary\}\{6\} (the page number ‘6’ is automatically added by \TeX{}; see Section 25.1.2 \addcontentsline, page 235).

### 25.2 Indexes

If you tell \TeX{} what terms you want to appear in an index then it can produce that index, alphabetized and with the page numbers automatically maintained. This illustrates the basics.

\documentclass{article}
\usepackage{makeidx} % Provide indexing commands
\makeindex
% \usepackage{showidx} % Show marginal notes of index entries
...
\begin{document}
...
Wilson's Theorem\index{Wilson's Theorem}
says that a number $n>1$ is prime if and only if the factorial
of $n-1$ is congruent to $-1$ modulo~$n$.\index{congruence!and Wilson's Theorem}
...
\printindex
\end{document}

As that shows, declare index entries with the \index command (see Section 25.2.2 [\index],
page 240). When you run \LaTeX, the \index writes its information, such as 'Wilson's
Theorem' and the page number, to an auxiliary file whose name ends in .idx. Next, to
alphabetize and do other manipulations, run an external command, typically \makeindex
(see Section 25.2.3 [makeindex], page 242), which writes a file whose name ends in .ind.
Finally, \printindex brings this manipulated information into the output (see Section 25.2.4
[\printindex], page 244).

Thus, if the above example is in the file numth.tex then running 'pdflatex numth' will
save index entry and page number information to numth.idx. Then running 'makeindex
numth' will alphabetize and save the results to numth.ind. Finally, again running 'pdflatex
numth' will show the desired index, at the place where the \printindex command is in the
source file.

There are many options for the output. An example is that the exclamation point in
\index{congruence!and Wilson's Theorem} produces a main entry of 'congruence' with
a subentry of 'and Wilson's Theorem'. For more, see Section 25.2.3 [makeindex], page 242.

The \makeindex and \printindex commands are independent. Leaving out the
\makeindex will stop \LaTeX from saving the index entries to the auxiliary file. Leaving out
the \printindex will cause \LaTeX to not show the index in the document output.

There are many packages in the area of indexing. The \showidx package causes each
index entries to be shown in the margin on the page where the \index appears. This can help in preparing the index. The multind package, among others, supports multiple indexes. See also the \TeX FAQ entry on this topic, https://www.texfaq.org/FAQ-multind, and

25.2.1 Produce the index manually
Documents that are small and static can have a manually produced index. This will make a
separate page labeled 'Index', in twocolumn format.

\begin{theindex}
  \item acorn squash, 1
  \subitem maple baked, 2
  \indexspace
  \item bacon, 3
\end{theindex}
Note that the author must enter the page numbers, which is tedious and which will result in wrong numbers if the document changes. This is why in most cases automated methods such as \texttt{makeindex} are best. See Section 25.2 [Indexes], page 238.

However we cover the commands for completeness, and because the automated methods are based on these commands. There are three levels of entries. As the example shows, a main entry uses \texttt{item}, subentries use \texttt{subitem}, and the lowest level uses \texttt{subsubitem}. Blank lines between entries have no effect. The example above includes \texttt{indexspace} to produce vertical space in the output that some index styles use before the first entry starting with a new letter.

\textbf{25.2.2 \texttt{\index}}

Synopsis:

\texttt{\index{index-entry-string}}

Declare an entry in the index. This command is fragile (see Section 12.11 [\texttt{\protect}], page 123).

For example, as described in Section 25.2 [Indexes], page 238, one way to get an index from what’s below is to compile the document with \texttt{pdflatex test}, then process the index entries with \texttt{makeindex test}, and then compile again with \texttt{pdflatex test}.

% file test.tex
...
W~Ackermann (1896--1962).\index{Ackermann}
...
Ackermann function\index{Ackermann!function}
...
rate of growth\index{Ackermann!function!growth rate}

All three index entries will get a page number, such as ‘Ackermann, 22’. \LaTeX{} will format the second as a subitem of the first, on the line below it and indented, and the third as a subitem of the second. Three levels deep is as far as you can nest subentries. (If you add \texttt{\index{Ackermann!function!growth rate!comparison}} then \texttt{makeindex} says ‘Scanning input file test.idx...done (4 entries accepted, 1 rejected)’ and the fourth level is silently missing from the index.)

If you enter a second \texttt{\index} with the same \texttt{index-entry-string} then you will get a single index entry with two page numbers (unless they happen to fall on the same page). Thus, adding as for Ackermann. \texttt{\index{Ackermann}} later in the same document as above will give an index entry like ‘Ackermann, 22, 151’. Also, you can enter the index entries in any order, so for instance \texttt{\index{Ackermann!function}} could come before \texttt{\index{Ackermann}}.

Get a page range in the output, like ‘Hilbert, 23--27’, as here.

W~Ackermann (1896--1962).\index{Ackermann}
...
D~Hilbert (1862--1943)\index{Ackermann!Hilbert|)}
...
disapproved of his marriage.\index{Ackermann!Hilbert|)}
If the beginning and ending of the page range are equal then the system just gives a single page number, not a range.

If you index subentries but not a main entry, as with `\index{Jones!program} and `\index{Jones!results}`, then the output is the item ‘Jones’ with no comma or page number, followed by two subitems, like ‘program, 50’ and ‘results, 51’.

Generate an index entry that says ‘see’ by using a vertical bar character: `\index{Ackermann!function|see{P\'eter’s function}}`. You can instead get ‘see also’ with `\seealso`. (The text ‘see’ is defined by `\seename`, and ‘see also’ by `\alsoname`. You can redefine these either by using an internationalization package such as `babel` or `polyglossia`, or directly as with `\renewcommand\alsoname{Also see}`.)

The ‘see’ feature is part of a more general functionality. After the vertical bar you can put the name of a one-input command, as in `\index{group|textit}` (note the missing backslash on the `\textit` command) and the system will apply that command to the page number, here giving something like `\textit{7}`. You can define your own one-input commands, such as `\newcommand{\definedpage}[1]{\color{blue}#1}` and then `\index{Ackermann!function|definedpage}` will give a blue page number (see Chapter 22 [Color], page 215). Another, less practical, example is this,

```
\newcommand{indexownpage}[1]{#1, \thepage}
... Epimenides. \index{self-reference|indexownpage}
```

which creates an entry citing the page number of its own index listing.

The two functions just described combine, as here

```
\index{Ackermann!function|definedpage}
```

```
\index{Ackermann!function|)
```

which outputs an index entry like ‘function, 23--27’ where the page number range is in blue.

Consider an index entry such as ‘α-ring’. Entering it as `$\alpha$-ring` will cause it to be alphabetized according to the dollar sign. You can instead enter it using an at-sign, as `\index{alpha-ring@$\alpha$-ring}`. If you specify an entry with an at-sign separating two strings, `pos@text`, then `pos` gives the alphabetical position of the entry while `text` produces the text of the entry. Another example is that `\index{Saint Michael's College@SMC}` produces an index entry ‘SMC’ alphabetized into a different location than its spelling would naturally give it.

To put a !, or @, or |, or " character in an index entry, escape it by preceding it with a double quote, ". (The double quote gets deleted before alphabetization.)

A number of packages on CTAN have additional functionality beyond that provided by `makeidx`. One is `index`, which allows for multiple indices and contains a command `\index*{index-entry-string}` that prints the `index-entry-string` as well as indexing it.

The `\index` command writes the indexing information to the file `root-name.idx` file. Specifically, it writes text of the command `\indexentry{index-entry-string}{page-num}`, where `page-num` is the value of the `\thepage` counter. On occasion, when the `\printindex` command is confused, you have to delete this file to start with a fresh slate.

If you omit the closing brace of an `\index` command then you get a message like this.

```
Runaway argument? \{Ackermann!function
```
25.2.3 makeindex

Synopsis, one of:

\begin{itemize}
\item makeindex \textit{filename}
\item makeindex -s \textit{style-file} \textit{filename}
\item makeindex \textit{options} \textit{filename0} ...
\end{itemize}

Sort, and otherwise process, the index information in the auxiliary file \textit{filename}. This is a command line program. It takes one or more raw index files, \textit{filename.idx} files, and produces the actual index file, the \textit{filename.ind} file that is input by \texttt{\textbackslash printindex} (see Section 25.2.4 \cite{printindex}, page 244).

The first form of the command suffices for many uses. The second allows you to format the index by using an \textit{index style file}, a \textit{.isty} file. The third form is the most general; see the full documentation on CTAN.

This is a simple \textit{.isty} file.

\begin{verbatim}
\% book.isty
\% $ makeindex -s book.isty -p odd book.idx
\% creates the index as book.ind, starting on an odd page.
preamble
"\pagestyle{empty}
  \small
  \begin{theindex}
  \thispagestyle{empty}"

postamble
"\n
  \end{theindex}"\end{verbatim}

The description here covers only some of the index formatting possibilities in \textit{style-file}. For a full list see the documentation on CTAN.

A style file consists of a list of pairs: \textit{specifier} and \textit{attribute}. These can appear in the file in any order. All of the \textit{attributes} are strings, except where noted. Strings are surrounded with double quotes, ", and the maximum length of a string is 144 characters. The \texttt{\textbackslash n} is for a newline and \texttt{\textbackslash t} is for a tab. Backslashes are escaped with another backslash, \texttt{\textbackslash \textbackslash}. If a line begins with a percent sign, \texttt{\%}, then it is a comment.

\begin{itemize}
\item \texttt{preamble} \quad Preamble of the output index file. Defines the context in which the index is formatted. Default: "\texttt{\textbackslash begin\{theindex\}\textbackslash n}".
\item \texttt{postamble} \quad Postamble of the output index file. Default: "\texttt{\textbackslash n\textbackslash n\textbackslash end\{theindex\}\textbackslash n}".
\item \texttt{group\_skip} \quad Traditionally index items are broken into groups, typically a group for entries starting with letter ‘a’, etc. This specifier gives what is inserted when a new group begins. Default: "\texttt{\textbackslash n\textbackslash n \indexspace}\textbackslash n}" (\texttt{\indexspace} is a command inserting a rubber length, by default 10pt plus5pt minus3pt).
\end{itemize}
lethead_flag
An integer. It governs what is inserted for a new group or letter. If it is 0
(which is the default) then other than group_skip nothing will be inserted
before the group. If it is positive then at a new letter the lethead_prefix
and lethead_suffix will be inserted, with that letter in uppercase between them.
If it is negative then what will be inserted is the letter in lowercase. The default
is 0.

lethead_prefix
If a new group begins with a different letter then this is the prefix inserted
before the new letter header. Default: ""

lethead_suffix
If a group begins with a different letter then this is the suffix inserted after the
new letter header. Default: "".

text
What is put between two level 0 items. Default: "\n
item_0
What is put between two level 0 items. Default: "\n
item_1
Put between two level 1 items. Default: "\n
item_2
Put between two level 2 items. Default: "\n
item_01
What is put between a level 0 item and a level 1 item. Default: "\n
item_x1
What is put between a level 0 item and a level 1 item in the case that the level 0
item doesn’t have any page numbers (as in \index{aaa|see{bbb}}). Default:
"\n
item_x2
What is put between a level 1 item and a level 2 item, if the level 1 item doesn’t
have page numbers. Default: "\n
delim_0
Delimiter put between a level 0 key and its first page number. Default: a comma
followed by a blank, ", ".

delim_1
Delimiter put between a level 1 key and its first page number. Default: a comma
followed by a blank, ", ".

delim_2
Delimiter between a level 2 key and its first page number. Default: a comma
followed by a blank, ", ".

delim_n
Delimiter between two page numbers for the same key (at any level). Default: a
comma followed by a blank, ", ".

delim_r
What is put between the starting and ending page numbers of a range. Default:
"--".

line_max
An integer. Maximum length of an index entry’s line in the output, beyond
which the line wraps. Default: 72.

indent_space
What is inserted at the start of a wrapped line. Default: "\t\t".
indent_length
A number. The length of the wrapped line indentation. The default indent_s
pace is two tabs and each tab is eight spaces so the default here is 16.

page_precedence
A document may have pages numbered in different ways. For example, a book
may have front matter pages numbered in lowercase roman while main matter
pages are in arabic. This string specifies the order in which they will appear in
the index. The makeindex command supports five different types of numerals:
lowercase roman r, and numeric or arabic n, and lowercase alphabetic a, and
uppercase roman R, and uppercase alphabetic A. Default: "rnaRA".

