Advanced LaTeX

Tim Love

September 4, 1994

This document follows on from the *Word processing using LaTeX* handout. It describes the features of LaTeX 2$_\varepsilon$ that people at CUED are most likely to use. Further information is in the online handouts and in the books available for loan from the operators in the DPO. The *Local LaTeX* handout is a companion to this one. It describes some features especially suited to writing theses (bibliography and index compilation, for example) and includes a list of available packages.

Comments and bug reports to Tim Love (tpl@eng.cam.ac.uk).

Contents

1 LaTeX Concepts **2**

1.1 Environments and commands . . . 2
1.2 Classes and packages . . . 3
1.3 Errors . . . 3
1.4 Files created . . . 4
1.5 How to use LaTeX . . . 4

2 Document organisation and structuring **5**

2.1 Counters and Length parameters . . . 5
2.2 Document and page organisation . . . 6
2.3 Pagebreaks, footnotes, etc . . . 7

3 Color and Fonts **8**

3.1 Colored text . . . 8
3.2 Special characters . . . 9
3.3 Font Sizes . . . 10
3.4 Font Types . . . 10
3.5 Postscript Fonts . . . 11
3.6 Font attributes . . . 11
3.7 Selection commands . . . 12

Copyright ©1994 by T.P. Love. This document may be copied freely for the purposes of education and non-commercial research. Cambridge University Engineering Department, Cambridge CB2 1PZ, England.

4 Environments **13**
4.1 Alignments . . . 13
4.2 Listing Items . . . 13
4.3 Tabular . . . 15
4.4 Array . . . 15
4.5 Pictures . . . 16
4.6 Maths . . . 16
4.7 Figures . . . 17
4.8 Tabbing . . . 17
4.9 Verbatim . . . 18
4.10 Quote, abstract . . . 18
4.11 Letter . . . 18

5 Customising **18**
5.1 Macros . . . 18
5.2 New Commands . . . 19
5.3 Packages . . . 19
5.4 An Example . . . 19

6 More Information **20**

1 LaTeX Concepts

1.1 Environments and commands

LaTeX is a macro-package for TeX which has many preset *environments* where much of the setting up that TeX users have to do explicitly is done for you. An environment has the form

`\begin{`*environment name*`}`

.

`\end{`*environment name*`}`

LaTeX also has commands which affect the formatting of the document. Their arguments are given in braces. For example,

```
\textit{This is much more important} than this.
```

produces as output *This is much more important* than this.

The related `\itshape` command doesn't take an argument. It affects all the following text in the environment it's use in.

LaTeX tries to enforce the idea that the visual appearance of the document (use of fonts, indentation, etc) should derive from the logical structure of the document (*i.e.* rather than manually putting the section titles into bold, you should let the `\section` command do it). Resisting this philosophy can lead to extra, usually unnecessary work.

LaTeX is expandable. Many macros can be loaded in to provide added featues. You can also create your own commands and environments. Commands can take arguments that modify their action

- Some commands have a `*-form`, a variant on the standard command that you get by adding a `*` to the command name.
- Mandatory arguments are enclosed in `{}` braces
- Optional arguments are enclosed in `[ ]` brackets.

1.2 Classes and packages

At the top of your file you will have a line something like

```
\documentclass[12pt]{article}
```

which determines the font size and document class – the type of document you're writing. Each class has an associated `*.cls` file in the directory `/export/latex2e/inputs/` which is read in at start-up time. Other options accepted by `article` include `10pt`, `11pt` and `twocolumn`.

Then you'll probably need to load in extra macros. This is done using the `\usepackage` command. Each package has an associated `*.sty` file in `/export/latex2e/inputs/`. The packages inherit the options from the `\documentclass` line and can be given others of their own. *E.g.*

```
\usepackage[dvips]{graphics}
\usepackage[dvips]{color}
```

tells LaTeX that you want to use the extra `graphics` and `color` macros, and that you are going to use `dvips` to convert the resulting file to postscript. You can shorten this to

```
\usepackage[dvips]{graphics,color}
```

or even

```
\documentclass[dvips]{article}
\usepackage{graphics,color}
```

1.3 Errors

Errors can be reported either from the LaTeX phase (in which case it is reported as such) or the lower level TeX phase. Don't be too put off by the half-digested text displayed. A line number is reported which usually helps to detect the error. Typing `'h'` sometimes produces helpful diagnostics. The most common errors reported are

