

6 Layouts and fonts

This is the chapter that most users think they want first, because they come to structured documents from a wordprocessing environment where the *only* way to convey different types of information is to fiddle with the font and size drop-down menus.

As you will have seen by now, this is normally unnecessary in \LaTeX , which does most of the work for you automatically. However, there are occasions when you need to make manual typographic changes, and this chapter is about how to do them.

6.1 Changing layout

The design of the page can be a very subjective matter, and also a rather subtle one. Many organisations large and small pay considerable sums to designers to come up with page layouts to suit their purposes. Styles in page layouts change with the years, as do fashions in everything else, so what may have looked attractive in 1978 or 1991 may look rather dated in 2023.

As with most aspects of typography, making the document readable involves making it consistent, so the reader is not interrupted or distracted too much by apparently random changes in margins, widths, or placement of objects.¹ However, there are a

¹ Some authors — and perhaps some designers — believe that consistency is undesirable, and that double-page layouts in printed books should each be

number of different places where the layout usually *does* change, related to the frequency with which the format appears.

1. In books, the title page, the half-title, copyright and legal pages, dedication, acknowledgements, and other one-page preliminaries (if you use them) are usually designed individually, as the information on them only occurs once in that format anywhere in the document.
2. The Table of Contents and related lists like the List of Figures, List of Tables, List of Acronyms, Bibliography, References, Glossary, etc **SHOULD** all share one design with the preliminary sections like Preface, Introduction, and Foreword, which **SHOULD** be at section level, not chapter level (or in an article, at subsection level, not section level).
3. Chapter and Appendix start pages **SHOULD** always share a layout.
4. Other (normal) pages have a single layout, but within the page there **MAY** be individual variations to handle tables, lists, figures, sidebars, exercises, footnotes, etc.

The things that normally never change are the page size, margins, and body font size. There are very rare and exceptional circumstances when you can do this, but normally, once set, they stay fixed.

The exceptions to this are newspapers and magazines, where page layout is done individually, page by page (or pairs of facing pages together), but even here, most publications have strict rules about what blocks of material can be placed where, and use a carefully-designed set of templates to achieve this. While it would be technically possible to implement the huge range of page layouts needed by newspapers and magazines in \LaTeX , it would be impracticable to use them under the publication

designed independently. Kirschenbaum's magnificent *Goodbye Gutenberg* expresses this both eloquently and attractively, but the cost of such design labour and the cost of four-colour printing on all pages places it beyond the reach of most publishers' budgets until the economics of on-demand four-colour 'printing' makes it possible.

Layout conventions

Contrary to popular assumption (and contrary to \LaTeX 's defaults), navigation lists and any prelims and postlims (item above) SHOULD NOT be chapter-level headings but section-level. Only chapters and appendices should be at chapter-level in terms of layout and their rank in the Table of Contents.

deadline pressures in this field, as there is a constant need for modifications which would require a large number of \LaTeX -skilled programmers to implement.

If you are going to design a whole document yourself, it's probably a good idea to read a couple of books on layout design first, to get a feel for the conventions which contribute to making the reader comfortable reading.

While unusual or radical layouts have an important role in attention-grabbing, or in making a socio-political statement (*WIRED* magazine is an obvious example), they are usually out of place in business reports, white papers, books, theses, and journals. In ephemera, on the other hand, as in advertising, they are probably critical.

6.1.1 Margins and spacing

We mentioned in section 5.2 on page 137 and elsewhere the existence of the `geometry` package which lets you change margins. It also lets you set the text-area height and width and a lot of other layout settings: read the documentation for details (see section 3.1.3 on page 69 for how to read package documentation). Here is an example:

```
\usepackage[a4paper,left=2cm,top=1cm,bottom=2cm,
right=3cm,nohead,nofoot]{geometry}
```

Bear in mind when using the `geometry` package that you only need to specify some of *either* the margins *or* the text height/width.

Once it knows the paper size, if you give it the text width and the left-hand margin, for example, it can work out the right-hand margin. The package also provides the `\newgeometry` command, which lets you reset the margin settings in mid-document (at a page break, of course). This probably isn't something you want to do very often, though.

The spacing around the individual textual components (paragraphs, lists, footnotes, tables, figures, etc) can also be changed on a document-wide basis, as we saw with paragraph spacing and indentation in [section 2.7](#) on [page 63](#). There are a lot of packages available to do various aspects of this, far too many to go into detail here: search [CTAN](#) to find what you need.

Changing the spacing of section headings for the whole document can be done with the `sectsty` or `section` packages, designed to let you adjust section-head spacing without having to know about the internal L^AT_EX coding, which is quite complex.

The spacing for lists can be adjusted with the `enumitem` package. In both cases the user with highly specific requirements such as a publisher's Compositor's Specification should read the relevant sections in the *Companion* or ask for expert help, as there are many extra settings which can also be changed to fine-tune your design, but which need some understanding of L^AT_EX's internals.

All the above are for automating changes so that they occur every time in a consistent manner. You can also make manual changes:

Flexible vertical space: There are three commands `\smallskip`, `\medskip`, and `\bigskip`. These output flexible (dynamic, or 'rubber') space, approximately 3pt, 6pt, and 12pt high respectively, so they will automatically compress or expand a little, depending on the demands of the rest of the page (for example to allow one extra line to fit, or a heading to be moved to the next page without anyone except a typographer noticing the change). *These commands can only be used after a paragraph break* (a blank line or the command `\par`).

Fixed vertical space: For a fixed-height space which will *not* stretch or shrink, use the command `\vspace` followed by a

length in curly braces, eg `\vspace{18pt}` (again, this has to be after a paragraph break). Bear in mind that extra space which ends up at a page-break when the document is formatted *will get discarded entirely* to make the bottom and top lines fall in the correct places. To force a vertical space to remain and be taken into account even after a page break (very rare), use the starred variant, eg `\vspace*{19pt}`.

Double line-spacing: L^AT_EX's `\baselinestretch` value governs the amount of *extra* line-spacing based on the current font size (see section 6.2.5 on page 191). By default it is null, meaning no extra space. It is possible to set it to a *multiplier*, like `\renewcommand{\baselinestretch}{1.75}` to make it 1.75 times normal. However...

Double-spacing normal lines of text is usually A Bad Idea, as it looks very ugly, but increased line-spacing does become important if you are typesetting very wide lines, otherwise the reader's eye will not be able to pick up the start of a new line easily.

Double-spacing is still a requirement in many universities for thesis submission, partly because of the tendency of writers to use very wide lines on office-type paper sizes, and partly because the reviewers needed space to write in corrections. With the growth of electronic submission and editorial corrections in PDF files, it should become less necessary. Nowadays, 1⅓ or 1½ line spacing is considered acceptable, according to your font size.

Use the `setspace` package to do this. It has commands for double line-spacing (`\doublespacing`) and for one-and-a-half line spacing (`\onehalfspacing`): the `\singlespacing` command resets them). There is also a *spacing* environment to let you specify a different multiple as the argument:

```
\begin{spacing}{1.333}  
...  
\end{spacing}
```

Be aware that you may not want footnotes to be spaced by the same multiple as your normal text, and you may want other elements like lists, tables, figures, or quotations spaced differently.

As with theses, there are some perfectly genuine and normal reasons for wanting bigger line spacing, for example when typesetting a proof of a critical or variorum edition, where editors and contributors are going to want to add notes manually, or where the text is going to be overprinted by something else like Braille, or in advertising or display text for special effects.

Horizontal space: There is a horizontal equivalent to the `\vspace` command which works in the same way, so `\hspace{1in}` will insert a 1" space like this in mid-paragraph. There are also some predefined (shorter) spaces available:

- `\thinspace` ($\frac{1}{6}$ em), which we saw between single and double quotes in section 1.8 on page 25. It's also sometimes used between the full point after abbreviations and a following number, as in page references like p.42, where a word space would look too big, and setting it solid would look too tight.
- `\enspace` ($\frac{1}{2}$ em). There is no direct equivalent predefined in \LaTeX for 'mid' and 'thick' spaces as used by metal typesetters, although it would be possible to define them. The en as a unit is used as the width of a single digit in many fonts, as a convenience so that numbers in listings are easier to line up.
- `\quad` (1em) was originally the width of a capital M in metal type.
- `\qquad` (2em) is double a `\quad`.