There are a number of other programs that do the job makeindex does. One is xindy
(https://ctan.org/pkg/xindy), which does internationalization and can process indexes
for documents marked up using \LaTeX{} and a number of other languages. It is written in
Lisp, highly configurable, both in markup terms and in terms of the collating order of the
text, as described in its documentation.

A more recent indexing program supporting Unicode is xindex, written in Lua (https://
ctan.org/pkg/xindex).

25.2.4 \printindex
Synopsis:
\printindex
Place the index into the output.

To get an index you must first include \usepackage{makeidx}\makeindex in the docu-
ment preamble and compile the document, then run the system command makeindex, and
then compile the document again. See Section 25.2 [Indexes], page 238, for further discussion
and an example of the use of \printindex.

25.3 Glossaries
Synopsis:
\usepackage{glossaries} \makeglossaries
\newglossaryentry{label}{settings}
\gls{label}.
\printglossaries
The glossaries package allows you to make glossaries, including multiple glossaries, as
well as lists of acronyms.

To get the output from this example, compile the document (for instance with pdflatex
filename), then run the command line command makeglossaries filename, and then
compile the document again.
\documentclass{...}
\usepackage{glossaries} \makeglossaries
\newglossaryentry{tm}{%
  name=\text{Turing machine},
  description={A model of a machine that computes. The model is simple
  but can compute anything any existing device can compute.
  It is the standard model used in Computer Science.},
}
\begin{document}
Everything begins with the definition of a \text{\gls{tm}}.
...
\printglossaries
\end{document}

That gives two things. In the main text it outputs ‘... definition of a Turing machine’. In addition, in a separate sectional unit headed ‘Glossary’ there appears a description list. In boldface it says ‘Turing machine’ and the rest of the item says in normal type ‘A model of a machine ... Computer Science’.

The command \texttt{\makeglossary} opens the file that will contain the entry information, \texttt{root-file.glo}. Put the \texttt{\printglossaries} command where you want the glossaries to appear in your document.

The \texttt{glossaries} package is very powerful. For instance, besides the commands \texttt{\newglossaryentry} and \texttt{\gls}, there are similar commands for a list of acronyms. See the package documentations on CTAN.

\subsection{\texttt{\newglossaryentry}}

Synopsis, one of:

\begin{verbatim}
\newglossaryentry{label}
{
  name={name},
  description={description},
  other options, ...
}
\end{verbatim}

or

\begin{verbatim}
\longnewglossaryentry{label}
{
  name={name},
  other options ..., 
}
{description}
\end{verbatim}

Declare a new entry for a glossary. The \textit{label} must be unique for the document. The settings associated with the label are pairs: \texttt{key=value}.

This puts the blackboard bold symbol for the real numbers in the glossary.

\begin{verbatim}
\newglossaryentry{R}
{
  name={\ensuremath\mathbb{R}},
  description={the real numbers},
}
\end{verbatim}
Use the second command form if the description spans more than one paragraph.

For a full list of keys see the package documentation on CTAN but here are a few.

name (Required.) The word, phrase, or symbol that you are defining.

description (Required.) The description that will appear in the glossary. If this has more
than one paragraph then you must use the second command form given in the
synopsis.

plural The plural form of name. Refer to the plural form using \glsp{} or \Glsp{} (see
Section 25.3.2 [\gls{}], page 246).

sort How to place this entry in the list of entries that the glossary holds.

symbol A symbol, such as a mathematical symbol, besides the name.

25.3.2 \gls{}

Synopsis, one of:

\gls{label}
\glsp{label}
\Gls{label}
\Glsp{label}

Refer to a glossary entry. The entries are declared with \newglossaryentry (see Sec-
ction 25.3.1 [\newglossaryentry], page 245).

This

\newglossaryentry{N}{%
  name={the natural numbers},
  description={The numbers $0$, $1$, $2$, $\ldots$},
  symbol={\ensuremath{\mathbb{N}}},
}

... 

Consider \gls{N}.

gives the output ‘Consider the natural numbers’.

The second command form \glsp{label} produces the plural of name (by default it
tries adding an ‘s’). The third form capitalizes the first letter of name, as does the fourth
form, which also takes the plural.
26 Letters

Synopsis:
\documentclass{letter}
\address{senders address} \% return address
\signature{sender name}
\begin{document}
\begin{letter}{recipient address}
\opening{salutation}
\letter body
\closing{closing text}
\end{letter}

\end{document}

Produce one or more letters.

Each letter is in a separate letter environment, whose argument recipient address often contains multiple lines separated with a double backslash (\\). For example, you might have:

\begin{letter}{Ninon de l'Enclos \\
1'h\^otel Sagonne}

\end{letter}

The start of the letter environment resets the page number to 1, and the footnote number to 1 also.

The sender address and sender name are common to all of the letters, whether there is one or more, so these are best put in the preamble. As with the recipient address, often sender address contains multiple lines separated by a double backslash (\\). \LaTeX{} will put the sender name under the closing, after a vertical space for the traditional hand-written signature.

Each letter environment body begins with a required \opening command such as \opening{Dear Madam or Sir:}. The letter body text is ordinary \LaTeX{} so it can contain everything from enumerated lists to displayed math, except that commands such as \chapter that make no sense in a letter are turned off. Each letter environment body typically ends with a \closing command such as \closing{Yours,}.

Additional material may come after the \closing. You can say who is receiving a copy of the letter with a command like \cc{the Boss \ the Boss's Boss}. There’s a similar \encl command for a list of enclosures. And, you can add a postscript with \ps.

\LaTeX{}'s default is to indent the sender name and the closing above it by a length of \longindentation. By default this is 0.5\textwidth. To make them flush left, put \setlength{\longindentation}{0em} in your preamble.

To set a fixed date use something like \renewcommand{\today}{1958-Oct-12}. If put in your preamble then it will apply to all the letters.

This example shows only one letter environment. The three lines marked as optional are typically omitted.

\documentclass{letter}
Chapter 26: Letters 248

\address{Sender's street \ \ Sender's town}
\signature{Sender's name \ \ Sender's title}
% optional: \location{Mailbox 13}
% optional: \telephone{(102) 555-0101}
\begin{document}
\begin{letter}{Recipient's name \ \ Recipient's address}
\opening{Sir:}
% optional: \thispagestyle{firstpage}
I am not interested in entering a business arrangement with you.
\closing{Your most humble, etc.,}
\end{letter}
\end{document}

These commands are used with the letter class.

26.1 \address

Synopsis:

\address{senders address}

Specify the return address, as it appears on the letter and on the envelope. Separate multiple lines in senders address with a double backslash, \.

Because it can apply to multiple letters this declaration is often put in the preamble. However, it can go anywhere, including inside an individual letter environment.

This command is optional: if you do not use it then the letter is formatted with some blank space on top, for copying onto pre-printed letterhead paper. If you do use the \address declaration then it is formatted as a personal letter.

Here is an example.

\address{Stephen Maturin \ 
 The Grapes of the Savoy}

26.2 \cc

Synopsis:

\cc{name0 \ \ 
 ... }

Produce a list of names to which copies of the letter were sent. This command is optional. If it appears then typically it comes after \closing. Put the names on different lines by separating them with a double backslash, \, as in:

\cc{President \ 
 Vice President}

26.3 \closing

Synopsis:

\closing{text}
Chapter 26: Letters

Produce the letter's closing. This is optional, but usual. It appears at the end of a letter, above a handwritten signature. For example:

\closing{Regards,}

26.4 \encl

Synopsis:

\encl{first enclosed object \ \ .... }

Produce a list of things included with the letter. This command is optional; when it is used, it typically is put after \closing. Separate multiple lines with a double backslash, \.

\encl{License \ \ 
Passport}

26.5 \location

Synopsis:

\location{text}

The text appears centered at the bottom of the page. It only appears if the page style is firstpage.

26.6 \makelabels

Synopsis:

\makelabels % in preamble

Optional, for a document that contains letter environments. If you just put \makelabels in the preamble then at the end of the document you will get a sheet with labels for all the recipients, one for each letter environment, that you can copy to a sheet of peel-off address labels.

Customize the labels by redefining the commands \startlabels, \mlabel, and \returnaddress (and perhaps \name) in the preamble. The command \startlabels sets the width, height, number of columns, etc., of the page onto which the labels are printed. The command \mlabel{return address}{recipient address} produces the two labels (or one, if you choose to ignore the return address) for each letter environment. The first argument, return address, is the value returned by the macro \returnaddress. The second argument, recipient address, is the value passed in the argument to the letter environment. By default \mlabel ignores the first argument, the return address, causing the default behavior described in the prior paragraph.

This illustrates customization. Its output includes a page with two columns having two labels each.

\documentclass{letter}
\renewcommand*{\returnaddress}{Fred McGuilicuddy \ 
Oshkosh, Mineola 12305}
\newcommand*{\originalMlabel}{}
\let\originalMlabel\mlabel
Chapter 26: Letters

\begin{document}
\begin{letter}{A Einstein \\\n112 Mercer Street \\\nPrinceton, New Jersey, USA 08540}
\end{letter}
\begin{letter}{K G"odel \\\n145 Linden Lane \\\nPrinceton, New Jersey, USA 08540}
\end{letter}
\end{document}

The first column contains the return address twice. The second column contains the address for each recipient.

The package \texttt{enlab} makes formatting the labels easier, with standard sizes already provided. The preamble lines \texttt{\usepackage[personalenvelope]{enlab}} and \texttt{\makelabels} are all that you need to print envelopes.

\subsection{26.7 \texttt{name}}

Synopsis:

\begin{verbatim}
\name{name}
\end{verbatim}

Optional. Sender’s name, used for printing on the envelope together with the return address.

\subsection{26.8 \texttt{opening}}

Synopsis:

\begin{verbatim}
\opening{salutation}
\end{verbatim}

Required. Follows the \texttt{\begin{letter}{...}}. The argument \texttt{salutation} is mandatory. For instance:

\begin{verbatim}
\opening{Dear John:}
\end{verbatim}

\subsection{26.9 \texttt{ps}}

Synopsis:

\begin{verbatim}
\ps{text}
\end{verbatim}

Add a postscript. This command is optional and usually is used after \texttt{\closing}.

\begin{verbatim}
\ps{P.S. After you have read this letter, burn it. Or eat it.}
\end{verbatim}
26.10 \texttt{\textbackslash signature}

Synopsis:

\begin{verbatim}
\textbackslash signature\{first line \ \ \ \ \ \ ... \}
\end{verbatim}

The sender's name. This command is optional, although its inclusion is usual.

The argument text appears at the end of the letter, after the closing. \LaTeX{} leaves some vertical space for a handwritten signature. Separate multiple lines with a double backslash, \textbackslash. For example:

\begin{verbatim}
\textbackslash signature\{J Fred Muggs \ \ White House\}
\end{verbatim}

\LaTeX{}'s default for the vertical space from the \texttt{\textbackslash closing} text down to the \texttt{\textbackslash signature} text is 6\textit{\textbackslash medskipamount}, which is six times \texttt{\textbackslash medskipamount} (where \texttt{\textbackslash medskipamount} is equal to a \texttt{\textbackslash parskip}, which in turn is defined by default here to 0.7em).

This command is usually in the preamble, to apply to all the letters in the document. To have it apply to one letter only, put it inside a \texttt{letter} environment and before the \texttt{\textbackslash closing}.

You can include a graphic in the signature as here.

\begin{verbatim}
\textbackslash signature\{\vspace{-6\textbackslash medskipamount}\includegraphics{sig.png}\ \ My name\}
\end{verbatim}

For this you must put \texttt{\usepackage{graphicx}} in the preamble (see Chapter 21 [Graphics], page 204).

26.11 \texttt{\textbackslash telephone}

Synopsis:

\begin{verbatim}
\textbackslash telephone\{number\}
\end{verbatim}

The sender's telephone number. This is typically in the preamble, where it applies to all letters. This only appears if the \texttt{firstpage} pagestyle is selected. If so, it appears on the lower right of the page.
27 Input/output

\L A\TeX uses the ability to write to a file and later read it back in to build document components such as a table of contents or index. You can also read a file that other programs written, or write a file for others to read. You can communicate with users through the terminal. And, you can issue instructions for the operating system.