- a command misspelt
- a mismatched brace
- improper use of special characters

- forgetting to have the appropriate `\usepackage{...}` lines.
- a error like this

```
Overfull \hbox (15.42563pt too wide) in paragraph at lines 285--288
\OT1/ppl/m/n/10 You can cre-ate ver-tical space between lines or
ho-ri-zontal space   between
```

  means that lines 285 to 288 are producing a text line about 15 pts (about 5cm) too wide. `\OT1/ppl/m/n/10` is the specification of the font used. When TeX does right and left alignment, it works out how much space it needs to leave between words and where to hypenate words if necessary. But the amount of space it's prepared to leave has to fall within a certain range and it will only split words in certain places (shown in the error message by a hyphen). If these restrictions mean that LaTeX can't produce a satisfactory line, it will produce as much of the line as it can. A re-phrasing of the offending sentence will usually solve the problem. Another thing you might try is to control the way a troublesome word is hyphenated using something like

```
\hypenation{furthermore fur-ther-more}
```

  at the top of your file. If all else fails, use `{\sloppy...}` to enclose the offending text.

- LaTeX issues a `*`.

  This means LaTeX needs more input. It probably means you've missed out an `\end{document}`, but if not you may be able to get LaTeX to continue processing as best it can by typing `<Return>`.

- `! Argument of \label  has an extra }`

  If you're sure you haven't left out a left brace, then maybe you need to *protect* the inmost (fragile) command. For example, `\label` is fragile which causes a problem in `\caption{Picture\label{margin}}`, so the safer construction `\caption{Picture\protect\label{margin}}` has to be used.

1.4 Files created

More than just the `.dvi` file may be produced. Don't worry about them – except perhaps for the log file you won't need to look at them.

- `.aux` cross references, etc
- `.toc` created by `\tableofcontents`
- `.lof` created by `\listoffigures`
- `.lot` created by `\listoftables`
- `.log` a copy of the diagnostic output that usually comes out on the screen.

Some common errors can be found using the `lacheck` program on the LaTeX file.

1.5 How to use LaTeX

From the command line – After running `latex` at least twice, you should be able to preview your `*.dvi` document using `xdvi` and print it using `plotview`, but if postscript is implicated at all (if you load in graphics, use postscript fonts, scale, use color, or rotate) then you should convert your `*.dvi` document to postscript. A typical sequence of commands to process `doc.tex` would be

```
latex doc
latex doc
dvips -t a4 doc.dvi -o doc.ps
ghostview doc.ps
lp -dljmr1 -opostscript doc.ps
```

Using `xlatex` – `xlatex` has buttons to process, preview and print your document (or selected pages of it), and convert it to postscript. Just type `latex` *filename*.

Writing a LaTeX document is rather like writing a program. This makes using LaTeX more difficult in some respects than using a word processor, but there are advantages too. For instance creating a table of contents is trivial. Beginners often use unnecessary '`\\`' sequences and write '`{\large \textbf{2.1 Method}}\\`' when '`\subsection{Method}`' would be much better. Users who think they know more about typesetting than LaTeX (those who, for example, like underlining) will waste a lot of time too.

Avoid repeating constructions. Instead, write your own macros and commands, and familiarise yourself with the packages described in the *Local LaTeX* guide.

2 Document organisation and structuring

2.1 Counters and Length parameters

- **Counters** :- LaTeX maintains many counter variables (*e.g.* `page, part, equation, footnote, chapter, paragraph, section, subsection, subsubsection, enumi` etc). You can set these counters yourself. Some examples:-

  ```
  \setcounter{page}{0}
  \addtocounter{chapter}{2}
  ```

- **Length Parameters** :- LaTeX accepts the following units of length: `in, cm, mm, pt` (there are 72.27 pts to an inch), `em` (width of an `M`), `ex` (height of an `x`). These units can be used to set the values of length variables using `\setlength`. For example,

  ```
  \setlength{\parindent}{0in}
  ```

  sets to zero the amount by which the first line of a paragraph is indented. Other useful variables are:-

  `parskip`:- determines the gap between paragraphs.

baselineskip:- determines the usual distance from the bottom of one line to the bottom of the next. You can adjust this to produce double spacing, but a better way, which takes a stretch *factor* as argument is to use `baselinestretch`. For instance after `\begin{document}` you can do

```
\renewcommand{\baselinestretch}{1.2}
```

but note: it won't take effect until a font change so you might need to do something like `\small \normalsize`.