Beyond this, all horizontal space within paragraphs is automatically flexible, as this is what \LaTeX uses to achieve justification. Never be tempted to try and change the spacing between letters unless you have some professional training in typography. Some systems use adjustable inter-letter

spacing (incorrectly called ‘tracking’) as an aid to justification and *it is almost always wrong to do so* (and looks it). While it is of course possible to change letterspacing in L^AT_EX (with the [soul](#) package), it should only be done by a typographer, and then only very rarely, as the settings are very subtle and beyond the scope of this book.²

6.1.2 Headers and footers

L^AT_EX has built-in settings to control the page style of its default page layouts, and space at the top and bottom of the page is provided automatically for them (it can also be adjusted or turned off in the [geometry](#) package). These settings are implemented with the `\pagestyle` command, which can take one of the following arguments in curly braces:

plain for a page number centered at the bottom — this is the default;

empty for nothing at all, not even a page number — use this when you are doing one-page documents like posters or handouts, where a page number has no meaning;

headings for running heads based on the current chapter and section — this is common for articles, books, and reports, so that every page is identifiable even if extracted or printed or copied separately;

myheadings lets you use your own [re]programmed definitions of how to use the `\markright` and `\markboth` commands, which control how chapter and section titles get into page headers.

The command `\thispagestyle` (taking the same arguments) can be used to force a specific style for the current page only.

However, the easiest way to get specialist running heads is to use the [fancyhdr](#) package, which lets you redefine the

² This does not apply for the German technique in blackletter type of using letter-spacing instead of (non-existent) italics. The defaults in the [soul](#) package were designed to cater for this.

left-hand, centre, and right-hand headers and footers for both odd-numbered (left-hand) and even-numbered (right-hand) pages (twelve objects in all).

Table 6.1 – Header and footer locations in the `fancyhdr` package

top left, even	top centre, even	top right, even	top left, odd	top centre, odd	top right, odd
LH page, even- numbered			RH page, odd-numbered		
bottom left, even	bottom centre, even	bottom right, even	bottom left, odd	bottom centre, odd	bottom right, odd

These areas can contain a page number, fixed text, variable text (like the current chapter or section title, or the catch-words of a dictionary), or even a small image. They can also be used to do page backgrounds and frames, by making one of them the top corner of an invisible box which ‘hangs’ text or images down over the whole page.

The settings for the typeset version of this document can be used as an example: for the whole story you have to read the documentation.

```
\pagestyle{fancy}\fancyhead{}
\renewcommand\headrulewidth{.1pt}
\fancyhead[LO,RE]{\footnotesize\sffamily\leftmark}
\fancyhead[LE,RO]{\footnotesize\sffamily\itshape
\rightmark}
\fancyfoot[C]{}
\fancyfoot[LE,RO]{\setlength{\fboxsep}{2pt}\ovalbox%
{\footnotesize\sffamily\thepage}}
\fancyfoot[LO,RE]{\footnotesize\sffamily\@title}
\fancypagestyle{plain}{\fancyhf{}}
\fancyfoot[R]{\setlength{\fboxsep}{2pt}\ovalbox{%
```



```
\footnotesize\sffamily\thepage}}  
\fancyfoot[L]{\footnotesize\sffamily\@title}  
\renewcommand{\headrulewidth}{0pt}}
```

This is probably more complex than most documents, but it illustrates some common requirements:

1. Settings are prefixed by making the `\pagestyle` 'fancy' and setting the `\fancyhead` to null to zap any predefined values.
2. The thickness of the rule at the top of the page can be changed (or set to 0pt to make it disappear).
3. The header and footer settings are specified with L, C, and R for left, centre, and right; and with O and E for Odd and Even numbered pages. In each setting, the typeface style, size, and font can be specified along with commands which implement various dynamic texts (here, the current chapter and section titles, which L^AT_EX stores in `\rightmark` and `\leftmark`).
4. The 'plain' variant is used for chapter starts, and resets some of the parameters accordingly.

6.1.3 List spacing

The different types of list are explained in [section 4.1 on page 88](#).

To change the format of lists, use the `enumitem` package as recommended in [section 4.1 on page 88](#). L^AT_EX's default list layouts are generously spaced, with wide indentation, and a blank line above and beneath and between items. They do, however, cope excellently with continuation paragraphs (additional paragraphs within an item), which many other systems confuse with unnumbered items.

1. A very common requirement is the unspaced or compact list: `enumitem` provides two: `noitemsep`, which removes the vertical white-space between items; and `nosep`, which additionally removes the vertical white-space above and below the list.

2. Bullets and numbering can be changed using the *label* option. Margin spacing can be changed to accommodate very wide or very narrow bullets or numbers.
3. Description lists can be restyled in a variety of ways. However, you should probably pick one way of formatting for the whole document, and not go changing it for every list.

All of the factors controlling the list shape can be reset, but you need to be careful that you don't make the list unreadable by closing up the spacing too much when the items are large (multi-line).

6.2 Using fonts

'Why do we need more fonts?' asked Bill Gates. 'We've got a serif, a sans, and a monospace font. Why do we need more?'

(Berry 2017)

The default typeface in L^AT_EX is **Computer Modern (CM)**. This typeface was created by Knuth for use with T_EX. It is based on a Victorian book typeface, Monotype Series 8, because he designed T_EX originally for typesetting books. Because it is one of the very few book typefaces with a comprehensive set of mathematical fonts, it has remained the default, rather than the variations on Times that you find in wordprocessors and other DTP systems (until recently the full set of mathematical symbols for Times were an expensive commercial add-on).

Computer Modern is based on a 19th-century book typeface from Monotype, which is why it looks a little like an old-fashioned school book. This paragraph is set in Computer Modern so you can see what it looks like. The typeface was designed using METAFONT, the font-drawing program made by Knuth to accompany T_EX systems, but it is now also available in Type 1 and TrueType formats.

The standard distribution of T_EX Live comes with about 130 OpenType and 75 TrueType typefaces (see [section 6.2.2.1 on page 174](#)). There are also some 300 Postscript Type 1 typefaces (many of these are PS versions of the OT and TT faces), and about

165 legacy METAFONT (Postscript Type 3) typefaces, to preserve compatibility with older documents which use them.

L^AT_EX can use more different types of font than any other system

- ☐ The original L^AT_EX could use any METAFONT font;
- ☐ pdfL^AT_EX could use any METAFONT or Postscript Type 1 font;
- ☐ X_YL^AT_EX and LuaL^AT_EX can use any METAFONT or Postscript Type 1 or TrueType or OpenType font.

6.2.1 First time only: setting up fonts

X_YL^AT_EX and LuaL^AT_EX let you use all your *system fonts* (those that came preinstalled with your computer and your non-T_EX software) as well as the ones that came with your T_EX distribution. There is one small piece of preparation to do, the very first time you use L^AT_EX: index them (properly speaking, cache them) for fast access. After that, you only need to re-index them if you buy or download a new font.

Without this indexing, you can still use your system fonts in L^AT_EX documents but you would have to type in where to find each font file every time, which is tedious.

There are four places where fonts are usually installed on Linux T_EX systems:

1. the system fonts directory (installed by your operating system, wordprocessor, and other ‘office’ software);
2. the T_EX distribution font directory (where the fonts go that came with T_EX);
3. the shared T_EX additional fonts directory (only relevant for multi-user shared systems; traditionally this is where system managers would put extra fonts for everyone to use);

4. the 'local' shared T_EX fonts directory (only relevant for multi-user shared systems: traditionally this is where authorised users would put extra fonts for everyone to use);
5. your **Personal T_EX Directory** fonts subdirectory (where you put fonts you buy or download for your own use).

The last one (your **Personal T_EX Directory**) is automatically searched by L^AT_EX and never needs indexing, so it is not in the examples below.

Each **fonts** directory will normally have subdirectories for the different types of font, eg **truetype**, **opentype**, **type1**, etc.