27.1 \openin & \openout

Synopsis:

\openin number=filename

or:

\openout number=filename

Open a file for reading material, or for writing it. In most engines, the number must be between 0 and 15, as in \openin3; in Lua\L A\TeX, number can be between 0 and 127.

Here \TeX opens the file presidents.tex for reading.

\newread\presidentsfile\openin\presidentsfile=presidents\typeout{presidentsfile is \the\presidentsfile}\read\presidentsfile to\presidentline\typeout{\presidentline}

The \newread command allocates input stream numbers from 0 to 15 (there is also a \newwrite). The \presidentsfile is more memorable but under the hood it is still a number; the first \typeout gives something like 'presidentsfile is 1'. In addition, \newread keeps track of the allocation so that if you use too many then you get an error like '! No room for a new \read'. The second \typeout gives the first line of the file, something like '1 Washington, George'.

Ordinarily \TeX will not try to open the file until the next page shipout. To change this, use \immediate\openin number=filename or \immediate\openout number=filename.

Close files with \closein number and \closeout number.

How \L A\TeX handles filenames varies among distributions, and even can vary among versions of a distribution. If the file does not have an extension then \TeX will add a .tex. This creates presidents.tex, writes one line to it, and closes it.

\newwrite\presidentsfile\openout\presidentsfile=presidents\write\presidentsfile{1 Washington, George}\closeout\presidentsfile

But filenames with a period can cause trouble: if \TeX finds a filename of presidents.dat it could look first for presidents.dat.tex and later for presidents.dat, or it could do the opposite. Your distribution's documentation should say more, and if you find something that works for you then you are good, but to ensure complete portability the best thing is to use file names containing only the twenty six ASCII letters (not case-sensitive) and the ten digits, along with underscore and dash, and in particular with no dot or space.

For \openin, if \TeX cannot find the file then it does not give an error. It just considers that the stream is not open (test for this with \ifeof; one recourse is the command
\InputIfFileExists, see Section 12.14 [Class and package commands], page 125). If you try to use the same number twice, \LaTeX won’t give you an error. If you try to use a bad number then you get an error message like ‘! Bad number (16). <to be read again> = 1.30 \openin16=test.jh’.

27.2 \read

Synopsis:

\read number to macro

Make the command macro contain the next line of input from text stream number, as in \read5 to\data.

This opens the file email.tex for reading, puts the contents of the first line into the command \email, and then closes the file.

\newread\recipientfile
\openin\recipientfile=email
\read\recipientfile to\email
\typeout{Email address: \email}
\closein\recipientfile

If number is outside the range from 0 to 15 or if no file of that number is open, or if the file has ended, then \read will take input from the terminal (or exit if interaction is not allowed, e.g., \nonstopmode; see [interaction modes], page 259). (However, the natural way in \LaTeX to take input from the terminal is \typein (see Section 27.3 [\typein], page 253.)

To read an entire file as additional \LaTeX source, use \input (see Section 24.3 [\input], page 232) or \include (see Section 24.2 [\include & \includeonly], page 230).

A common reason to want to read from a data file is to do mail merges. CTAN has a number of packages for that; one is datatool.

27.3 \typein

Synopsis, one of:

\typein{prompt-msg}
\typein{cmd}{prompt-msg}

Print prompt-msg on the terminal and cause \LaTeX to stop and wait for you to type a line of input. This line of input ends when you hit the return key.

For example, this

As long as I live I shall never forget \typein{Enter student name:}
coupled with this command line interaction

Enter student name:

@typein=Aphra Behn

gives the output ‘... never forget Aphra Behn’.

The first command version, \typein{prompt-msg}, causes the input you typed to be processed as if it had been included in the input file in place of the \typein command.
In the second command version the optional argument \texttt{cmd} argument must be a command name, that is, it must begin with a backslash, \\texttt{\}. This command name is then defined or redefined to be the input that you typed. For example, this
\begin{verbatim}
\typein{\student}{Enter student name:}
\typeout{Recommendation for \student .}
\end{verbatim}
gives this output on the command line,

Enter student name:

\begin{verbatim}
\student=John Dee
Recommendation for John Dee.
\end{verbatim}

where the user has entered ‘John Dee.’

\section*{27.4 \texttt{\write}}

Synopsis:
\begin{verbatim}
\write number\{string\}
\end{verbatim}
Write \texttt{string} to the log file, to the terminal, or to a file opened by \texttt{\openout}. For instance, \texttt{\write6} writes to text stream number 6.

If the following appears in basefile.tex then it opens basefile.jh, writes ‘Hello World!’ and a newline to it, and closes that file.
\begin{verbatim}
\newwrite\myfile
\immediate\openout\myfile=\jobname.jh \ % \jobname is root file basename
\end{verbatim}
The `\newwrite` allocates a stream number, giving it a symbolic name to make life easier, so that `stream \newwrite\myfile\the\myfile` produces something like ‘stream 3’. Then `\openout` associates the stream number with the given file name. \TeX ultimately executed `\write3` which puts the string in the file.

Typically `number` is between 0 and 15, because typically \TeX authors follow the prior example and the number is allocated by the system. If `number` is outside the range from 0 to 15 or if it is not associated with an open file then \TeX writes `string` to the log file. If `number` is positive then in addition \TeX writes `string` to the terminal.

Thus, `test \write-1{Hello World!}` puts ‘Hello World!’ followed by a newline in the log file. (This is what the `\wlog` command does; see Section 27.5.3 [\wlog], page 257). And `\write100{Hello World!}` puts the same in the log file but also puts ‘Hello World!’ followed by a newline in the terminal output. (But 16, 17, and 18 are special as `number`; see below.)

In Lua\TeX, instead of 16 output streams there are 256 (see Section 2.3 [\TeX engines], page 3).

Use `\write@auxout{string}` to write to the current `.aux` file, which is associated with either the root file or with the current include file; and use `\write@mainaux{string}` to write to the main `.aux`. These symbolic names are defined by \TeX.

By default \TeX does not write `string` to the file right away. This is because, for example, you may need `\write` to save the current page number, but when \TeX comes across a `\write` it typically does not know what the page number is, since it has not yet done the page breaking. So, you use `\write` in one of three contexts:

1. With the first, \TeX writes `string` to the file immediately. Any macros in `string` are fully expanded (just as in `\edef`) so to prevent expansion you must use `\noexpand`, `\toks`, etc., except that you should use `#` instead of `##`.
2. With the second, `string` is stored on the current list of things (as a \TeX “whatsit” item) and kept until the page is shipped out and likewise the macros are unexpanded until `\shipout`. At `\shipout`, `string` is fully expanded.
3. The third, `\protected@write`, is like the second except that you can use `\protect` to avoid expansion. The extra first argument allows you to locally insert extra definitions to make more macros protected or to have some other special definition for the write.

As a simple example of expansion with `\write`, `string` here contains a control sequence `\triplex` which we’ve defined to be the text ‘XYZ’:

```
\newwrite\jhfile
\openout\jhfile=test.jh
\newcommand{\triplex}{XYZ}
\write\jhfile{test \triplex test}
```

This results in the file `test.jh` containing the text ‘test XYZtest’ followed by a newline.
The cases where number is 16, 17, or 18 are special. Because of \texttt{write}'s behavior when number is outside the range from 0 to 15 described above, in Plain \TeX \texttt{write16} and \texttt{write17} were sometimes used to write to the log file and the terminal; however, in \LaTeX, the natural way to do that is with \texttt{typeout} (see Section 27.4 [\texttt{typeout}], page 254). The \texttt{write18} command is even more special; modern \TeX systems use it for giving commands to the operating system (see Section 27.5.4 [\texttt{write18}], page 257).

Ordinarily \texttt{write} outputs a single line. You can include a newline with \texttt{\^J}. Thus, this produces two lines in the log file:

\texttt{\wlog{Parallel lines have a lot in common.\^JBut they never meet.}}

A common case where authors need to write their own file is for answers to exercises, or another situation where you want to write out verbatim, without expanding the macros. CTAN has a number of packages for this; one is \texttt{answers}.

### 27.5.1 \texttt{write} and security

The ability to write files raises security issues. If you compiled a downloaded \LaTeX file and it overwrote your password file then you would be justifiably troubled.

Thus, by default \TeX systems only allow you to open files for writing that are in the current directory or output directory, if specified (see [output directory], page 260), or in a subdirectory of those. So, this code

\begin{verbatim}
\newwrite\jhfile
\openout\jhfile=../test.jh
\end{verbatim}

gives an error like:

\texttt{Not writing to ../test.jh (openout\_any = p).
! I can't write on file `../test.jh'
}

You can get just such an error when using commands such as \texttt{\include{../filename}} because \LaTeX will try to open \texttt{../filename.aux}. The simplest solution is to put the included files in the same directory as the root file, or in subdirectories.

### 27.5.2 \texttt{message}

Synopsis:

\begin{verbatim}
\message{string}
\end{verbatim}

Write \texttt{string} to the log file and the terminal.

Typically, \LaTeX authors use \texttt{typeout} (see Section 27.4 [\texttt{typeout}], page 254). It allows you to use \texttt{protect} on any fragile commands in \texttt{string} (see Section 12.11 [\texttt{protect}], page 123). But \texttt{typeout} always inserts a newline at the end of \texttt{string} while \texttt{message} does not, so the latter can be useful.

With this example document body.

\begin{verbatim}
before\message{One Two}\message{Three}\message{Four\^J}
\end{verbatim}

under some circumstances (see below) \LaTeX writes the following to both the terminal and the log file.

\texttt{One Two Three Four}

\texttt{I declare a thumb war.}
The `\textbackslash J` produces a newline. Also, in the output document, between ‘before’ and ‘After’ will be a single space (from the end of line following ‘I’).

While `\message` allows you more control over formatting, a gotcha is that \LaTeX{} may mess up that formatting because it inserts line breaks depending on what it has already written. Contrast this document body, where the ‘Two’ has moved, to the one given above.

```
before\message{One}\message{Two Three}\message{Four\textbackslash J}
\message{declare a thumb war.}After
```

This can happen: when \LaTeX{} is outputting the messages to the terminal, now the message with ‘One’ is shorter and it fits at the end of the output terminal line, and so \LaTeX{} breaks the line between it and the ‘Two Three’. That line break appears also in the log file. This line break insertion can depend on, for instance, the length of the full path names of included files. So producing finely-formatted lines in a way that is portable is hard, likely requiring starting your message at the beginning of a line.

### 27.5.3 \wlog

**Synopsis:**

```
\wlog{string}
```

Write `string` to the log file.

```
\wlog{Did you hear about the mathematician who hates negatives?}
\wlog{He'll stop at nothing to avoid them.}
```

Ordinarily `string` appears in a single separate line. Use `\textbackslash J` to insert a newline.

```
\wlog{Helvetica and Times Roman walk into a bar.}
\wlog{The barman says,\textbackslash JWe don't serve your type.}
```

### 27.5.4 \write18

**Synopsis:**

```
\write18{shell\_command}
```

Issue a command to the operating system shell. The operating system runs the command and \LaTeX{}’s execution is blocked until that finishes.

This sequence (on Unix)

```
\usepackage{graphicx} \% in preamble
...
\newcommand{\fignum}{1}
\immediate\write18{cd pix && asy figure\fignum}
\includegraphics{pix/figure\fignum.pdf}
```

will run Asymptote (the `asy` program) on `pix/figure1.asy`, so that the document can later read in the resulting graphic (see Section 21.3.1 `\includegraphics`, page 208). Like any `\write`, here \LaTeX{} expands macros in `shell\_command` so that `\fignum` is replaced by ‘1’.

Another example is that you can automatically run Bib\TeX{} at the start of each \LaTeX{} run (see Section 8.24.4 [Using BibTeX], page 93) by including `\immediate\write18{bibtex8 \jobname}` as the first line of the file. Note that `\jobname` expands to the basename of the root file unless the `--jobname` option is passed on the command line, in which case this is the option argument.
You sometimes need to do a multi-step process to get the information that you want. This will insert into the input a list of all PDF files in the current directory (but see texosquery below):

\immediate\write18{ls *.pdf > tmp.dat}
\input{tmp.dat}

The standard behavior of any \write is to wait until a page is being shipped out before expanding the macros or writing to the stream (see Section 27.5 \[\write\], page 254). But sometimes you want it done now. For this, use \immediate\write18{shell_command}.