2.2 Document and page organisation

- **Document classes**: Choose from `article, book, letter, report, slides`. Except for `slides`, the differences between these are minor. A `book` can have chapters. In a `report` sections begin at `0` whereas in an `article` they begin at `1`. Just about all the handouts are `articles`.

- **Big Documents** :- It's best to split your document into smaller files and have a master file looking like this

```
\documentclass[dvips,12pt]{book}
\usepackage{a4,color,graphics,palatino,fancyheadings}
\begin{document}
\pagestyle{empty}
\tableofcontents
\listoffigures
\pagestyle{fancy}
\include{chapter1}
\include{chapter2}
\include{fig2}
\include{chapter3}
\include{chapter4}
\end{document}
```

  on which you can run `latex` just as if the master file contained all the text of `chapter1.tex` etc. The advantage of this is that once you have a chapter correct, you can comment out the corresponding 'include' line and avoid unnecessary processing. Remember to take out the `\begin{document}` and `\end{document}` lines from the component files.

- **Page Size** :- The default `a4` style has wide margins. The `a4_local` version uses more of the page. You can choose the margin sizes yourself by changing the following dimensions *before* the `\begin{document}` line.

```
\setlength{\topmargin}{-0.4in}
\setlength{\topskip}{0.3in}    % between header and text
\setlength{\textheight}{9.5in} % height of main text
\setlength{\textwidth}{6in}
\setlength{\oddsidemargin}{0.75in}
\setlength{\evensidemargin}{0.75in}
```

  To see the current values of these dimensions, use the `layout` package, which defines a `\layout` command.

- **Title Pages** :- The title page of this document was created by the following LaTeX commands.

```
\title{Advanced \LaTeX}
\author{Tim Love}
\date{\today}
\maketitle
```

- **Contents** :- Use `\tableofcontents` to create a contents list at that point in the document. LaTeX will pick out the sections, subsections etc for you. You'll have to run LaTeX at least twice though.

- **Page numbers and Headings** :- These are determined by the argument given in `pagestyle{}`.

  `plain` :- the page numbers are put at the bottom of the page. The top of the page is empty. This is the default mode.

  `empty` :- this suppresses page numbering altogether.

  `headings` :- this puts the page numbers at the top and adds a header whose contents depend on the document style.

  `fancyheadings` is a popular package that adds useful page headers when `\pagestyle{fancy}` is used. This handout uses it.

- **Sectioning** :- To start a chapter called **Life** in a `book`, just use `\chapter{Life}`. Similar commands to start a `part`, `section` or `subsection` also exist in most document classes (articles don't have chapters or parts though).

  If you use the `*-form` of the command then the sections will not be numbered, neither will it appear in the table of contents.

  A title will only be numbered if its 'depth' isn't more than the value of `secnumdepth` and will only appear in the contents page if the 'depth' isn't more than the value of `tocdepth`. So, for example, doing

```
\setcounter{tocdepth}{2}
\setcounter{secnumdepth}{3}
```

  will cause section `1.4.3` to be numbered, but it won't appear in the contents.

2.3 Pagebreaks, footnotes, etc

- **Page Breaks** :- you can force a page break using `\newpage`.

- **Preventing line breaks** :- If there's a word that you don't want broken, put it in an `mbox`. Eg,

```
One shouldn't try to break up \mbox\emph{relationships}
```

- **Space** :-

```
You can create vertical space \vspace{.5cm} between lines
or horizontal space \hspace{1.5cm} between words.
The arguments to these commands can be negative.
```

You can create vertical space between lines or horizontal space between words. The arguments to these commands can be negative.

- **Footnotes** :-

```
Do them like this\footnote{I told you so.}
```

Do them like this[1] The footnotes are numbered by default. If you want to use symbols (stars, daggers etc) then you need to redefine how the `footnote` counter is displayed.

```
\def\thefootnote{\fnsymbol{footnote}}
```

- **Margin notes** :-

```
These are easy too. Do them like this\marginpar{margin note}
```

These are easy too. Do them like this margin note

- **Cross References** :- At the end of this source file is the line

```
\label{THE_END}
```

`\pageref{THE_END}` will refer to the final page by number, and `\ref{THE_END}` will refer to it by section number. The last page (page 21) is in section 6. You'll have to run LaTeX at least twice to pick up forward references like these.

- **Boxed Text** :- For short pieces of text, use `\fbox`

```
Help. I'm \fbox{trapped}
```

Help. I'm trapped

- **Comments** :- Anything to the right of a % character is ignored by LaTeX .