6.2.1.1 Indexing your fonts under Linux

This section covers three types of Linux installation:

1. T_EX Live installed from the **TUG DVD** or download on any type of Linux (see Exercise 26 on p.171);
2. Debian and its derivatives like Ubuntu that share the **.deb** repositories (see Exercise 27 on p.172);
3. Red Hat and its derivatives like CentOS that share the **.rpm** repositories (see Exercise 28 on p.173).

The principles are the same but the names of the directories differ slightly. If you have information on how this works on other types of distribution (eg Arch, SuSE, etc) please contact the author.

6.2.1.2 Indexing your fonts under Windows

T_EX Live installed from the **TUG DVD** or download on Windows systems needs no separate configuration, as the *fc-cache* program (included with T_EX Live) is run automatically after installation.

If you intend adding new fonts that you have bought or downloaded, you should create a **Personal T_EX Directory** with the appropriate subdirectories (eg **fonts/truetype** etc) and put the fonts in there. No indexing is then needed.

(Be aware that MiK_TE_X *does* require that you update MiK_TE_X's **File Name Database (FNDB)** when you add new fonts or personal (non-CTAN) packages: see 3. on page 274 for details.)

Exercise 26 – Font indexing in T_EX Live installed from TUG DVD or download on Linux systems

tug.org/texlive/doc/texlive-en/texlive-en.html#xetexfontconfig contains TUG's online details which you can check for updates.

1. Open a Command or Terminal window;
2. Become root by typing `sudo su -` and giving your password when asked;
3. Determine the location of your T_EX Live installation by typing

```
$ kpsewhich -var-value TEXMFSYSVAR
```

4. Using the value of the location of `$TEXMFSYSVAR` found in step above, copy the file at `TEXMFSYSVAR/fonts/conf/texlive-fontconfig.conf` to `/etc/fonts/conf.d/09-texlive.conf`;
5. Update the font cache:

```
$ fc-cache -fsv
```

(wait a few minutes while it caches your fonts).

6. Type `exit` to leave superuser mode;
7. Close the window if you want.

6.2.1.3 Indexing your fonts under Apple Mac OS X

The Apple Mac distribution of T_EX Live, MacT_EX, can already use the Mac systems fonts, but you need to add the T_EX Live fonts via the *FontBook* app.

Exercise 27 – Font indexing in T_EX Live installed from `.deb` repositories on Debian-based Linux systems

1. Open a Command or Terminal window;
2. Become root by typing `sudo su -` and giving your password when asked;
3. Open your favourite text editor (eg *emacs*, *vi*, *kate*, *gedit*, etc);
4. Create a new, empty file
`/etc/fonts/09-texlive.conf`;
5. Copy and paste this configuration into the file:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
  <!-- Fonts installed by Debian's tex* packages -->
  <dir>/usr/share/texlive/texmf-dist/fonts/opentype</dir>
  <dir>/usr/share/texlive/texmf-dist/fonts/truetype</dir>
  <dir>/usr/share/texlive/texmf-dist/fonts/type1</dir>
  <!-- Fonts installed on shared multiuser TeX systems -->
  <dir>/usr/share/texmf/fonts/opentype</dir>
  <dir>/usr/share/texmf/fonts/truetype</dir>
  <dir>/usr/share/texmf/fonts/type1</dir>
  <!-- Fonts pre-installed by the operating system -->
  <dir>/usr/share/fonts/opentype</dir>
  <dir>/usr/share/fonts/truetype</dir>
  <dir>/usr/share/fonts/type1</dir>
  <!-- Fonts added by users on some systems -->
  <dir>/usr/local/share/fonts/opentype</dir>
  <dir>/usr/local/share/fonts/truetype</dir>
  <dir>/usr/local/share/fonts/type1</dir>
</fontconfig>
```

6. Save the file;
7. Update the font cache:

```
fc-cache -fsv
```

(wait a few minutes while it caches your fonts).

8. Type `exit` to leave superuser mode;
9. Close the window if you want.

Exercise 28 – Font indexing in T_EX Live installed from `.rpm` repositories on RedHat-based Linux systems

1. Open a Command or Terminal window;
2. Become root by typing `sudo su` – and giving your password when asked;
3. Open your favourite text editor (eg *emacs*, *vi*, *kate*, *gedit*, etc);
4. Create a new, empty file in `/etc/fonts/` called `09-texlive.conf`;
5. Copy and paste this configuration into the file:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
  <dir>/usr/local/texlive/YYYY/texmf-dist/fonts/opentype</dir>
  <dir>/usr/local/texlive/YYYY/texmf-dist/fonts/truetype</dir>
  <dir>/usr/local/texlive/YYYY/texmf-dist/fonts/type1</dir>
  <dir>/usr/local/texlive/texmf-local/fonts/opentype</dir>
  <dir>/usr/local/texlive/texmf-local/fonts/truetype</dir>
  <dir>/usr/local/texlive/texmf-local/fonts/type1</dir>
  <dir>/usr/share/fonts</dir>
</fontconfig>
```

6. Replace `YYYY` with the year of your T_EX Live distribution;
7. Save the file;
8. Update the font cache:

```
fc-cache -fv
```

(wait a few minutes while it caches your fonts).

9. Type `exit` and close the window.

6.2.2 Set the default font family for a document

As explained in section 6.2 on page 168, Computer Modern is the built-in default typeface, so that's what you get if you don't specify anything else. There are three ways to specify other typefaces and individual fonts: *a*) by using a package; *b*) by font name; or *c*) by filename.

Using a package is more convenient, especially for whole typefaces, because the configuration of all the component fonts (eg roman, italic, bold, bold-italic, math, etc) has already been done by the package author, but font names or filenames let you specify your system (non- \TeX) fonts, which packages cannot do.

6.2.2.1 OpenType and TrueType typeface packages for $\text{X}_3\text{L}\text{A}\text{T}\text{E}\text{X}$ and $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$

The list below gives packages for about 40 **OpenType** (OT) and **TrueType** (TT) typefaces installed with a full distribution of TEX , and below that another 20 or so which can be downloaded from CTAN. Both sets are listed in pkgs.de/fontpackages.html.

(A few packages are not included here because they are not actually fonts in themselves, but 'enabling' packages which make specific combinations available for special purposes, such as the [hep-font](#) package for math combinations for the High Energy Physics community.)

Most of these packages support a *default* option, which sets them as the default font for the document, eg

```
\usepackage[default]{cabin}
```

will set the Cabin typeface as the default for the document. The [fontspec](#) package, which is required for using OT and TT fonts, is built into these packages and does not need to be specified separately.

OpenType and TrueType faces available at installation

(Samples are links to the package pages)

Alegreya
Alegreya

Sphinx of black quartz, judge my vow

almendra Almendra	Sphinx of black quartz, judge my vow
bitter Bitter	Sphinx of black quartz, judge my vow
cabin Cabin	Sphinx of black quartz, judge my vow
cantarell Cantarell	Sphinx of black quartz, judge my vow
Chivo Chivo	Sphinx of black quartz, judge my vow
cinzel Cinzel	SPHINX OF BLACK QUARTZ, JUDGE MY VOW
cochineal Cochineal	Sphinx of black quartz, judge my vow
coelacanth Coelacanth	Sphinx of black quartz, judge my vow
comfortaa Comfortaa	Sphinx of black quartz, judge my vow
crimson Crimson Test	Sphinx of black quartz, judge my vow
CrimsonPro Crimson Pro	Sphinx of black quartz, judge my vow
dejavu-otf DejaVu	Sphinx of black quartz, judge my vow
droidsans Droid Sans	Sphinx of black quartz, judge my vow
droidserif Droid Serif	Sphinx of black quartz, judge my vow