There are obvious security issues with allowing system commands inside a \LaTeX file. If you download a file off the net and it contains commands to delete all your files then you would be unhappy. The standard settings in modern distributions turn off full shell access. You can turn it on, if you are sure the shell commands are safe, by compiling with \texttt{latex --enable-write18 filename} (see Section 28.1 [Command line options], page 259). (The \texttt{--shell-escape} option is a synonym, in \TeX Live.)

In the place of full shell access, modern distributions by default use a restricted version that allows some commands to work, such as those that run Metafont to generate missing fonts, even if you do not use the \texttt{enable-write18} option. By default this list of allowed commands is short and features only commands that are under the control of the distribution maintainers (see Section 28.1 [Command line options], page 259).

The \texttt{shell_command} text is always passed to /bin/sh on Unix-like operating systems, and the DOS command interpreter cmd.exe on Windows. Any different shell set by the user, and the SHELL environment variable, is ignored.

If what you need is system information, such as the operating system name, locale information, or directory contents, take a look at the texosquery package, which provides a convenient and secure interface for this, unlike the above examples using the raw \write18:

https://ctan.org/pkg/texosquery.

\LaTeX provides a package shellesc on top of the primitive \write18 command. Its primary purpose is to provide a command \texttt{\ShellEscape} which works identically on all \TeX engines; Lua\TeX intentionally did not retain \texttt{\write18} as a way to invoke a shell command, so some engine-specific code is needed. The shellesc package also provides a command \texttt{\DelayedShellEscape}, executed at \texttt{\output} time, for the same reason.
28 Command line interface

Synopsis (from a terminal command line):

\texttt{pdflatex options argument}

Run \LaTeX{} on \textit{argument}. In place of \texttt{pdflatex} you can also use (for PDF output) \texttt{xelatex} or \texttt{lualatex}, or (for DVI output) \texttt{latex} or \texttt{dvilualatex}, among others (see Section 2.3 \texttt{[\LaTeX{} engines]}, page 3).

For example, this will run \LaTeX{} on the file \texttt{thesis.tex}, creating the output \texttt{thesis.pdf}.

\texttt{pdflatex thesis}

Note that \texttt{.tex} is the default file name extension.

\texttt{pdf\TeX{}} is an extension of the original \TeX{} program, as are \texttt{Xe\TeX{}} and \texttt{Lua\TeX{}} (see Section 2.3 \texttt{[\TeX{} engines]}, page 3). The first two are completely backward compatible and the latter, almost so. Perhaps the most fundamental new feature for all three is that the original \TeX{} output its own DVI format, while the newer ones can output directly to PDF. This allows them to take advantage of the extra features in PDF such as hyperlinks, support for modern image formats such as JPG and PNG, and ubiquitous viewing programs. In short, if you run \texttt{pdflatex} or \texttt{xelatex} or \texttt{lualatex} then you will by default get PDF and have access to all its modern features. If you run \texttt{latex}, or \texttt{dvilualatex}, then you will get DVI. The description here assumes \texttt{pdflatex}.

See Section 28.1 \texttt{[Command line options]}, page 259, for a selection of the most useful command line options. As to \textit{argument}, the usual case is that it does not begin with a backslash, so the system takes it to be the name of a file and it compiles that file. If \textit{argument} begins with a backslash then the system will interpret it as a line of \LaTeX{} input, which can be used for special effects (see Section 28.2 \texttt{[Command line input]}, page 261).

If you gave no arguments or options then \texttt{pdflatex} prompts for input from the terminal. You can escape from this by entering \texttt{CTRL-D}.

If \texttt{\LaTeX{}} finds an error in your document then by default it stops and asks you about it. See Section 28.4 \texttt{[Recovering from errors]}, page 263, for an outline of what to do.

28.1 Command line options

These are the command-line options relevant to ordinary document authoring. For a full list, try running \texttt{`latex --help'} from the command line.

With many implementations you can specify command line options by prefixing them with `\texttt{--}' or `\texttt{--}'\texttt{--}'. This is the case for both \TeX{} Live (including Mac\TeX{}) and MiK\TeX{}. We will use both conventions interchangeably. If an option takes a value, it can be specified either as a separate argument (`\texttt{--foo val}'), or as one argument with an `\texttt{=}' sign (`\texttt{--foo=val}'), but there can be no spaces around the `\texttt{=}’. We will generally use the `\texttt{=}' syntax.

\texttt{-version} \ Show the current version, like `\texttt{pdf\TeX{} 3.14159265-2.6-1.40.16 (\TeX{} Live 2015/Debian)}’ along with a small amount of additional information, and exit.

\texttt{-help} \ Give a brief usage message that is useful as a prompt and exit.

\texttt{-interaction=mode} \ \LaTeX{} compiles a document in one of four interaction modes: \texttt{batchmode}, \texttt{nonstopmode}, \texttt{scrollmode}, \texttt{errorstopmode}. In \texttt{errorstopmode} (the default),
\TeX\ stops at each error and asks for user intervention. In \textit{batchmode} it
prints nothing on the terminal, errors are scrolled as if the user hit \texttt{RETURN}
at every error, and missing files cause the job to abort. In \textit{nonstopmode},
diagnostic message appear on the terminal but as in batch mode there is no
user interaction. In \textit{scrollmode}, \TeX\ stops for missing files or keyboard input,
but nothing else.

For instance, starting \LaTeX\ with this command line

\begin{verbatim}
pdflatex -interaction=batchmode filename
\end{verbatim}

eliminates most terminal output.

\texttt{-jobname=string}

Set the value of \TeX's \texttt{jobname} to the string. The log file and output file
will then be named \texttt{string.log} and \texttt{string.pdf}. see Section 28.3 \[Jobname\],
page 262.

\texttt{-output-directory=directory}

Write files in the directory \texttt{directory}. It must already exist. This applies to all
external files created by \TeX\ or \LaTeX, such as the \texttt{.log} file for the run, the
\texttt{.aux}, \texttt{.toc}, etc., files created by \LaTeX, as well as the main \texttt{.pdf}
or \texttt{.dvi} output file itself.

When specified, the output directory \texttt{directory} is also automatically checked
first for any file that it is input, so that the external files can be read back
in, if desired. The true current directory (in which \LaTeX was run) remains
unchanged, and is also checked for input files.

\texttt{--enable-write18}

\texttt{--disable-write18}

\texttt{--shell-escape}

\texttt{--no-shell-escape}

Enable or disable \texttt{\write18{shell\_command}} (see Section 27.5.4 \[\write18\],
page 257). The first two options are supported by both \TeX\ Live and MiK\TeX,\ while the second two are synonyms supported by \TeX\ Live.

Enabling this functionality has major security implications, since it allows a
\LaTeX file to run any command whatsoever. Thus, by default, unrestricted
\texttt{\write18} is not allowed. (The default for \TeX\ Live, Mac\TeX, and MiK\TeX\ is
to allow the execution of a limited number of \TeX-related programs, which they
distribute.)

For example, if you invoke \LaTeX\ with the option \texttt{no-shell-escape}, and in
your document you call \texttt{\write18{ls -1}}, then you do not get an error but the
log file says 'r\texttt{unsistem}(ls -1)...disabled'.

\texttt{-halt-on-error}

Stop processing at the first error.

\texttt{-file-line-error}

\texttt{-no-file-line-error}

Enable or disable \texttt{filename:lineno:er notoriousy messages. These are only
available with \TeX\ Live or Mac\TeX.}
28.2 Command line input

As part of the command line invocation

latex-engine options argument

you can specify arbitrary \TeX input by starting argument with a backslash. (All the engines support this.) This allows you to do some special effects.

For example, this file (which uses the hyperref package for hyperlinks) can produce two kinds of output, one to be read on physical paper and one to be read online.

\ifdef\paperversion % in preamble
\newcommand{\urlcolor}{black}
\else
\newcommand{\urlcolor}{blue}
\fi
\usepackage[colorlinks=true,\urlcolor=\urlcolor]{hyperref}
...
\href{https://www.ctan.org}{CTAN} % in body
...

Compiling this document book.tex with the command line pdflatex book will give the 'CTAN' link in blue. But compiling it with

pdflatex "\def\paperversion{}\input book.tex"

has the link in black. We use double quotes to prevent interpretation of the symbols by the command line shell. (This usually works on both Unix and Windows systems, but there are many peculiarities to shell quoting, so read your system documentation if need be.)

In a similar way, from the single file main.tex you can compile two different versions.

pdflatex -jobname=students "\def\student{}\input{main}"
pdflatex -jobname=teachers "\def\teachers{}\input{main}"

The jobname option is there because otherwise both files would be called main.pdf and the second would overwrite the first (see Section 28.3 [Jobname], page 262).

In this example we use the command line to select which parts of a document to include. For a book named mybook.tex and structured like this.

\documentclass{book}
\begin{document}
...
\include{chap1}
\include{chap2}
...
\end{document}

the command line

pdflatex "\includeonly{chap1}\input{mybook}"

will give output that has the first chapter but no other chapter. See Chapter 24 [Splitting the input], page 229.
28.3 Jobname

Running \LaTeX{} creates a number of files, including the main PDF (or DVI) output but also including others. These files are named with the so-called \jobname{}. The most common case is also the simplest, where for instance the command \texttt{pdflatex thesis} creates \texttt{thesis.pdf} and also \texttt{thesis.log} and \texttt{thesis.aux}. Here the job name is \texttt{thesis}.

In general, \LaTeX{} is invoked as \texttt{latex-engine \texttt{options argument}}, where \texttt{latex-engine} is \texttt{pdflatex}, \texttt{lualatex}, etc. (see Section 2.3 \texttt{[\LaTeX{} engines]}, page 3). If \texttt{argument} does not start with a backslash, as is the case above with \texttt{thesis}, then \TeX{} considers it to be the name of the file to input as the main document. This file is referred to as the \texttt{root file} (see Chapter 24 \texttt{[Splitting the input]}, page 229, and Section 24.3 \texttt{[\input]}, page 232). The name of that root file, without the .\texttt{tex} extension if any, is the \jobname{}. If \texttt{argument} does start with a backslash, or if \TeX{} is in interactive mode, then it waits for the first \texttt{\input} command, and the \jobname{} is the argument to \texttt{\input}.

There are two more possibilities for the \jobname{}. It can be directly specified with the \texttt{-jobname} option, as in \texttt{pdflatex -jobname=myname} (see Section 28.2 \texttt{[Command line input]}, page 261, for a practical example).

The final possibility is \texttt{texput}, which is the final fallback default if no other name is available to \TeX{}. That is, if no \texttt{-jobname} option was specified, and the compilation stops before any input file is met, then the log file will be named \texttt{texput.log}.

A special case of this is that in \LaTeX{} versions of (approximately) 2020 or later, the \jobname{} is also \texttt{texput} if the first \texttt{\input} occurs as a result of being called by either \texttt{\documentclass} or \texttt{\RequirePackage}. So this will produce a file named \texttt{texput.pdf}:

\begin{verbatim}
pdflatex "\documentclass{minimal}\begin{document}Hello!\end{document}"
\end{verbatim}

However, this special case only applies to those two commands. Thus, with

\begin{verbatim}
pdflatex "\documentclass{article}\usepackage{lipsum}\input{thesis}"
\end{verbatim}

the output file is \texttt{lipsum.pdf}, as \texttt{\usepackage} calls \texttt{\input}.

Within the document, the macro \texttt{\jobname{}} expands to the \jobname{}. (When you run \LaTeX{} on a file whose name contains spaces, the string returned by \texttt{\jobname{}} contains matching start and end quotes.) In the expansion of that macro, all characters are of catcode 12 (other) except that spaces are category 10, including letters that are normally catcode 11.