3 Color and Fonts

3.1 Colored text

LaTeX 2_ε introduce some commands to control foreground and background colors. To use them you need

```
\usepackage[dvips]{color}
```

after the `\documentclass` line but before `\begin{document}`.

- `\textcolor{`*colorname*`}{`*text*`}` writes text in a color which can be specified by name (`black, white, red, green, blue` or a color name you've defined), RGB components, or grayscale.
- `\colorbox{`*colorname*`}{`*text*`}` writes text in a box with a colored background.

[1] I told you so.

- `\fcolorbox{`*colorname*`}{`*text*`}` writes text in a colored frame.
- `\pagecolor{`*colorname*`}` sets the color of the page's background.
- `\definecolor{`*colorname*`}{`*color specification*`}` lets you define new color names.

```
\definecolor{gold}{rgb}{0.85,.66, 0}
This is in \textcolor{red}{red} and this box is \colorbox{gold}{gold}.
Text color can be set using RGB values
(\textcolor[rgb]{0, 0.5, 0}{like so}), or \textcolor[gray]{0.2}{shades}
\textcolor[gray]{0.5}{of} \textcolor[gray]{0.8}{grey}.
```

produces This is in red and this box is gold. Text color can be set using RGB values (like so), or shades of grey.

3.2 Special characters

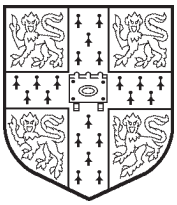
† is created by `\dag`, ‡ by `\ddag`, § by `\S`, ¶ by `\P`, £ by `\pounds`, ö by `\"{o}`. Many others are available in the `math` environment, including all the lower case greek letters and most of the upper case ones. If you only want to use a few characters you can bracket the symbols using `$` and `$` rather than `\begin{math}` and `\end{math}`. You can put a slash through any of these characters by prefacing them with `\not`

$\sqrt{i}$	`\sqrt{i}`	$\sqrt[5]{x+iy}$	`\sqrt[5]{x+iy}`	$\ldots$	`\ldots`
$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
α	`\alpha`	β	`\beta`	γ	`\gamma`
δ	`\delta`	ω	`\omega`	Γ	`\Gamma`
Θ	`\Theta`	Ω	`\Omega`	$\pm$	`\pm`
$\mp$	`\mp`	$\times$	`\times`	$\div$	`\div`
$*$	`\ast`	$\star$	`\star`	$\circ$	`\circ`
$\bullet$	`\bullet`	$\cdot$	`\cdot`	$\cap$	`\cap`
$\bigcap$	`\bigcap`	$\cup$	`\cup`	$\bigcup$	`\bigcup`
$\uplus$	`\uplus`	$\biguplus$	`\biguplus`	$\sqcap$	`\sqcap`
$\sqcup$	`\sqcup`	$\bigsqcup$	`\bigsqcup`	$\vee$	`\vee`
$\bigvee$	`\bigvee`	$\wedge$	`\wedge`	$\bigwedge$	`\bigwedge`
$\setminus$	`\setminus`	$\wr$	`\wr`	$\diamond$	`\diamond`
$\triangle$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\triangleleft$	`\triangleleft`
$\triangleright$	`\triangleright`	$\oplus$	`\oplus`		
$\bigoplus$	`\bigoplus`	$\ominus$	`\ominus`	$\otimes$	`\otimes`
$\bigotimes$	`\bigotimes`	$\oslash$	`\oslash`	$\odot$	`\odot`
$\bigodot$	`\bigodot`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\leq$	`\leq`	$\prec$	`\prec`	$\preceq$	`\preceq`
$\ll$	`\ll`	$\subset$	`\subset`	$\subseteq$	`\subseteq`
$\in$	`\in`	$\vdash$	`\vdash`		
$\geq$	`\geq`	$\succ$	`\succ`	$\succeq$	`\succeq`
$\gg$	`\gg`	$\supset$	`\supset`	$\supseteq$	`\supseteq`
$\sqsupseteq$	`\sqsupseteq`	$\ni$	`\ni`	$\dashv$	`\dashv`
$\equiv$	`\equiv`	$\sim$	`\sim`	$\simeq$	`\simeq`
$\asymp$	`\asymp`	$\approx$	`\approx`	$\cong$	`\cong`
$\neq$	`\neq`	$\doteq$	`\doteq`	$\propto$	`\propto`
$\models$	`\models`	$\perp$	`\perp`	$\mid$	`\mid`
$\parallel$	`\parallel`	$\bowtie$	`\bowtie`	$\smile$	`\smile`
$\frown$	`\frown`	$\leftarrow$	`\leftarrow`	$\Leftarrow$	`\Leftarrow`
$\rightarrow$	`\rightarrow`	$\Rightarrow$	`\Rightarrow`	$\leftrightarrow$	`\leftrightarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\mapsto$	`\mapsto`	$\hookleftarrow$	`\hookleftarrow`