<code>ebgaramond</code> EB Garamond	Sphinx of black quartz, judge my vow
<code>forum</code> Forum	Sphinx of black quartz, judge my vow
<code>lato</code> Lato	Sphinx of black quartz, judge my vow
<code>libertine</code> Linux Libertine	Sphinx of black quartz, judge my vow
<code>libertinus-otf</code> Libertinus	Sphinx of black quartz, judge my vow
<code>LibreBodoni</code> Libre Bodoni	Sphinx of black quartz, judge my vow
<code>librecaslon</code> Libre Caslon Text	Sphinx of black quartz, judge my vow
<code>linguisticspro</code> Linguistics Pro	Sphinx of black quartz, judge my vow
<code>marcellus</code> Marcellus	Sphinx of black quartz, judge my vow
<code>merriweather</code> Merriweather	Sphinx of black quartz, judge my vow
<code>newpxtext</code> New PX Text	Sphinx of black quartz, judge my vow
<code>newtxtext</code> New TX Text	Sphinx of black quartz, judge my vow
<code>noto-serif</code> Noto	Sphinx of black quartz, judge my vow
<code>OldStandard</code> Old Standard	Sphinx of black quartz, judge my vow

opensans Open Sans	Sphinx of black quartz, judge my vow
quattrocento Quattrocento	Sphinx of black quartz, judge my vow
Rosario Rosario	Sphinx of black quartz, judge my vow
sourceserifpro Source Serif Pro	Sphinx of black quartz, judge my vow
TheanoDidot Theano Didot	Sphinx of black quartz, judge my vow
TheanoModern Theano Modern	Sphinx of black quartz, judge my vow
TheanoOldStyle Theano Old Style	Sphinx of black quartz, judge my vow
tinos Tinos	Sphinx of black quartz, judge my vow

These packages represent a selection of typefaces from suppliers like Google; donated T_EX sources like [CTAN](#); foundries like Impallari, [Summer Institute of Linguistics \(SIL\)](#), Ascender, and many others; and learned societies and individuals. Those in the list above are installed with T_EX Live; those in the list below can be downloaded and installed from [CTAN](#).

More OpenType and TrueType faces available from CTAN

(Samples are links to the package pages)

accanthis Accanthis ADF Std	Sphinx of black quartz, judge my vow
andika Andika	Sphinx of black quartz, judge my vow
caladea Caladea	Sphinx of black quartz, judge my vow

CharisSIL Charis SIL	Sphinx of black quartz, judge my vow
CormorantGaramond Cormorant Garamond	Sphinx of black quartz, judge my vow
fourier-otf Fourier OTF	Sphinx of black quartz, judge my vow
garamondlibre Garamond Libre	Sphinx of black quartz, judge my vow
gfsneohellenicot GFS Neo-Hellenic OT	Sphinx of black quartz, judge my vow
heros-otf T _E X Gyre Heros (Helvetica)	Sphinx of black quartz, judge my vow
ibarra Ibarra Real Nova	Sphinx of black quartz, judge my vow
imfellEnglish IM Fell English	Sphinx of black quartz, judge my vow
kpfonts-otf Kepler Project OT	Sphinx of black quartz, judge my vow
lexend Lexend	Sphinx of black quartz, judge my vow
librebaskerville Libre Baskerville	Sphinx of black quartz, judge my vow
noto Noto	Sphinx of black quartz, judge my vow
pagella-otf T _E X Gyre Pagella (Palatino)	Sphinx of black quartz, judge my vow

PlayfairDisplay Playfair Display	Sphinx of black quartz, judge my vow
plex-serif Plex Serif	Sphinx of black quartz, judge my vow
roboto Roboto	Sphinx of black quartz, judge my vow
schola-otf T _E X Gyre Schola (Century)	Sphinx of black quartz, judge my vow
spectral Spectral	Sphinx of black quartz, judge my vow
xcharter-otf Xcharter	Sphinx of black quartz, judge my vow

L^AT_EX includes versions of the popular (some would say overused) Adobe ‘35’ fonts which have been built into PDF readers, laser printers, printer drivers, and most DTP systems since the dawn of desktop publishing shortly after T_EX was written. These comprised eight text (Latin-alphabet) typefaces and two fonts of symbols or dingbats (35 fonts in total). They are now provided by non-Adobe versions known as the ‘T_EX Gyre’ collection, derived from the [Unternehmensberatung Rubow Weber \(URW\)](#) equivalents shown in the list below.

The replacements for the old Adobe ‘35’

(Samples are links to the package pages)

tgadventor URW Gothic L (ITC Avant Garde)	Sphinx of black quartz, judge my vow
tgbonum URW Bookman L (Bookman Old Style)	Sphinx of black quartz, judge my vow

`tgchorus`

URW Chancery L
Medium Italic (ITC
Zapf Chancery)

Sphinx of black quartz, judge my vow

`tgcursor`

URW Nimbus Mono
L (IBM Courier)

Sphinx of black quartz, judge my vow

`tgheros`

URW Nimbus Sans L
(Helvetica)

Sphinx of black quartz, judge my vow

`tgschola`

URW Century
Schoolbook L
(Century
Schoolbook)

Sphinx of black quartz, judge my vow

`tgpagella`

URW Palladio L
(Palatino)

Sphinx of black quartz, judge my vow

`tgtermes`

URW Nimbus
Roman Nog L (Times
New Roman)

Sphinx of black quartz, judge my vow

(If you need the old Microsoft Symbol font, it can be [downloaded](#), but Scott Pakin's [Comprehensive L^AT_EX Symbol List](#) is probably a better place to find symbols. The same applies to the Zapf Dingbats font, for which the [bbding](#) and [marvosym](#) packages provide alternatives. Jonathan Kew has posted [details](#) of how to access the actual Zapf Dingbats individually if required.)

The original Computer Modern typeface family was a METAFONT design by Donald Knuth, and was accompanied by a selection of other fonts, also made using METAFONT, some of which are in the list below

Some of the original fonts still in use

`cmr`

Computer Modern
Roman

Sphinx of black quartz, judge my vow

<code>cms</code> Computer Modern Sans-Serif	Sphinx of black quartz, judge my vow
<code>cmtt</code> Computer Modern Typewriter	Sphinx of black quartz, judge my vow
<code>panr</code> Pandora Roman	Sphinx of black quartz, judge my vow
<code>pss</code> Pandora Sans	Sphinx of black quartz, judge my vow
<code>pntt</code> Pandora Typewriter	Sphinx of black quartz, judge my vow
<code>uni</code> Universal	Sphinx of black quartz, judge my vow
<code>ccr</code> Concrete	Sphinx of black quartz, judge my vow
<code>eiad</code> Eiad	Ḣíl ḁon ṱĩntēḁn mḁr ṱo ṱĩntēḁn féin
<code>rust</code> Rustic	SPHINX OF BLACK QUARTZ, JUDGE MY VOW
<code>uncl</code> Uncial	sphinx of black quartz, judge my vow
<code>cdr</code> Dürer	SPHINX OF BLACK QUARTZ JUDGE MY VOW

The X Consortium donated a number of Latin-alphabet fonts in Postscript Type 1 format: have a look at Charter, Utopia,³ URW Antiqua Condensed, and URW Grotesk. There are hundreds of other fonts downloadable from CTAN: see Palle Jørgensen's comprehensive [L^AT_EX Font Catalogue](#) published by the Danish T_EX Users Group, categorised by type (serif, sans, monospace, decorative, etc) with samples and links to the packages.

³ The licence for Utopia does not allow it to be distributed automatically to users, but you can download it personally.

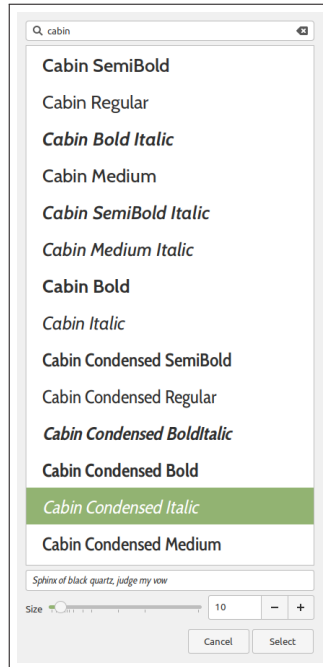
At any given time, T_EX expects there to be three typefaces available: `rm` for Roman (Serif), `sf` for Sans-Serif, and `tt` for Typewriter (Monospace).