Because of this catcode situation, using the \jobname{} in a conditional can become complicated. One solution is to use the macro \texttt{\IfBeginWith} from the \texttt{xstring} package, which is insensitive to catcode. For example, in the following text the footnote “Including Respublica Bananensis Francorum.” is only present if the task name starts with \texttt{my-doc}.

\begin{verbatim}
If a democracy is just a regime where citizens vote then all banana republics \IfBeginWith\{my-doc\}{\footnote{Including Respublica Bananensis Francorum.}}%\end{verbatim}

Manipulating the value of \texttt{\jobname{}} inside of a document does not change the name of the output file or the log file.
28.4 Recovering from errors

If \LaTeX finds an error in your document then it gives you an error message and prompts you with a question mark, ?. For instance, running \LaTeX on this file

\begin{verbatim}
\newcommand{\NP}{\ensuremath{\textbf{NP}}} \\
The $\NP$ problem is a million dollar one.
\end{verbatim}

causes it show this, and wait for input.

! Undefined control sequence.
1.5 The $\NP$

{} problem is a million dollar one.

?

The simplest thing is to enter \texttt{x} and RETURN and fix the typo. You could instead enter ? and RETURN to see other options.

There are two other error scenarios. The first is that you forgot to include the \texttt{\end{document}} or misspelled it. In this case \LaTeX gives you a ‘*’ prompt. You can get back to the command line by typing \texttt{\textbackslash stop} and RETURN; this command does its best to exit \LaTeX immediately, whatever state it may be in.

The last scenario is that you mistyped the filename. For instance, instead of \texttt{pdflatex test} you might type \texttt{pdflatex tste}.

! I can't find file `tste'.

<*> tste

(Press Enter to retry, or Control-D to exit)

Please type another input file name:

The simplest thing is to enter \texttt{CTRL d} (holding the Control and d keys down at the same time), and then retype the correct command line.
Appendix A Document templates

Although illustrative material, perhaps these document templates will be useful. Additional template resources are listed at https://tug.org/interest.html#latextemplates.

A.1 beamer template

The `beamer` class creates presentation slides. It has a vast array of features, but here is a basic template:

```latex
\documentclass{beamer}
\title{Beamer Class template}
\author{Alex Author}
\date{July 31, 2020}
\begin{document}
\maketitle
% without [fragile], any \{verbatim\} code gets mysterious errors.
\begin{frame}[fragile]
  \frametitle{First Slide}
  \begin{verbatim}
  This is \verbatim!
  \end{verbatim}
\end{frame}
\end{document}
```

The Beamer package on CTAN: https://ctan.org/pkg/beamer.

A.2 article template

A simple template for an article.

```latex
\documentclass{article}
\title{Article Class Template}
\author{Alex Author}
\begin{document}
\maketitle
\section{First section}
Some text.
\subsection{First section, first subsection}
Additional text.
```

```
\section{Second section}
Some more text.
\end{document}

A.3 book template
This is a straightforward template for a book. See Section A.4 [Larger book template], page 265, for a more elaborate one.
\documentclass{book}
\title{Book Class Template}
\author{Alex Author}
\begin{document}
\maketitle
\chapter{First}
Some text.
\chapter{Second}
Some other text.
\section{A subtopic}
The end.
\end{document}

A.4 Larger book template
This is a somewhat elaborate template for a book. See Section A.3 [book template], page 265, for a simpler one.

This template uses \texttt{\frontmatter}, \texttt{\mainmatter}, and \texttt{\backmatter} to control the typography of the three main areas of a book (see Section 6.7 [\frontmatter & \mainmatter & \backmatter], page 43). The book has a bibliography and an index.

Also notable is that it uses \texttt{\include} and \texttt{\includeonly} (see Chapter 24 [Splitting the input], page 229). While you are working on a chapter you can comment out all the other chapter entries from the argument to \texttt{\includeonly}. That will speed up compilation without losing any information such as cross-references. (Material that does not need to come on a new page is brought in with \texttt{\input} instead of \texttt{\include}. You don’t get the cross-reference benefit with \texttt{\input}.)
\documentclass[titlepage]{book}
\usepackage{makeidx}\makeindex
\title{Book Class Template}
\author{Alex Author}
\includeonly{\%
% frontcover,
  preface,
  chap1,
% appenA,
}

\begin{document}
\frontmatter
\include{frontcover}
% maybe comment out while drafting:
\maketitle \input{dedication} \input{copyright}
\tableofcontents
\include{preface}

\mainmatter
\include{chap1}
...
\appendix
\include{appenA}
...

\backmatter
\bibliographystyle{apalike}
\addcontentsline{toc}{chapter}{Bibliography}
\bibliography
\addcontentsline{toc}{chapter}{Index}
\printindex

\include{backcover}
\end{document}
Index

#
# character, meaning of ........................................... 7

$
$ character, meaning of ........................................... 6
$ inline math ......................................................... 71
$$...$$ plain TeX display math ....................................... 57

%
% character, meaning of ........................................... 6

&
& (for table cells) .................................................. 86
& character, meaning of ........................................... 6

*
'*' prompt ............................................................ 263
'*', to \nocite all keys ............................................... 94
* form of environment commands .................................... 118
* form of sectioning commands ..................................... 36
*-form, defining new commands .................................... 112

--disable-write18 command-line option ......................... 260
--enable-write18 command-line option ......................... 260
--file-line-error command-line option .......................... 260
--halt-on-error command-line option ............................ 260
--help command-line option ........................................ 259
--interaction command-line option .............................. 259
--jobname command-line option ................................... 260
--no-file-line-error command-line option ...................... 260
--no-shell-escape command-line option .......................... 260
--output-directory command-line option ....................... 260
--shell-escape command-line option ............................. 260
--version command-line option ..................................... 259
-1, write stream number ......................................... 255

. aux file ............................................................... 3
. aux file and BibTeX commands .................................... 94
.dvi file ............................................................... 3
.fmt file ............................................................... 3
.glo file ............................................................... 245
.idx file ............................................................... 238, 242
.ind file ............................................................... 242
.isty file ............................................................. 242
.lof file ............................................................... 3, 233, 237
.log file ............................................................... 3
.lot file ............................................................... 3, 233, 237
.pdf file ............................................................... 3
.tex, default extension ............................................. 259
.toc file ............................................................... 3, 233, 237
.xdv file ............................................................... 4

/ \bin/sh, used by \write18 ......................................... 258

: \ (for math) .......................................................... 175

[ [... ] (for optional arguments) ................................... 7

^ ^ character, meaning of ........................................... 6
^ superscript .......................................................... 149
^ J, in \write ......................................................... 256

_ _ character, meaning of ........................................... 6
_ subscript ............................................................ 149

\_ character, meaning of ........................................... 6
\^ superscript .......................................................... 149
\^ J, in \write ......................................................... 256
\_ subscript ............................................................ 149

\_ character, meaning of ........................................... 6
\^ superscript .......................................................... 149
\^ J, in \write ......................................................... 256
\_ subscript ............................................................ 149

\_ character, meaning of ........................................... 6
\^ superscript .......................................................... 149
\^ J, in \write ......................................................... 256
\_ subscript ............................................................ 149
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>\	extbackslash{}DE (IE)</td>
<td>227</td>
</tr>
<tr>
<td>\int</td>
<td>155</td>
</tr>
<tr>
<td>\oldstylenums</td>
<td>22</td>
</tr>
<tr>
<td>\omega</td>
<td>155</td>
</tr>
<tr>
<td>\Omega</td>
<td>155</td>
</tr>
<tr>
<td>\ominus</td>
<td>155</td>
</tr>
<tr>
<td>\onecolumn</td>
<td>26</td>
</tr>
<tr>
<td>\openin</td>
<td>252</td>
</tr>
<tr>
<td>\opening</td>
<td>250</td>
</tr>
<tr>
<td>\opnout</td>
<td>252</td>
</tr>
<tr>
<td>\oplus</td>
<td>155</td>
</tr>
<tr>
<td>\OptionNotUsed</td>
<td>130</td>
</tr>
<tr>
<td>\oslash</td>
<td>155</td>
</tr>
<tr>
<td>\otimes</td>
<td>155</td>
</tr>
<tr>
<td>\oval</td>
<td>78</td>
</tr>
<tr>
<td>\overbrace{}(math)</td>
<td>170</td>
</tr>
<tr>
<td>\overleftarrow{}(math)</td>
<td>170</td>
</tr>
<tr>
<td>\overline{}(math)</td>
<td>170</td>
</tr>
<tr>
<td>\overrightarrow{}(math)</td>
<td>170</td>
</tr>
<tr>
<td>\owns</td>
<td>155</td>
</tr>
<tr>
<td>\PackageError</td>
<td>126</td>
</tr>
<tr>
<td>\PackageInfo</td>
<td>126</td>
</tr>
<tr>
<td>\PackageInfoNoLine</td>
<td>126</td>
</tr>
<tr>
<td>\PackageWarning</td>
<td>126</td>
</tr>
<tr>
<td>\PackageWarningNoLine</td>
<td>126</td>
</tr>
<tr>
<td>\pagbreak</td>
<td>106</td>
</tr>
<tr>
<td>\pagening</td>
<td>180</td>
</tr>
<tr>
<td>\pageref</td>
<td>49</td>
</tr>
<tr>
<td>\pagestyle</td>
<td>181</td>
</tr>
<tr>
<td>\paperheight</td>
<td>29</td>
</tr>
<tr>
<td>\paperwidth</td>
<td>29</td>
</tr>
<tr>
<td>\par</td>
<td>145</td>
</tr>
<tr>
<td>\paragraph</td>
<td>36, 41</td>
</tr>
<tr>
<td>\parallel</td>
<td>155</td>
</tr>
<tr>
<td>\parbox</td>
<td>199</td>
</tr>
<tr>
<td>\parindent</td>
<td>72, 146</td>
</tr>
<tr>
<td>\parisep</td>
<td>67</td>
</tr>
<tr>
<td>\parskip</td>
<td>146</td>
</tr>
<tr>
<td>\parskip example</td>
<td>64</td>
</tr>
<tr>
<td>\part</td>
<td>36, 37</td>
</tr>
<tr>
<td>\partial</td>
<td>155</td>
</tr>
<tr>
<td>\partopsep</td>
<td>67</td>
</tr>
<tr>
<td>\PassOptionsToClass</td>
<td>130</td>
</tr>
<tr>
<td>\PassOptionsToClass, ignoring</td>
<td>129</td>
</tr>
<tr>
<td>\PassOptionsToPackage</td>
<td>130</td>
</tr>
<tr>
<td>\pdfpageheight</td>
<td>11</td>
</tr>
<tr>
<td>\pdfpagewidth</td>
<td>11</td>
</tr>
<tr>
<td>\perp</td>
<td>155</td>
</tr>
<tr>
<td>\phantom</td>
<td>173</td>
</tr>
<tr>
<td>\phi</td>
<td>155</td>
</tr>
<tr>
<td>\Phi</td>
<td>155</td>
</tr>
<tr>
<td>\Pi</td>
<td>155</td>
</tr>
<tr>
<td>\pm</td>
<td>155</td>
</tr>
<tr>
<td>\pmod</td>
<td>169</td>
</tr>
<tr>
<td>\poptabs</td>
<td>84</td>
</tr>
<tr>
<td>\pounds</td>
<td>221</td>
</tr>
<tr>
<td>\Pr</td>
<td>169</td>
</tr>
<tr>
<td>\prec</td>
<td>156</td>
</tr>
<tr>
<td>\preceq</td>
<td>156</td>
</tr>
<tr>
<td>\prevdepth</td>
<td>31</td>
</tr>
<tr>
<td>\prime</td>
<td>156</td>
</tr>
<tr>
<td>\printhglossaries</td>
<td>244, 245</td>
</tr>
<tr>
<td>\printindex</td>
<td>244</td>
</tr>
<tr>
<td>\ProcessOptions</td>
<td>130</td>
</tr>
<tr>
<td>\prod</td>
<td>156</td>
</tr>
<tr>
<td>\protect</td>
<td>123</td>
</tr>
<tr>
<td>\protect, and message text</td>
<td>127</td>
</tr>
<tr>
<td>\Protectedfootnote</td>
<td>255</td>
</tr>
<tr>
<td>\ProvidesCommand</td>
<td>114</td>
</tr>
<tr>
<td>\providernobustcmd (etoolbox package)</td>
<td>128</td>
</tr>
<tr>
<td>\ProvidesClass</td>
<td>131</td>
</tr>
<tr>
<td>\ProvidesFile</td>
<td>132</td>
</tr>
<tr>
<td>\ProvidesPackage</td>
<td>131</td>
</tr>
<tr>
<td>\ProvideTextCommand</td>
<td>17</td>
</tr>
<tr>
<td>\ProvideTextCommandDefault</td>
<td>17</td>
</tr>
<tr>
<td>\qbezier</td>
<td>76</td>
</tr>
<tr>
<td>\quad</td>
<td>172, 184</td>
</tr>
<tr>
<td>\quad</td>
<td>172, 184</td>
</tr>
<tr>
<td>\quotedblbase (,)</td>
<td>221</td>
</tr>
<tr>
<td>\quotedblbase (.)</td>
<td>221</td>
</tr>
<tr>
<td>\r (ring accent)</td>
<td>225</td>
</tr>
<tr>
<td>\raggedbottom</td>
<td>28</td>
</tr>
<tr>
<td>\raggedleft</td>
<td>63</td>
</tr>
<tr>
<td>\raggedright</td>
<td>62</td>
</tr>
<tr>
<td>\raisebox</td>
<td>200</td>
</tr>
<tr>
<td>\rangle</td>
<td>156</td>
</tr>
<tr>
<td>\rbrace</td>
<td>156</td>
</tr>
<tr>
<td>\rbrack</td>
<td>156</td>
</tr>
<tr>
<td>\rceil</td>
<td>156</td>
</tr>
<tr>
<td>\Re</td>
<td>156</td>
</tr>
<tr>
<td>\read</td>
<td>253</td>
</tr>
<tr>
<td>\ref</td>
<td>49</td>
</tr>
<tr>
<td>\reflectbox</td>
<td>214</td>
</tr>
<tr>
<td>\refname</td>
<td>91</td>
</tr>
<tr>
<td>\restepcounter</td>
<td>135</td>
</tr>
<tr>
<td>\renewcommand</td>
<td>112</td>
</tr>
<tr>
<td>\renewenvironment</td>
<td>118</td>
</tr>
<tr>
<td>\renewrobustcmd (etoolbox package)</td>
<td>128</td>
</tr>
<tr>
<td>\RequirePackage</td>
<td>132</td>
</tr>
<tr>
<td>\RequirePackage, and texput jobname</td>
<td>262</td>
</tr>
<tr>
<td>\RequirePackageWithOptions</td>
<td>132</td>
</tr>
<tr>
<td>\resizebox</td>
<td>214</td>
</tr>
<tr>
<td>\restorec</td>
<td>100</td>
</tr>
<tr>
<td>\restriction</td>
<td>156</td>
</tr>
<tr>
<td>\reverserect</td>
<td>156</td>
</tr>
<tr>
<td>\reversecommand</td>
<td>147</td>
</tr>
<tr>
<td>\rfloor</td>
<td>156</td>
</tr>
<tr>
<td>\rhd</td>
<td>156</td>
</tr>
</tbody>
</table>
\varsigma ............................................. 159
\vartheta .......................................... 159
\vbox (plain TeX) ................................ 72
\vDash ............................................. 159
\vdots ............................................. 166
\vrule ............................................ 80
\vspace ........................................... 169
\verb .............................................. 97
\verb .............................................. 97
\verb .............................................. 97
\vtop plain TeX ................................... 72
\wedge .............................................. 159
\widehat .......................................... 169
\widetilde ......................................... 170
\wlog .............................................. 257
\wp ............................................... 160
\wr ............................................... 160
\write ............................................. 254
\write and security .............................. 256
\write streams 16, 17, 18 ......................... 255
\write18 .......................................... 257
\write18, enabling ................................ 260
\XeTeXdefaultencoding ......................... 227
\XeTeXinputencoding ............................ 227
\xii .............................................. 160
\Xi .................................................. 160
\xspace .......................................... 125
\year .............................................. 136
\zeta .............................................. 160

\{ 
  \{ character, meaning of \} ......................... 6
  \{ \ldots \} (for required arguments) ............ 7
\}

\} 

\} character, meaning of .......................... 6

\sim ............................................... 6

\sim character, meaning of ........................ 6

1

10pt option ..................................... 10
11pt option ..................................... 10
12pt option ..................................... 10

A

\texttt{a4paper} option ............................ 10
\texttt{a5paper} option ............................ 10
\texttt{abstract} environment .................. 52
\texttt{abstract} package ........................ 52
\texttt{abstracts} .................................. 52
\texttt{acccents} .................................. 224
\texttt{acccents, defining} ...................... 15, 16, 18, 19
\texttt{acccents, mathematical} ................. 169
\texttt{accessibility} .............................. 8
\texttt{accessing any character of a font} .... 220
\texttt{acronyms, list of} ......................... 244
\texttt{acute accent} .............................. 224
\texttt{acute accent, math} ....................... 169
\texttt{additional packages, loading} .......... 12
\texttt{adjustbox} package ........................ 197
\texttt{ae ligature} ................................ 226
\texttt{algorithm2e} package ................... 84
\texttt{align} environment, from \texttt{amsmath} 59
\texttt{aligning equations} ....................... 59
\texttt{alignment via tabbing} ................... 82
\texttt{alphabetic presentation forms Unicode block} 6
\texttt{ampersand character, meaning of} ...... 6
\texttt{amscd} package ............................ 161
\texttt{amsfonts} package ...................... 148, 160
\texttt{amsmath} package ........................ 53, 54, 56, 60, 95, 148, 163, 166, 167, 169, 171, 173, 175, 190
\texttt{amsmath package, replacing eqnarray} ... 59
\texttt{amsthm} package ........................... 95, 228
\texttt{answers} package .......................... 256
\texttt{appendices} ................................ 42
\texttt{appendix} ................................... 42
\texttt{appendix} package .......................... 43
\texttt{arguments, optional, defining and using} 112
\texttt{array} ....................................... 226
\texttt{array environment} ....................... 53
\texttt{array} package ................................ 54
\texttt{arrays, math} ................................ 53
\texttt{arrow, left, in text} ...................... 223
\texttt{arrow, right, in text} ..................... 224
\texttt{arrows} ..................................... 160
\texttt{article} class .............................. 10
\texttt{ascender height} ......................... 222
\texttt{ASCII circumflex, in text} ............... 221
\texttt{ASCII tilde, in text} ...................... 221
\texttt{astertisk}, centered, in text .......... 221
\texttt{Asymptote} package ...................... 77, 193, 198, 257
\texttt{at clause, in font definitions} .......... 122
\texttt{at-sign} .................................... 187
\texttt{atenddvii} package ....................... 126
\texttt{author}, for titlepage .................... 179
\texttt{auxiliary file} ............................ 3
<table>
<thead>
<tr>
<th>B</th>
<th>brev accent, math .................................................. 169</th>
</tr>
</thead>
<tbody>
<tr>
<td>babel option ...................................................... 10</td>
<td></td>
</tr>
<tr>
<td>back matter of a book ........................................... 43</td>
<td></td>
</tr>
<tr>
<td>background, colored ............................................... 218</td>
<td></td>
</tr>
<tr>
<td>backslash in text ................................................ 221</td>
<td></td>
</tr>
<tr>
<td>bar, double vertical, in text .................................... 222</td>
<td></td>
</tr>
<tr>
<td>bar, vertical, in text ............................................. 222</td>
<td></td>
</tr>
<tr>
<td>bar-over accent .................................................... 224</td>
<td></td>
</tr>
<tr>
<td>bar-over accent, math ............................................. 169</td>
<td></td>
</tr>
<tr>
<td>bar-under accent ................................................... 225</td>
<td></td>
</tr>
<tr>
<td>basics of \LaTeX .................................................. 2</td>
<td></td>
</tr>
<tr>