Font family	Code
Roman (serif, with tails on the uprights), the default	<code>rm</code>
Sans-serif, with no tails on the uprights	<code>sf</code>
Monospace (fixed-width or typewriter)	<code>tt</code>

It is common to want to change all three defaults at the same time in order for new fonts to match each other. If a typeface provides Roman, Sans, and Monospace all in its own style, the package will change all three defaults automatically. Packages loading a single font family just load that one, so the others would remain Computer Modern. You can load another package for other fonts to replace them, or specify them individually as shown in the rest of this section.

6.2.2.2 OpenType and TrueType fonts and faces by fontname

The *fontname* of a font is the name that the designer declares is the name of the font family (like Cabin), or sometimes the name of the individual font (like Almendra-Bold). It is *not* the filename, although sometimes they happen to be the same. The fontname is the name you see in lists of fonts like the font dialog drop-down menu in editors.

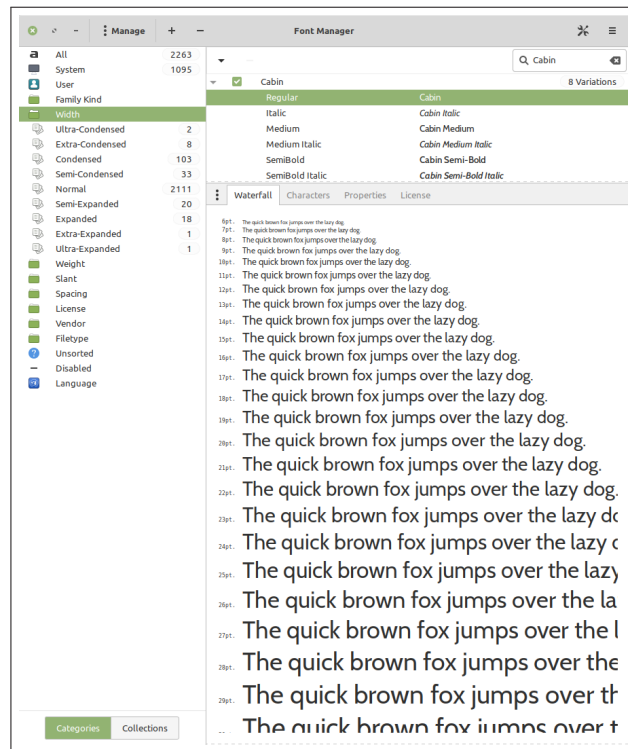


You need the `fontspec` package to use OT and TT faces and fonts by fontname. This package provides three commands to select font families: `\setmainfont` (for the roman or main face); `\setsansfont` (for the sans-serif face); and `\setmonofont` (for the typewriter or monospace face). These all take one compulsory argument: the fontname of the font family or typeface (we will see in section 6.2.2.3 on page 185 how to do this with filenames).

Most of the time that's all you need.

You can find the fontnames of any of your installed fonts by using your font browser or indexing command provided by your operating system.

CHAPTER 6. LAYOUTS AND FONTS



On Unix & GNU/Linux systems, and on Windows systems with T_EX Live installed, the command `fc-list` can be used to list your installed fonts, giving the filename (location), fontname (and synonyms), and the style, eg

```
$ fc-list Cabin
/usr/share/fonts/opentype/
cabin/Cabin-Bold.otf: Cabin:style=Bold
cabin/Cabin-BoldItalic.otf: Cabin:style=Bold Italic
cabin/Cabin-Italic.otf: Cabin:style=Italic
cabin/Cabin-Medium.otf: Cabin:style=Medium
cabin/Cabin-MediumItalic.otf: Cabin:style=Medium Italic
cabin/Cabin-Regular.otf: Cabin:style=Regular
cabin/Cabin-SemiBold.otf: Cabin:style=SemiBold
cabin/Cabin-SemiBoldItalic.otf: Cabin:style=SemiBold Italic
/usr/share/texlive/texmf-dist/fonts/opentype/
impallari/cabin/Cabin-Bold.otf: Cabin:style=Bold
impallari/cabin/Cabin-BoldItalic.otf: Cabin:style=Bold Italic
impallari/cabin/Cabin-Italic.otf: Cabin:style=Italic
impallari/cabin/Cabin-Medium.otf: Cabin:style=Medium
impallari/cabin/Cabin-MediumItalic.otf: Cabin:style=Medium Italic
impallari/cabin/Cabin-Regular.otf: Cabin:style=Regular
impallari/cabin/Cabin-SemiBold.otf: Cabin:style=SemiBold
impallari/cabin/Cabin-SemiBoldItalic.otf: Cabin:style=SemiBold Italic
```

From which you can see that, on my own system, I have Cabin installed twice: once by the operating system and once by T_EX Live. The important bit is between the first and second colons: the fontname **Cabin**, in this case a font family because there are no other values. Each entry gives the filename, the fontname, and any variants, separated by colons. If I do the same for Almendra, I get:

```
$ fc-list Almendra
/usr/share/texlive/texmf-dist/fonts/truetype/public/
  almendra/AlmendraSmallCaps.ttf: AlmendraSmallCaps:style=Regular
  almendra/Almendra-Bold.ttf: Almendra:style=Bold
  almendra/Almendra-Regular.ttf: Almendra:style=Regular
  almendra/Almendra-BoldItalic.ttf: Almendra:style=Bold Italic
  almendra/Almendra-Italic.ttf: Almendra:style=Italic
```

Some font entries have multiple values for the fontname, separated by commas, like **Alegreya, Alegreya Black**. In these cases the font family's fontname is the first one (before the first comma).

6.2.2.3 OpenType and TrueType fonts and faces by filename

Both TT and OT fonts are usually named in a pattern where the font family name (eg LiberationSerif, all one word) is followed by the variant (regular, italic, bold, bold-italic, etc) separated by a hyphen or underscore character, or sometimes by 'camel-casing', where descriptors starting mid-word use a capital letter. Here is the directory listing of my installation of the Liberation typeface:

```
$ ls -l /usr/share/fonts/truetype/liberation
total 2088
-rw-r--r-- 1 root root 118296 Feb 22 2020 LiberationMono-BoldItalic.ttf
-rw-r--r-- 1 root root 105460 Feb 22 2020 LiberationMono-Bold.ttf
-rw-r--r-- 1 root root 124012 Feb 22 2020 LiberationMono-Italic.ttf
-rw-r--r-- 1 root root 108172 Feb 22 2020 LiberationMono-Regular.ttf
-rw-r--r-- 1 root root 135124 Feb 22 2020 LiberationSans-BoldItalic.ttf
-rw-r--r-- 1 root root 137052 Feb 22 2020 LiberationSans-Bold.ttf
-rw-r--r-- 1 root root 162036 Feb 22 2020 LiberationSans-Italic.ttf
-rw-r--r-- 1 root root 128468 Feb 22 2020 LiberationSansNarrow-BoldItalic.ttf
-rw-r--r-- 1 root root 110252 Feb 22 2020 LiberationSansNarrow-Bold.ttf
-rw-r--r-- 1 root root 132452 Feb 22 2020 LiberationSansNarrow-Italic.ttf
-rw-r--r-- 1 root root 113028 Feb 22 2020 LiberationSansNarrow-Regular.ttf
-rw-r--r-- 1 root root 139512 Feb 22 2020 LiberationSans-Regular.ttf
-rw-r--r-- 1 root root 151452 Feb 22 2020 LiberationSerif-BoldItalic.ttf
-rw-r--r-- 1 root root 147132 Feb 22 2020 LiberationSerif-Bold.ttf
-rw-r--r-- 1 root root 145028 Feb 22 2020 LiberationSerif-Italic.ttf
-rw-r--r-- 1 root root 152408 Feb 22 2020 LiberationSerif-Regular.ttf
```

As you can see, there are four families of the Liberation typeface: Mono, Sans, Sans Narrow, and Serif. In each case there is a Regular (roman), Italic, Bold, and Bold Italic variant. The `fontspec` lets you load a font by specifying the filetype (extension) of the font file, where it is installed (the path), and how the font filenames fit the pattern:

```
\setmainfont{LiberationSerif}[Extension=.ttf,  
  Path=/usr/share/fonts/truetype/liberation/  
  UprightFont=*-Regular,  
  BoldFont=*-Bold,  
  ItalicFont=*-Italic,  
  BoldItalicFont=*-BoldItalic]
```

The pattern-matching is done by an asterisk which gets replaced by the font family name you give in the first argument, and the filetype (extension) is added to the end, so that `fontspec` can construct the whole filename. Repeating this for the sans and mono variants gives us the full set (the Narrow face is an exception and less commonly used):

```
\setsansfont{LiberationSans}[Extension=.ttf,  
  Path=/usr/share/fonts/truetype/liberation/  
  UprightFont=*-Regular,  
  BoldFont=*-Bold,  
  ItalicFont=*-Italic,  
  BoldItalicFont=*-BoldItalic]  
\setmonofont{LiberationMono}[Extension=.ttf,  
  Path=/usr/share/fonts/truetype/liberation/  
  UprightFont=*-Regular,  
  BoldFont=*-Bold,  
  ItalicFont=*-Italic,  
  BoldItalicFont=*-BoldItalic]
```

These three now automatically fill the L^AT_EX roles of the `rm`, `sf`, and `tt` families. The advantage of using filenames is that you do not have to rely on X_YL^AT_EX and LuaL^AT_EX accessing the fonts via the font index/cache system, as it goes straight to the filename in the directory you specify.