<td>batchmode .......................................................... 259</td>
<td></td>
</tr>
<tr>
<td>beamer package ..................................................... 264</td>
<td></td>
</tr>
<tr>
<td>beamer template and class ........................................ 264</td>
<td></td>
</tr>
<tr>
<td>bibliography format, open ........................................ 11</td>
<td></td>
</tr>
<tr>
<td>bibliography, creating (automatically) ......................... 93</td>
<td></td>
</tr>
<tr>
<td>bibliography, creating (manually) ............................... 91</td>
<td></td>
</tr>
<tr>
<td>Bib\LaTeX error messages ........................................ 94</td>
<td></td>
</tr>
<tr>
<td>big circle symbols, in text ....................................... 222</td>
<td></td>
</tr>
<tr>
<td>big point .......................................................... 138</td>
<td></td>
</tr>
<tr>
<td>bigfoot package .................................................... 111</td>
<td></td>
</tr>
<tr>
<td>black boxes, omitting ............................................. 11</td>
<td></td>
</tr>
<tr>
<td>blackboard bold ................................................... 162</td>
<td></td>
</tr>
<tr>
<td>blanks, after control sequences .................................. 113</td>
<td></td>
</tr>
<tr>
<td>bm package ........................................................ 161</td>
<td></td>
</tr>
<tr>
<td>bold math .......................................................... 22</td>
<td></td>
</tr>
<tr>
<td>bold typewriter, avoiding ........................................ 56</td>
<td></td>
</tr>
<tr>
<td>boldface mathematics ............................................... 161</td>
<td></td>
</tr>
<tr>
<td>book class .......................................................... 10</td>
<td></td>
</tr>
<tr>
<td>book, back matter ................................................ 43</td>
<td></td>
</tr>
<tr>
<td>book, end matter .................................................. 43</td>
<td></td>
</tr>
<tr>
<td>book, front matter ................................................ 43</td>
<td></td>
</tr>
<tr>
<td>book, main matter ................................................ 43</td>
<td></td>
</tr>
<tr>
<td>bottomnumber ....................................................... 34</td>
<td></td>
</tr>
<tr>
<td>bounding box ........................................................ 210</td>
<td></td>
</tr>
<tr>
<td>box ................................................................. 197</td>
<td></td>
</tr>
<tr>
<td>box, allocating new ............................................... 118</td>
<td></td>
</tr>
<tr>
<td>box, bounding ..................................................... 210</td>
<td></td>
</tr>
<tr>
<td>box, colored ...................................................... 218</td>
<td></td>
</tr>
<tr>
<td>box, save .......................................................... 201</td>
<td></td>
</tr>
<tr>
<td>box, use saved box ................................................ 202</td>
<td></td>
</tr>
<tr>
<td>boxes ............................................................... 197</td>
<td></td>
</tr>
<tr>
<td>bp ................................................................. 138</td>
<td></td>
</tr>
<tr>
<td>brace, left, in text ............................................... 222</td>
<td></td>
</tr>
<tr>
<td>brace, right, in text .............................................. 222</td>
<td></td>
</tr>
<tr>
<td>braces ............................................................ 162</td>
<td></td>
</tr>
<tr>
<td>braces, meaning of ................................................ 6</td>
<td></td>
</tr>
<tr>
<td>brackets .......................................................... 162</td>
<td></td>
</tr>
<tr>
<td>breaking lines ..................................................... 99</td>
<td></td>
</tr>
<tr>
<td>breaking pages ..................................................... 104</td>
<td></td>
</tr>
<tr>
<td>breaks, multiplication discretionary ............................ 175</td>
<td></td>
</tr>
<tr>
<td>ñ .......................... 11</td>
<td></td>
</tr>
</tbody>
</table>
# Index

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>248</td>
</tr>
<tr>
<td>221</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>138</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>258</td>
</tr>
<tr>
<td>96</td>
</tr>
<tr>
<td>175</td>
</tr>
<tr>
<td>215, 216, 218</td>
</tr>
<tr>
<td>215</td>
</tr>
<tr>
<td>216</td>
</tr>
<tr>
<td>215</td>
</tr>
<tr>
<td>216</td>
</tr>
<tr>
<td>218</td>
</tr>
<tr>
<td>218</td>
</tr>
<tr>
<td>216</td>
</tr>
<tr>
<td>259</td>
</tr>
<tr>
<td>125</td>
</tr>
<tr>
<td>112, 114</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>208</td>
</tr>
<tr>
<td>123, 125</td>
</tr>
<tr>
<td>112</td>
</tr>
<tr>
<td>257</td>
</tr>
<tr>
<td>115</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>149</td>
</tr>
<tr>
<td>96</td>
</tr>
<tr>
<td>205</td>
</tr>
<tr>
<td>130</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>114</td>
</tr>
<tr>
<td>114</td>
</tr>
<tr>
<td>114</td>
</tr>
<tr>
<td>220</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>133</td>
</tr>
<tr>
<td>117</td>
</tr>
<tr>
<td>134</td>
</tr>
<tr>
<td>133</td>
</tr>
<tr>
<td>135</td>
</tr>
<tr>
<td>96, 97, 123</td>
</tr>
<tr>
<td>73</td>
</tr>
<tr>
<td>84</td>
</tr>
<tr>
<td>180</td>
</tr>
<tr>
<td>48</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>49</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>49</td>
</tr>
<tr>
<td>183</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>162</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>253</td>
</tr>
<tr>
<td>129</td>
</tr>
<tr>
<td>180</td>
</tr>
<tr>
<td>228</td>
</tr>
<tr>
<td>228</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>54</td>
</tr>
<tr>
<td>138</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>127</td>
</tr>
<tr>
<td>216</td>
</tr>
<tr>
<td>112</td>
</tr>
<tr>
<td>118</td>
</tr>
<tr>
<td>122</td>
</tr>
<tr>
<td>120</td>
</tr>
<tr>
<td>112</td>
</tr>
<tr>
<td>255</td>
</tr>
<tr>
<td>243</td>
</tr>
<tr>
<td>243</td>
</tr>
<tr>
<td>243</td>
</tr>
<tr>
<td>243</td>
</tr>
<tr>
<td>243</td>
</tr>
<tr>
<td>162</td>
</tr>
<tr>
<td>163</td>
</tr>
<tr>
<td>246</td>
</tr>
<tr>
<td>55</td>
</tr>
<tr>
<td>55</td>
</tr>
<tr>
<td>122</td>
</tr>
<tr>
<td>138</td>
</tr>
<tr>
<td>224</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>137</td>
</tr>
<tr>
<td>258</td>
</tr>
<tr>
<td>175</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>177</td>
</tr>
<tr>
<td>174</td>
</tr>
<tr>
<td>82</td>
</tr>
<tr>
<td>82</td>
</tr>
<tr>
<td>56, 148</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>57</td>
</tr>
<tr>
<td>262</td>
</tr>
<tr>
<td>264</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>222</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>224</td>
</tr>
</tbody>
</table>
executepaper option .......................... 10
exp3 package ............................... 7, 220
exponent .................................... 149
expressions .................................. 142
extended Latin ............................... 226
external commands ............................ 257
external files, writing ....................... 61

F
f-ligatures ................................. 5
fallback jobname ............................ 262
families, of fonts ............................ 23
fancyhdr package ......................... 179, 181
fancyvrb package ........................... 84, 96
feminine ordinal symbol .................... 223
figure counter .............................. 133
figure dash character ....................... 223
figure environment ........................ 133
figure number, cross referencing ............ 49
figures, footnotes in ......................... 72
figures, inserting ........................... 60
file, closing .................................. 252
file, opening .................................. 252
file, reading .................................. 253
file, root ..................................... 229, 262
filecontents environment .................... 61
filecontents* environment ................... 61
filename for current job .................... 260
final option ................................ 11
first-latex-doc document ................... 1
fixed-width font ............................ 21
flafter package ............................. 33
fleqn option ................................ 11
float package ................................ 32
float page .................................... 32
flush left equations ......................... 11
flushing floats and starting a page .......... 104
flushleft environment ....................... 62
flushright environment ...................... 63
font catalogue ............................... 23
font commands, low-level .................... 23
font dimension, slant ....................... 190
font encoding ............................... 14, 16
font encoding, declaring .................... 15
font size ..................................... 25
font sizes ................................... 22
font styles ................................... 20
font symbols, by number .................... 220
fontenc ...................................... 14
fonts ........................................ 14
fonts, new commands for ................... 122
fonts, script ................................ 162
fontspec package ............................ 14, 161
footer style .................................. 181
footer, parameters for ...................... 28
footnote counter ............................ 133
footnote number, cross referencing ........... 49

force option for filecontents .............. 61
format files, \TeX ............................ 3
format, requiring ........................... 129
formulas, environment for .................. 59
formulas, math ............................... 148
forward reference ........................... 48
forward references, resolving ............... 3
fraction ...................................... 176
fragile commands ............................ 123
frame rule width ............................ 199
frame, line width ............................ 199
frame, separation from contents .......... 199
French quotation marks ..................... 221
front matter of a book ...................... 43
functions, math ............................. 168

G
gometry package ........................... 11
global options ................................ 10, 12, 131
glossaries ................................. 244
glossary ..................................... 244
glossary, entries ........................... 245
glossary, entry reference .................... 246
glue plain \TeX ............................... 137
glue register, plain \TeX ..................... 117
graphics ..................................... 204, 205, 208
graphics package ............................ 204, 205, 208
graphics package commands .................. 208
graphics package options .................... 204
graphics packages ........................... 77
graphics, resizing ........................... 214
graphics, scaling ............................ 214
graphpap package ........................... 77
grave accent ............................... 224
grave accent, math ......................... 169
greater than symbol, in text ............... 223
greek letters ............................... 149
Greek letters ............................... 167
grfext package ............................. 207
group, and environments .................... 52
group_skip .................................. 242
guillemots, birds ........................... 221
Index 281

H
hacek accent ........................................... 225
háček accent, math .................................... 169
Halmos symbol ........................................ 228
hard space ........................................... 189
hash character (#), meaning of ...................... 7
hat accent ............................................ 224
hat accent, math ..................................... 169
hat character, meaning of ......................... 6
header style ......................................... 181
header, parameters for ................................ 28
hello, world .......................................... 2
here, putting floats ................................... 32
\hbox{x} ................................................... 72
HNIT format .......................................... 4
H\TeX .................................................... 4
home page for manual .................................. 1
horizontal bar character ............................. 223
horizontal paragraph indentation .................. 146
horizontal space ...................................... 185
horizontal space, stretchable ...................... 185
horizontal spacing ................................... 173
hungarian umlaut accent ............................... 225
\hyperref{package} ..................................... 109, 180, 237, 261
hyphen character, non-breaking .................. 223
hyphenation, defining ............................... 103
hyphenation, discretionary ......................... 102
hyphenation, forcing ................................ 101
hyphenation, preventing ......................... 197
infinite horizontal stretch ........................ 185
infinite vertical stretch ............................ 195
inline formulas ....................................... 71
inner paragraph mode ................................ 177
input encodings ...................................... 5
input file ........................................... 229
input, on command line ................................ 261
input/output ......................................... 252
input/output, to terminal ......................... 252
\inputenc ............................................ 277
inserting figures ...................................... 60
insertions of special characters .................. 219
interface, command line ............................ 259
interline space ....................................... 30
internal vertical mode ............................... 177
inverted exclamation mark, as ligature .......... 5
inverted question mark, as ligature .............. 5
invisible character .................................. 173, 174
italic correction .................................... 190
italic font .......................................... 21
\item\_0 ................................................ 243
\item\_01 ............................................... 243
\item\_1 ............................................... 243
\item\_12 ............................................. 243
\item\_2 ............................................... 243
\item\_x1 ............................................. 243
\item\_x2 ............................................. 243
\itemize environment ................................... 63

J
jobname ............................................. 260, 262
JPEG files ........................................... 208
JPG files ............................................. 208
justification, ragged left ......................... 63
justification, ragged right ....................... 62

K
Knuth, Donald E ....................................... 2

L
label .................................................. 48
labelled lists, creating ............................ 55
Lamport \TeX ........................................... 2
Lamport, Leslie ....................................... 2
landscape option ..................................... 11
landscape orientation ................................ 11
latex .................................................. 3
latex command ....................................... 3
latex-dev ............................................ 4
latex-doc-ptr document ............................. 1
latex-lab package .................................. 8
\LaTeX\ format (.fmt) files ......................... 3
\LaTeX\ logo ......................................... 220
\LaTeX\ overview ..................................... 2
\LaTeX\ Project team ................................ 1
mathtools package ........................................... 148, 163, 171, 173
meaning of special characters .......................... 6
metadata, adding .......................................... 8
MetaPost package ......................................... 77
mfistuc package ........................................... 220
mhchem package ........................................... 149
millimeter .................................................. 138
minipage environment .................................... 71
minipage, creating a ..................................... 71
minted package ........................................... 84, 96
mirrors of CTAN ........................................... 9
mm .......................................................... 138
mobile output ............................................. 4
modes ...................................................... 177
monospace font .......................................... 21
moving arguments ........................................ 123
mpfootnote counter ..................................... 108, 133
mu .......................................................... 139
mu, math unit ............................................ 139
multicolumn text ......................................... 26
multilingual support ..................................... 224
multind package ......................................... 239
multiple indexes ......................................... 239
multiplication, discretionary ............................ 175

N
name .......................................................... 246
name of document root .................................... 262