If you are mixing OT or TT fonts from different families, you can use the `Scale=MatchLowercase` option for the sans and mono

setups, making them scale to match the roman face, exactly as in Exercise 30 on p. 201tryfonts.

6.2.3 Changing the font-family temporarily

To use a different font [family] for a specific purpose, use the command `\newfontfamily`. This works exactly the same as the commands above for setting the main, sans, and mono font families but takes an extra parameter first, to specify the command you want it known by, so it does *not* replace the Roman, Sand, or Monospace defaults but is available *in addition* to them. To make the command `\tablesfont` invoke Liberation Sans Narrow, for example, you would use:

```
\newfontfamily{\tablesfont}{Liberation Sans Narrow}
```

If you want to use the filename, the format is:

```
\newfontfamily{\tablesfont}{LiberationSansNarrow}[
  Extension=.ttf,
  Path=/home/peter/texmf/fonts/truetype/liberation/,
  UprightFont=*-Regular,
  BoldFont=*-Bold,
  ItalicFont=*-Italic,
  BoldItalicFont=*-BoldItalic]
```

Then the new command (here, `\tablesfont`) can be used to switch to that typeface.

To load a solitary font (that is, not a whole family), there is a command `\newfontface`, which also works in the same way, by creating a new command to switch to it.

```
\newfontfamily{\headlinefont}{LobsterTwo-Bold}
[Extension=.otf,
 Path=/usr/share/fonts/opentype/lobstertwo/]
...
{\centering\headlinefont\fontsize{20}{24}\selectfont
 Lobster Rolls\dotfill \,$3.95\par}
```

Lobster Rolls.....\$3.95

These commands created by `\newfontfamily`, like the ones in Table 6.2 on the next page, are called ‘unscoped’ because they have global effect from that point on. In order to restrict the effect to a smaller scope (a few words, for example), you **MUST** put the command *and* the text inside a *group* (enclosed in curly braces as in the example, or within an environment), otherwise they will apply to the end of the document. See the panel ‘Grouping’ on p. 202 for more detail.

In a normal document, of course, arbitrary typeface changes like this are rare: people don’t (or at least, probably shouldn’t) randomly flip from one font to another. You select your default typefaces once, using packages or commands, at the start of the document, and stick with them — bold and italics are handled by the document class or stylesheet packages you use.

However, in advertising or magazines, a wide range of typefaces changes is common, but they are usually part of predefined styles for handling that type of formatting, built into the document class, so it is rare to have to do them manually.

Most cases where people want unusual typeface changes involve things like special symbols or effects on a repetitive basis, and L^AT_EX provides much easier (programmable) ways to make these changes into shorthand commands (called macros: see Chapter 7 starting on page 203).

This is jumping ahead a bit, but you could, for example, define a new macro called `\product` which would let you typeset product names in a distinct typeface (usually italics):

```
Andlinger, Inc., has replaced \product{Splosh} with  
\product{SuperSplosh}.
```

This is one of L^AT_EX’s most powerful features. It means that if you needed to change your `\product` command at some later stage to use a different font, you only have to change the font-family name in the macro, and you don’t need to edit your document text at all. What’s more, a macro could do other things at the same time, like add an entry to an index of products.

Vastly more common are changes to type *style*, while staying within the same font-family.

6.2.4 Changing type style

Within each typeface or font family there are usually several different ‘looks’ to the type design. L^AT_EX distinguishes mainly between **font shape** and **font series**. *Italics* is a shape (look carefully: the actual shape of the letters changes, as well as their slope); whereas **bold** is a series (same shapes, same slope, just thicker strokes).

Table 6.2 – Typeface styles, families, shapes, and series (unscoped)

Type style	Command	Example
Upright	<code>\upshape*</code>	The quick brown fox jumps over the lazy dog
Italic	<code>\itshape</code>	<i>Quick brown fox jumps over the lazy dog</i>
Slanted	<code>\slshape*</code>	<i>Quick brown fox jumps over the lazy dog</i>
Small Caps	<code>\scshape*</code>	QUICK BROWN FOX JUMPS OVER LAZY
Bold	<code>\bfseries*</code>	Quick brown fox jumps over the
Extended	<code>\bfseries†</code>	Quick brown fox jumps over the lazy
Sans-serif	<code>\sffamily</code>	The quick brown fox jumps over the lazy dog
Monospace	<code>\ttfamily</code>	quick brown fox jumps over the lazy

These commands are *unscoped*, so they must be used within a group.

* Not all typefaces have all variants! Some only have bold and italics.

† Some typefaces do not have both bold and bold extended: by default L^AT_EX uses `\bfseries` for bold extended.

The ‘shape’, ‘series’, and ‘family’ commands in above are **commutative** (each adds to the preceding form), so you can combine a shape with a series and/or a family, without the need to use `\selectfont`:

This is `{\bfseries bold \itshape italic \sffamily sans-serif}` type.

This is **bold italic sans-serif** type.

Beware of pushing your fonts beyond their limits unless you have typographic skills. It is not normally meaningful to combine one shape or series class with another of the same class, such as trying to get slanted-italics. It’s also sometimes impossible

to combine one family with another (such as seriffed sans-serif type!). Slanted plus italics, for example, doesn't make any sense, as italics are already slanted; and while some typefaces may well possess sans-serif italic small caps, they are not in common use.

If you really feel you need such combinations, try the `fontaxes`, which splits the 'shape' axis into a primary axis (upright, italic, slanted, upright italic, etc) and a secondary axis (small caps on or off). It redefines the `\itshape` and `\scshape` commands to combine instead of override each other. The `fontspec` package loads `fontaxes`. So do many legacy font packages. (Thanks to @Davislor on tex.stackexchange for this information.)

Sans-serif and monospace (typewriter) are not just different fonts, they are often different typeface families entirely.⁴

To avoid the problem of forgetting to put curly braces around the commands *and* text you want formatted, there is an alternative set of **scoped** commands for the most common type shape and series commands. These use curly braces in the 'argument' manner, so their effect applies only to the text in curly braces. These are the normal commands for changing the style of a word or phrase.

Table 6.3 – Typeface styles, families, shapes, and series (scoped)

Type style	Command	Example (using Computer Modern)
Italic	<code>\textit{text}</code>	puts <i>text</i> into italics
Slanted	<code>\textsl{text}</code>	puts <i>text</i> into slanted type*
Small Capitals	<code>\textsc{text}</code>	puts TEXT into small caps
Bold	<code>\textbf{text}</code>	puts text into bold type
Sans-serif	<code>\textsf{text}</code>	puts text into sans-serif type
Monospace	<code>\texttt{text}</code>	puts text into typewriter type

* If slanted is available separately from italics.

⁴ Although if you're a typographer wanting to experiment with typewriter typefaces with and without serifs, you can use METAFONT or *FontForgeto* to do exactly this kind of thing. But that's way outside the scope of this document.

These are commutative too, so you can nest them inside one another:

```
...\textbf{bold \itshape{italic \textsf{sans-serif}}}\ type...
```

What we know as ***underlining*** isn't a font: it was used in the days of typewriters where italics were not available, and it is extremely rare in typography except for specialist purposes. If you think you need it, use the `ulem` package with the *normalem* option, which provides a `\uline` command.

6.2.5 Font sizes

L^AT_EX has built into its defaults a set of predefined font size steps corresponding more or less to the traditional sizes available to metal typesetters. This is deliberate, as these sizes have grown up over 500 years of experience in printing as those which go best together for book-work, which is where T_EX originated.

These sizes are also reflected in the ***size steps*** at which Computer Modern was designed in the METAFONT program. It often comes as a surprise to new users that many typefaces are not designed as a single font and just scaled up or down, but specially drawn at different sizes to make them more legible.

As an example, here's 12pt Computer Modern, and here's 5pt Computer Modern scaled up to 12pt, and here's 17pt Computer Modern scaled down to 12pt so you can see there really is a significant difference.

Modern type formats have ***hinting*** parameters that allow scaling to implement the effects of design-sizes, but in general, you probably don't want to go scaling fonts too far beyond their design size because the spacing will start to look very odd.

- ☐ The default sizes (and the commands that operate them) are based on the use of a 10pt font, which is the default size for book work.
- ☐ Using the larger document class options (11pt and 12pt) will use 11pt and 12pt designs (explicit or hinted), with the other sizes (such as for headings) rescaled to match.

Table 6.4 – L^AT_EX font step sizes

Command	Example	Nominal point size	Exact point size
<code>\tiny</code>	The quick brown fox jumps over the lazy dog	5	5
<code>\scriptsize</code>	The quick brown fox jumps over the lazy dog	7	7
<code>\footnotesize</code>	The quick brown fox jumps over the lazy dog	8	8
<code>\small</code>	The quick brown fox jumps over the lazy dog	9	9
<code>\normalsize</code>	The quick brown fox jumps over the lazy dog	10	10
<code>\large</code>	The quick brown fox jumps over the lazy dog	12	12
<code>\Large</code>	The quick brown fox jumps over the lazy dog	14	14.40
<code>\LARGE</code>	The quick brown fox jumps over the lazy dog	18	17.28
<code>\huge</code>	The quick brown fox jumps over the lazy dog	20	20.74
<code>\Huge</code>	The quick brown fox jumps over the lazy dog	24	24.88

- Note that these are unscoped commands, so they should be used inside a group, either an environment or a set of curly braces terminated with a `\par` inside the closing brace. There are no scoped equivalents.
- Mathematics users should not confuse the text-mode `\scriptsize` command here with mathematics-mode `\scriptstyle`. [Thanks to Doug McKenna and David Carlisle on the T_EXhax mailing list.]

- The exact sizes used are listed in the macros in the Class Option files `size10.clo`, `size11.clo` and `size12.clo`.
- L^AT_EX's default fonts above 10pt are in fact scaled by a factor of 1.2, as shown in the fourth column of above.

While these shorthand commands relieve the beginner of having to worry about the appropriate point-size for a given task, if you need very specific sizes you can use the `\fontsize` command to specify exact sizes. This takes two arguments: the point size and the baseline distance. The example below gives you 22pt type on a 28pt baseline (ie with 6pt extra space or 'leading' between the lines).

```
{\fontsize{22}{28}\selectfont The example below gives you  
22pt type on a 28pt baseline (ie with 6pt extra space or  
'leading' between the lines).\par}
```

The example below gives you 22pt type on a 28pt baseline (ie with 6pt extra space or 'leading' between the lines).

The term 'leading' comes from the old metal-type practice of adding a strip of typemetal between the lines, or casting the type on a deeper body, to increase the line spacing, so it's pronounced 'ledding' after the metal.

If you are using *pdf_latex* or the original *latex* processor, you will need to use the `fix-cm` package to override the step sizes. This needs special placement: it **MUST** come at the start of the document, *even before* the `\documentclass` command, and **MUST** be invoked with the `\RequirePackage` command normally used only by document class designers:

```
\RequirePackage{fix-cm}
```

```
\documentclass{article}  
...
```

6.2.6 Logical markup

All this playing around with fonts is very pretty but you normally only do it for a reason, even if that reason is just to be decorative. Italics, for example, are used for many things:

Cause	Effect
Foreign words	<i>ex officio</i>
Scientific names	<i>Ranunculus ficaria</i>
Emphasis	<i>must not</i>
Titles of documents	<i>Accounting in Business</i>
Product names	<i>Corel WordPerfect</i>
Variables in maths	$E = mc^2$
Subtitles or headings	<i>42. How to get started</i>
Use of a letter as a word	Who knocked the <i>L</i> out of London?
Decoration	<i>FREE UPGRADE!!!</i>

Humans usually have no problem telling the difference between these reasons, because they can read and understand the meaning and context, and we've been exposed to many of these meanings since we started to read. Computers cannot (yet) do this reliably, so it has become conventional to use descriptive names which make the distinction explicit, even though the appearance may be the same.

L^AT_EX has some of these built in, like `\emph`, which provides *emphasis*. This has a special feature because *when the surrounding text is already italic, emphasis automatically reverts to upright type*, which is the normal practice in typesetting.

```
This has a special feature because {\itshape when  
the surrounding text is already italic,  
\emph{emphasis} automatically reverts to  
\emph{upright type}}, which is the normal practice  
in typesetting.
```

This sensitivity to logic is programmed into the definition of `\emph` and it's not hard to make up other commands of your own which could do the same, such as `\foreign` or `\product`.

But why would you bother? In a short document it's probably not important, but if you're writing a long report, or a formal document like an article, a book, or a thesis, it makes writing and editing hugely easier if you can control whole groups of special effects with a single command, such as italicising, indexing, or cross-referencing to a glossary. If a format needs changing, you only have to change the definition, and every occurrence automatically follows suit.

It also makes it possible to find and act on groups of meanings — such as making an index of scientific names, or retrieving all product names — if they are identified as such. Otherwise you'd spend weeks hunting manually through every `\textit` command to find the ones you wanted. This is the bottom line of automation: it can save you time and money.

In Chapter 7 starting on page 203 we will see how to make your own simple commands to do things like this.

6.2.7 Colour

You can typeset anything in L^AT_EX in any colour you want using the `xcolor` package. Adding the command `\usepackage{xcolor}` to your Preamble (note the US spelling of color) makes available a default palette of primary colours: `red`, `green`, and `blue` for the RGB colour model used for emitted light (computer and television screens), and `cyan`, `magenta`, `yellow`, and `black` for the CMYK colour model used for reflected light (printing).

For the occasional word or phrase in colour, use the command `\textcolor` with two arguments, the colour name and the text: `\textcolor{red}{like this}` to get red `like this`. There is an unscoped `\color` command as well, for use within groups:

```
...\color{blue}some text in blue}...
```

There are several package options for additional colours: two popular ones are *dvipsnames*, which provides a 64-colour palette of predefined colour names exactly matching the big box of

64 *Crayola*[™] colouring pencils much favoured by artists and designers; and *svgnames*, which provides the 256 colours defined in the specification for the [Scalable Vector Graphics \(SVG\)](#) drawing and diagramming language (which includes the 64 colours of *dvipsnames*). There are others too: see the documentation for the *xcolor* package.

If you want the *Crayola* colour **Crimson**, and you have loaded *xcolor* with the *svgnames* or *dvipsnames* option, you can use it as a colour name (colour names are case-sensitive):

```
\color{Crimson}some red text}
\textcolor{Crimson}{some red text}
```

As some of the predefined colour names are quite long, you can create a short name of your own for colours you use frequently, using the `\definecolor` command:

```
\definecolor{mb}{named}{MidnightBlue}
```

The `\definecolor` command needs three arguments: your shorthand name, the name of the colour model, and the colour specification. In the case of the *named* model, the last argument is one of the colour names specified by the *named* option you loaded the package with.