NBSP ....................................................... 189
nested \include, not allowed ............................ 231
new class commands ...................................... 12
new command, checking definition of .................. 126
new command, definition ................................ 127
new commands, defining ................................ 112, 114
new line, output as input ................................ 100
new line, starting ........................................ 99
new line, starting (paragraph mode) ................... 101
new page, starting ....................................... 105
newline, in \write ........................................ 256
no-break space, Unicode U+00A0 ....................... 189
noheader option for filecontents ....................... 61
non-breaking hyphen character ......................... 223
non-English characters ................................ 226
non-sentence-ending punctuation ....................... 187
nonstopmode ............................................. 259
nosearch option for filecontents ....................... 61
notes in the margin ..................................... 147
notitlepage option ...................................... 11
null delimiter ............................................ 163
number sign character (#), meaning of .............. 7
numbered items, specifying counter ................... 134
numerals, old-style ..................................... 22

O
oblique font ............................................... 21
oe ligature .............................................. 227
ogonk ..................................................... 225
old-style numerals ....................................... 22
one-column output ....................................... 26
onecolumn option ....................................... 11
textwidth, calculating .................................. 11
opening quote ........................................... 221
openright option ....................................... 11
OpenType fonts .......................................... 3
operating system information ......................... 258
option clash ............................................ 130
option processing by default ......................... 127
option, currently being processed ................... 127
optional arguments, defining and using .............. 112
options, class ........................................... 127
options, color package ................................ 215
options, command line ................................ 259
options, document class ................................ 10, 12
options, global ......................................... 12
options, global and local ................................ 131
options, graphics package ............................. 204
options, list of unused ................................ 130
options, package ....................................... 12, 127
options, processing .................................... 130
ordinals, feminine and masculine ..................... 223
oslash ..................................................... 227
OT1 ....................................................... 14
OT1 encoding ........................................... 15
outer paragraph mode ................................... 177
output directory for all external files ............... 260
overbar accent .......................................... 224
overdot accent, math .................................. 169
overlining ............................................... 170
overview of LATEX ..................................... 2
overwrite option for filecontents .................... 61

P
package and class commands ......................... 125
package file layout .................................... 12
package options ....................................... 12, 127
package, abstract ...................................... 52
package, adjustbox .................................... 197
package, algorithm2e ................................ 84
package, amscd ......................................... 161
package, amsfonts ..................................... 148, 160
package, amsmath ..................................... 53, 54, 56, 60, 95, 148, 163, 166, 167, 169, 171, 173, 175, 190
package, amsthm ....................................... 95, 228
package, answers ...................................... 256
package, appendix ..................................... 43
package, array ......................................... 54
package, Asymptote ................................... 77, 193, 198, 257
<table>
<thead>
<tr>
<th>Package</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>atenddvi</td>
<td>126</td>
</tr>
<tr>
<td>babel</td>
<td>39, 91, 224, 228, 235, 241</td>
</tr>
<tr>
<td>beamer</td>
<td>264</td>
</tr>
<tr>
<td>bigfoot</td>
<td>111</td>
</tr>
<tr>
<td>bm</td>
<td>161</td>
</tr>
<tr>
<td>caption</td>
<td>35</td>
</tr>
<tr>
<td>cleveref</td>
<td>48, 50, 109</td>
</tr>
<tr>
<td>cm-super</td>
<td>15</td>
</tr>
<tr>
<td>comprehensive</td>
<td>149</td>
</tr>
<tr>
<td>cprotect</td>
<td>96, 97, 123</td>
</tr>
<tr>
<td>datatool</td>
<td>253</td>
</tr>
<tr>
<td>datetime</td>
<td>228</td>
</tr>
<tr>
<td>dcoulmn</td>
<td>54</td>
</tr>
<tr>
<td>enunitex</td>
<td>64, 69</td>
</tr>
<tr>
<td>envlab</td>
<td>250</td>
</tr>
<tr>
<td>etoolbox</td>
<td>128</td>
</tr>
<tr>
<td>eurosym</td>
<td>222</td>
</tr>
<tr>
<td>exp13</td>
<td>7, 220</td>
</tr>
<tr>
<td>fancyhdr</td>
<td>179, 181</td>
</tr>
<tr>
<td>fancyverb</td>
<td>84, 96</td>
</tr>
<tr>
<td>flafter</td>
<td>33</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
</tr>
<tr>
<td>fontspec</td>
<td>14, 161</td>
</tr>
<tr>
<td>geometry</td>
<td>11</td>
</tr>
<tr>
<td>graphpap</td>
<td>77</td>
</tr>
<tr>
<td>gffext</td>
<td>207</td>
</tr>
<tr>
<td>hyperref</td>
<td>109, 180, 237, 261</td>
</tr>
<tr>
<td>indentfirst</td>
<td>38, 40, 41, 42, 146</td>
</tr>
<tr>
<td>index</td>
<td>241</td>
</tr>
<tr>
<td>latex-doc</td>
<td>8</td>
</tr>
<tr>
<td>latex-sym</td>
<td>160</td>
</tr>
<tr>
<td>listings</td>
<td>84, 96</td>
</tr>
<tr>
<td>modern</td>
<td>15</td>
</tr>
<tr>
<td>luainputenc</td>
<td>227</td>
</tr>
<tr>
<td>macros2e</td>
<td>115</td>
</tr>
<tr>
<td>makeidx</td>
<td>244</td>
</tr>
<tr>
<td>mathhools</td>
<td>148, 163, 171, 173</td>
</tr>
<tr>
<td>MetaPost</td>
<td>77</td>
</tr>
<tr>
<td>mfirstruc</td>
<td>220</td>
</tr>
<tr>
<td>mhchem</td>
<td>149</td>
</tr>
<tr>
<td>minted</td>
<td>84, 96</td>
</tr>
<tr>
<td>multind</td>
<td>239</td>
</tr>
<tr>
<td>pict2e</td>
<td>77</td>
</tr>
<tr>
<td>polyglossia</td>
<td>224, 228, 235, 241</td>
</tr>
<tr>
<td>PSTricks</td>
<td>77</td>
</tr>
<tr>
<td>setspace</td>
<td>31</td>
</tr>
<tr>
<td>shellsc</td>
<td>238</td>
</tr>
<tr>
<td>showidx</td>
<td>239</td>
</tr>
<tr>
<td>siunitx</td>
<td>73, 189</td>
</tr>
<tr>
<td>suffix</td>
<td>116</td>
</tr>
<tr>
<td>tablefootnote</td>
<td>110</td>
</tr>
<tr>
<td>texosquery</td>
<td>258</td>
</tr>
<tr>
<td>textcase</td>
<td>220</td>
</tr>
<tr>
<td>textcomp</td>
<td>220</td>
</tr>
<tr>
<td>tikz-cd</td>
<td>161</td>
</tr>
<tr>
<td>TikZ</td>
<td>77, 193, 198</td>
</tr>
<tr>
<td>titlesec</td>
<td>36, 38, 39, 40, 41, 42</td>
</tr>
<tr>
<td>tocbibbind</td>
<td>234</td>
</tr>
<tr>
<td>tocloft</td>
<td>234, 237</td>
</tr>
<tr>
<td>ulem</td>
<td>170</td>
</tr>
<tr>
<td>unicode-math</td>
<td>162, 167, 168</td>
</tr>
<tr>
<td>url</td>
<td>97</td>
</tr>
<tr>
<td>verbatimbox</td>
<td>96</td>
</tr>
<tr>
<td>xparse</td>
<td>7, 116</td>
</tr>
<tr>
<td>xr</td>
<td>50</td>
</tr>
<tr>
<td>xr-hyper</td>
<td>50</td>
</tr>
<tr>
<td>xspace</td>
<td>125</td>
</tr>
<tr>
<td>xstring</td>
<td>262</td>
</tr>
<tr>
<td>packages, loading additional</td>
<td>12</td>
</tr>
<tr>
<td>page break, forcing</td>
<td>106</td>
</tr>
<tr>
<td>page break, preventing</td>
<td>106</td>
</tr>
<tr>
<td>page breaking</td>
<td>104</td>
</tr>
<tr>
<td>page_counter</td>
<td>133</td>
</tr>
<tr>
<td>page layout parameters</td>
<td>28</td>
</tr>
<tr>
<td>page number representation</td>
<td>183</td>
</tr>
<tr>
<td>page number, cross referencing</td>
<td>49</td>
</tr>
<tr>
<td>page numbering style</td>
<td>180</td>
</tr>
<tr>
<td>page style, this page</td>
<td>182</td>
</tr>
<tr>
<td>page styles</td>
<td>179</td>
</tr>
<tr>
<td>page, colored</td>
<td>218</td>
</tr>
<tr>
<td>page_precedence</td>
<td>244</td>
</tr>
<tr>
<td>paired delimiters</td>
<td>163</td>
</tr>
<tr>
<td>paragraph</td>
<td>36, 41</td>
</tr>
<tr>
<td>paragraph counter</td>
<td>133</td>
</tr>
<tr>
<td>paragraph indentation</td>
<td>146</td>
</tr>
<tr>
<td>paragraph indentation, in minipage</td>
<td>72</td>
</tr>
<tr>
<td>paragraph indentations in quoted text</td>
<td>82</td>
</tr>
<tr>
<td>paragraph indentations in quoted text, omitting</td>
<td>82</td>
</tr>
<tr>
<td>paragraph mode</td>
<td>177, 199</td>
</tr>
<tr>
<td>paragraph symbol</td>
<td>221</td>
</tr>
<tr>
<td>paragraph, ending</td>
<td>145</td>
</tr>
<tr>
<td>paragraph, in a box</td>
<td>199</td>
</tr>
<tr>
<td>paragraphs</td>
<td>144</td>
</tr>
<tr>
<td>parameters, for footnotes</td>
<td>107</td>
</tr>
<tr>
<td>parameters, page layout</td>
<td>28</td>
</tr>
<tr>
<td>parameters, substituting</td>
<td>113</td>
</tr>
<tr>
<td>parent directories, cannot write to</td>
<td>256</td>
</tr>
<tr>
<td>parentheses</td>
<td>162</td>
</tr>
<tr>
<td>parentheses and ends of sentences</td>
<td>187</td>
</tr>
<tr>
<td>part</td>
<td>36, 37</td>
</tr>
<tr>
<td>part_counter</td>
<td>133</td>
</tr>
<tr>
<td>pattern, current tab stops, tabbing</td>
<td>83</td>
</tr>
<tr>
<td>pc</td>
<td>138</td>
</tr>
<tr>
<td>PDF graphic files</td>
<td>208</td>
</tr>
<tr>
<td>PDF, tagged</td>
<td>8</td>
</tr>
<tr>
<td>pdflatex</td>
<td>3</td>
</tr>
<tr>
<td>pdflatex command</td>
<td>3</td>
</tr>
<tr>
<td>pdflatex-dev</td>
<td>4</td>
</tr>
<tr>
<td>pdflatex engine</td>
<td>3</td>
</tr>
<tr>
<td>percent character, meaning of</td>
<td>6</td>
</tr>
<tr>
<td>period, abbreviation-ending</td>
<td>187</td>
</tr>
<tr>
<td>period, centered, in text</td>
<td>223</td>
</tr>
<tr>
<td>period, sentence-ending</td>
<td>187</td>
</tr>
<tr>
<td>period, spacing after</td>
<td>187</td>
</tr>
<tr>
<td>pica</td>
<td>138</td>
</tr>
</tbody>
</table>
Index

pict2e package ........................................ 77
picture environment ................................. 73
pictures, creating .................................. 73
pinrow ................................................. 221
placement of floats ................................. 32
plaintext ............................................. 4
plaintext-dev ......................................... 4
plural ................................................ 246
PNG files ............................................. 208
poetry, an environment for ....................... 97
point .................................................. 138
polish ................................................. 226
polyglossia package .................................. 224, 228, 235, 241
portrait orientation ................................. 11
position, in picture .................................. 74
postamble ............................................. 242
PostScript point ...................................... 138
postscript, in letters ................................. 250
pounds symbol ....................................... 221
preamble .............................................. 242
preamble, defined ................................... 2
prerelase testing ..................................... 4
printing special characters ......................... 219
processing options .................................. 130
prompt, ‘*’ ............................................ 263
pronunciation ........................................ 2
PSTricks package ..................................... 77
pt ...................................................... 138
punctuation, sentence-ending ....................... 187

Q
quad .................................................. 172
question mark, upside-down ......................... 172
question marks, ending a sentence .................. 187
quotation environment ................................ 82
quotation marks, as ligatures ......................... 5
quotation marks, French ............................. 221
quote environment ................................... 82
quote, single straight .................................. 223
quote, straight base .................................. 223
quoted text with paragraph indentation,
  displaying ......................................... 82
quoted text without paragraph indentation,
  displaying ......................................... 82
quotes and ends of sentences ....................... 187

R
radical ................................................. 176
ragged left text ...................................... 63
ragged left text, environment for .................. 63
ragged right text ..................................... 62
ragged right text, environment for ................. 62
read a file .......................................... 253
redefining a command ................................ 112
redefining environments ............................ 118
reference, forward .................................... 48
references, resolving forward ....................... 3
registered symbol .................................... 224
relation, text above .................................. 176
release candidates .................................... 4
remarks in the margin ................................ 147
report class ......................................... 10
reporting bugs ....................................... 1
reserved characters, meaning of ................... 6
reserved characters, printing ....................... 219
resizing .............................................. 214
restricted shell access ............................... 258
right angle quotation marks ......................... 221
right arrow, in text .................................. 224
right brace, in text .................................. 222
right brace, meaning of .............................. 6
right parentheses/quotes, and spacing ............ 187
right quote .......................................... 221
right quote, double ................................... 223
right quote, single .................................... 223
right-hand equation numbers ....................... 11
right-justifying text ................................ 63
right-justifying text, environment for ............ 63
rigid lengths ......................................... 137
ring accent .......................................... 225
ring accent, math ..................................... 169
robust command, defining ......................... 127
robust commands ..................................... 133
roman font .......................................... 21
root file ............................................... 229, 262
roots .................................................. 176
rotating graphics .................................... 213
rotating text ......................................... 213
rotation ............................................. 213
row, tabbing ......................................... 83
rubber lengths ....................................... 137
rubber lengths, defining new ....................... 117
running header and footer ......................... 28
running header and footer style .................... 181

S
samepage environment .............................. 106
sans serif font ........................................ 21
scaled point ......................................... 138
scaling .............................................. 214
script fonts .......................................... 162
script letters for math ............................... 21
script style ......................................... 174
scriptscript style .................................... 174
scrollmode .......................................... 259
secnumdepth ......................................... 37
secnumdepth counter ................................ 37
section .............................................. 36, 39
section counter ...................................... 133
section number, cross referencing ................. 49
section numbers, printing ........................... 37
section symbol ...................................... 221
section, redefining .................................. 43
## W
- weights, of fonts ........................................ 24
- whatsit item ............................................. 255
- white space ............................................... 184
- wide hat accent, math ................................. 169
- wide tilde accent, math ............................... 170
- widths, of fonts ........................................ 24
- Wright, Joseph ......................................... 220
- writing external files ................................. 61
- writing letters ......................................... 247

## X
- x-height .................................................... 138
- xdvi command .......................................... 3
- xdvipdfmx ................................................ 4
- xelatex .................................................. 4
- xelatex-dev ............................................ 4
- XeLaTeX ................................................ 4
- xindex program ....................................... 244
- xindy program ......................................... 244
- xparse package ....................................... 7, 116
- xr package ............................................ 50
- xr-hyper package .................................... 50
- xspace package ....................................... 125
- xstring package ..................................... 262