Using the `\definecolor` command, you can also define any colour you want by giving it a name, specifying which colour model, and providing the [Red-Green-Blue \(RGB\)](#) or [Cyan-Magenta-Yellow-Black \(CMYK\)](#) colour values *expressed as decimal fractions of 255, separated by commas*. For example, an RGB colour given as (37,125,224) in decimal integer form can be given as:

```
\definecolor{midblue}{rgb}{0.145,0.490,0.882}
```

To get the fractional value, divide the integer value by 255, the maximum for each of the hues in the Red-Green-Blue colour model. You can then use `\textcolor` with your new colour name: **midblue looks like this** if you're reading in colour. Alternatively, use the [HTML hexadecimal colour model](#), the same as used in web pages and [CSS stylesheets](#):

```
\definecolor{midblue}{HTML}{250FE0}
```

The `xcolor` package also provides two colour versions of `\fbox` (see section 4.6.2 on page 127) called `\colorbox` and `\fcolorbox` which create a box with a coloured background:

```
\colorbox{midblue}{\color{magenta}Magenta on midblue}
```

The material in the second argument can have its own text colour, as in the example. The `\fcolorbox` has an extra first argument to specify the colour of the frame or border placed around the box. The border width is controlled by the `\fboxrule` setting and the separation between rule and content is controlled by the `\fboxsep` setting as we already saw in section 4.6.2 on page 127.

However, combining colours is an art and a skill: using the command above to get the effect `magenta on midblue` illustrates why it is important to learn about colour models and palettes before trying to use them!

6.3 The L^AT_EX font catalogue

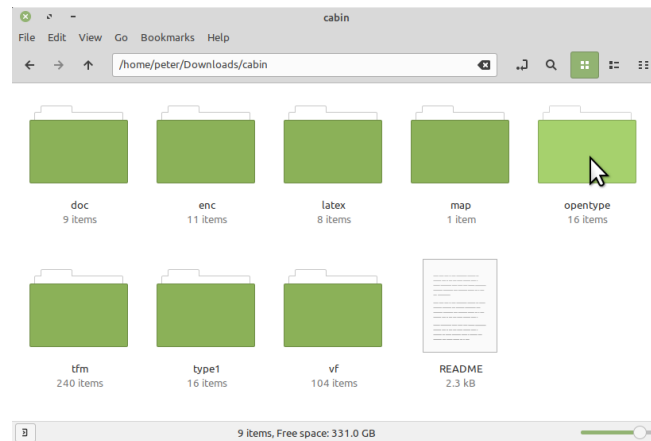
The L^AT_EX Font Catalog is a web site created and maintained by Palle Jørgensen at www.tug.dk/FontCatalogue/. It lists over 200 typefaces for use with L^AT_EX, many of them available nowhere else, with samples and links to the directories on CTAN where you can download them. You can spend many fascinating hours downloading and installing them and trying them out in your documents.

Installing a font from the Font Catalogue automatically

Over 100 of the typefaces in the L^AT_EX Font Catalog are prebuilt for L^AT_EX using the T_EX Directory Structure (TDS)

1. Download the `.tds.zip` file for the typeface you want from mirrors.ctan.org/tex/install/fonts/;
2. Unzip it directly into your Personal T_EX Directory.

Figure 6.1 – Layout of a font zip file downloaded from CTAN



If no TDS zip file is provided (check on the typeface’s web page on CTAN at www.ctan.org/tex-archive/fonts), you will need to use the plain zip file and move the subdirectories into The Right Places yourself.

Installing a font from the Font Catalogue manually

1. Download the zip file from the link **download the contents of this package in one zip archive** on the typeface’s CTAN web page, which is either at the bottom or the right-hand side, depending on the width of your screen.
2. Open the zip file in your directory browser (see row 2 in Table 1 on p. xxxhandlingfiles). Inside the zip file there will be several subdirectories, shown in below

You will want three folders: **doc**, **latex**, and *either* the **truetype** or the **opentype**, whichever one is there.

3. Open the **doc** zip directory and check the **README** or **AUTHORS** file to identify the company or individual responsible for the typeface. This is commonly called the ‘foundry’, following the habits of the old hot-metal type era. It should be one word, or an acronym.

4. In your **Personal T_EX Directory**, create places to put these, replacing foundry and typeface names with meaningful one-word values:

Zip file directory	TDS directory
doc	fonts/doc/ <u>foundry</u> / <u>typeface</u>
latex	tex/latex/ <u>typeface</u>
opentype ¹	fonts/opentype/ <u>foundry</u> / <u>typeface</u>
truetype ¹	fonts/truetype/ <u>foundry</u> / <u>typeface</u>






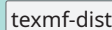

¹ Font zip files normally provide *either* a **truetype** folder *or* an **opentype** folder. If both, pick one.

5. Extract the contents of each of the three zip folders in turn into the folders you have created in your **Personal T_EX Directory** according to the table in step above.

The foundry name is not essential, but it helps to use a separate folder because it identifies where the fonts originated, which can help with choosing fonts.

If there was a package (**.sty**) file in the **latex** zip directory, you can use it in your documents, otherwise load the font[s] you want as shown in **section 6.2.2.2** on page 182.

Exercise 29 – Font indexing in MacTeX on Apple Mac OS X systems

1. Open the Finder;
2. Click on the Macintosh HD entry;
3. Press  +  +  (shift splat dot together) to reveal system folders;
4. Navigate to `/usr/local/texlive/YYYY/texmf-dist` (replacing YYYY with the year of your MacTeX distribution);
5. In the menus, click  Add to Sidebar;
6. Open FontBook;
7. In FontBook, click  to add fonts;
8. In the Finder panel that opens, click the  folder in the sidebar and navigate to the `fonts/truetype` subfolder;
9. Click ;
10. If the system prompts for your password, type it in;
11. Wait. If a window opens saying some fonts fail validation, DO NOT install them^a
12. Repeat from step above for `fonts/opentype`;
13. When done, browse FontBook for fonts available to X_YLaTeX and LuaLaTeX.

^a In some cases, FontBook may believe them to have missing tables, and in others, they may duplicate existing Mac system fonts, and FontBook's behaviour seems to be to use the TeX Live fonts in preference to its own.

Exercise 30 – Try setting up fonts by fontname

1. Open a new L^AT_EX file in your editor (pick a blank or empty one if your editor offers templates);
2. Copy and paste this text into the file:

```
\documentclass[12pt]{article}
\usepackage{fontspec}
\setmainfont{Crimson Pro}
\setsansfont{Cabin Regular}
\setmonofont{TeX Gyre Cursor}
\AtBeginDocument{\LARGE}
\begin{document}
This is the main (default) font

\sffamily This is the sans-serif font

\ttfamily This is the monospace font
\end{document}
```

3. Process the document and examine the PDF. You may notice that the sans-serif font (Cabin) and the monospace font (Cursor) looks larger than the main font (Crimson), even though they are all set to the `\LARGE` size (about 18pt, see Table 6.4 on page 192). This is because fonts are designed with different heights to the lowercase and uppercase characters.
4. Compensate for this by adding the option `Scale=MatchLowercase` to the sans and mono commands:

```
\setsansfont{Cabin Regular}[Scale=MatchLowercase]
\setmonofont{TeX Gyre Cursor}[Scale=MatchLowercase]
```

5. Reprocess and see that the sans and mono fonts have now been loaded at a size which matches the main font.

There are packages for the T_EX Gyre fonts, like `tgcursor` (they all start with `tg`), have a `matchlowercase` package option which does this scaling.

Grouping

The use of curly braces to restrict the scope of a typographic change is called a T_EX **group**. This is different to putting the text into the argument of a command. Inside a group, the effect of any changes is local, so they will not interfere with the text following the closing curly-brace.

This is a different way of using curly braces to how we have used them before.

If you use a paragraph-formatting command like `\centering`, `\flushleft`, or `\flushright` inside a group, you **MUST** end the text with a `\par` command *inside the group* to cause the paragraph to be typeset with the desired format, otherwise the formatting simply will not take effect.

Environments like *center*, *quotation*, *table*, or *figure* are themselves groups, so the same rules apply, except that you do *not* need the `\par` at the end because most such environments are inherently paragraph-based and will do it for you.

A warning from the past

Beware of this 'vaine conceipt of simple men, which judge things by ther effects, and not by ther causes'. (Edmund Spenser, 1633)

It's hugely more efficient and productive to have control of the cause than the effect.