$\leftharpoonup$	\leftharpoonup	$\leftharpoondown$	\leftharpoondown	$\rightleftharpoons$	\rightleftharpoons
$\longleftarrow$	\longleftarrow	$\Longleftarrow$	\Longleftarrow	$\longrightarrow$	\longrightarrow
$\Longrightarrow$	\Longrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\longmapsto$	\longmapsto	$\hookrightarrow$	\hookrightarrow	$\rightharpoonup$	\rightharpoonup
$\rightharpoondown$	\rightharpoondown	$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
$\downarrow$	\downarrow	$\Downarrow$	\Downarrow	$\updownarrow$	\updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow	$\swarrow$	\swarrow
$\nwarrow$	\nwarrow	$\aleph$	\aleph	$\hbar$	\hbar
$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\wp$	\wp	$\Re$	\Re	$\Im$	\Im
$\prime$	\prime		\empty	∇	\nabla
$\surd$	\surd	$\top$	\top	$\bot$	\bot
$\|$	|	$\angle$	\angle	$\forall$	\forall
$\exists$	\exists	$\neg$	\neg	$\flat$	\flat
$\natural$	\natural	$\sharp$	\sharp	$\backslash$	\backslash
∂	\partial	∞	\infty	$\triangle$	\triangle
$\clubsuit$	\clubsuit	$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit
$\spadesuit$	\spadesuit	$\sum$	\sum	$\prod$	\prod
$\coprod$	\coprod	$\int$	\int	$\oint$	\oint

One locally produced font with only one character is called `crest`.

```
\font\crest=crest
``This is the {\crest A} CUED crest''
```

produces "This is the  CUED crest"

3.3 Font Sizes

These are the available sizes.

tiny scriptsize footnotesize small normalsize

large Large LARGE huge Huge

If, for example, you want to use the smallest size, do

```
{\tiny ... }
```

If `Huge` isn't big enough for you, you can scale a postscript font up using the commands designed for graphics. `\resizebox{!}{5cm}{BIG}` produces

3.4 Font Types

Independent of size, these font types are at your disposal :- `\textrm` (roman), `textit` *(italic)*, `\textsc` (SMALL CAPS), `\emph` *(emphasis, but note that if you use* `emph` *within emphasized text, you will get* roman text*)*, `\textsl` *(slanting)*, `\texttt (teletype)`, `\textbf` **(boldface)**, `\textsf` (sans serif). As long as there's no conflict, these commands can be combined so that, for instance, **this is bold sans serif** can be produced by `\textsf{\textbf{this is bold sans serif}}`.

3.5 Postscript Fonts

It is easy to write a document that has postscript fonts. We have package support for helvetica (`helvetic`), utopia, times, optima, newcentury (`newcentu`), palatino and courier. To use palatino, all you need to do is add

```
\usepackage{palatino}
```

to your file. The `pifont` package has special commands for using the Zapf Dingbats font. `\dingfill{40}` completes the line with the specified symbol ✍ ✍

and `\dingline{36}` draws a whole line of symbols.

✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁ ✁

It's a good idea to use a font that's installed in the printer you intend to use. See the table on page 11 for information about the teaching system's `ljmr1` printer.

3.6 Font attributes

The commands above should give you sufficient control over fonts. If you don't want to know more at the moment then turn to section 4

Every text font in LaTeX has five *attributes*:

encoding This specifies the order that characters appear in the font. The most common values for the font encoding is `OT1`.

family The name for a collection of fonts, usually grouped under a common name by the font foundry. For example, 'Adobe Times' and Knuth's 'Computer Modern Roman' are font families. There are far too many font families to list them all, but some common ones are:

Internal fontname	*Fontname*	*In ljmr1?*
cmr	Computer Modern Roman	No
cmss	Computer Modern Sans	No
cmtt	Computer Modern Typewriter	No
cmm	Computer Modern Math Italic	No
cmsy	Computer Modern Math Symbols	No
cmex	Computer Modern Math Extensions	No
ptm	Adobe Times	Yes
phv	Adobe Helvetica	Yes
pcr	Adobe Courier	Yes
pun	Univers	No
ppl	Palatino	Yes
pagk	AvantGarde-Book	Yes
pagd	AvantGarde-Demi	Yes
pbk	Bookman	Yes
put	Utopia	No
pop	Optima	No
pnc	New Century Schoolbook	Yes
pzd	ZapfDingbats	Yes
rpad	Garamond	No

series How heavy or expanded a font is. For example, 'medium weight', 'narrow' and 'bold extended' are all series. The most common values for the font series are:

m	Medium
b	Bold
bx	Bold extended
sb	Semi-bold
c	Condensed

shape The form of the letters within a font family. For example, 'italic', 'oblique' and 'upright' are all font shapes. The most common values for the font shape are:

n	Normal (that is 'upright' or 'roman')
it	Italic
sl	Slanted (or 'oblique')
sc	Caps and small caps

size The design size of the font, for example '10pt'.

These five parameters specify every LaTeX font, for example:

	LaTeX specification				*Font*
OT1	cmr	m	n	10pt	Computer Modern Roman 10pt
OT1	cmss	m	sl	12pt	Computer Modern Sans Oblique 12pt
OML	cmm	m	it	10pt	Computer Modern Math Italic 10pt
T1	ptm	b	it	18pt	Adobe Times Bold Italic 18pt

3.7 Selection commands

There are commands to set attributes one at a time:

Command	*Attribute*	*Value in* article *class*, 10pt
`\textrm{..}` or `\rmfamily`	family	`cmr`
`\textsf{..}` or `\sffamily`	family	`cmss`
`\texttt{..}` or `\ttfamily`	family	`cmtt`
`\textmd{..}` or `\mdseries`	series	`m`
`\textbf{..}` or `\bfseries`	series	`bx`
`\textup{..}` or `\upshape`	shape	`n`
`\textit{..}` or `\itshape`	shape	`it`
`\textsl{..}` or `\slshape`	shape	`sl`
`\textsc{..}` or `\scshape`	shape	`sc`
`\tiny`	size	`5pt`
`\scriptsize`	size	`7pt`
`\footnotesize`	size	`8pt`
`\small`	size	`9pt`
`\normalsize`	size	`10pt`
`\large`	size	`12pt`
`\Large`	size	`14.4pt`
`\LARGE`	size	`17.28pt`
`\huge`	size	`20.74pt`
`\Huge`	size	`24.88pt`

The low-level commands used to change font attributes are as follows.

`\fontencoding{`*encoding*`}`
`\fontfamily{`*family*`}`
`\fontseries{`*series*`}`
`\fontshape{`*shape*`}`
`\fontsize{`*size*`}{`*baselineskip*`}`

Each of these commands sets one of the font attributes; `\fontsize` also sets `\baselineskip`. The actual font in use is not altered by these commands, but the current attributes are used to determine which font to use after the next `\selectfont` command.

`\selectfont` selects a text font, based on the current values of the font attributes. There *must* be a `\selectfont` command immediately after any settings of the font parameters by (some of) the five `\font<parameter>` commands, before any following text. For example, it is legal to say:

```
\fontfamily{ptm}\fontseries{b}\selectfont Some text.
```

to select bold Times Roman, but it is *not* legal to say:

```
\fontfamily{ptm} Some \fontseries{b}\selectfont text.
```

`\usefont{`*encoding*`}{`*family*`}{`*series*`}{`*shape*`}`
is short hand for the equivalent `\font...` commands followed by a call to `\selectfont`.

4 Environments

Some examples of how to use environments are given here.

4.1 Alignments

In these environments `\\` starts a new line.

```
\begin{flushleft}
Some people like to stay firmly\\ on the left whereas others
\end{flushleft}
\begin{flushright}
feel much more at home\\ on the right.\\
\end{flushright}
\begin{center}
but most of us prefer to stay\\ dead in the center.
\end{center}
```

Some people like to stay firmly
on the left whereas others

feel much more at home
on the right.

but most of us prefer to stay
dead in the center.

4.2 Listing Items

The items can be marked in one of three way:

```
\begin{itemize}
\item just by a bullet, using \texttt{itemize}

\item numbered, using \texttt{enumerate}

\begin{enumerate}
\item one
\item two
\item three
\end{enumerate}

\item or with a label, using \texttt{description}

\begin{description}
\item[itemize] bullets
\item[enumerate] automatic numbering
\item[description] labelling
\end{description}

\end{itemize}
```

- just by a bullet
- numbered
  1. one
  2. two
  3. three

- or with a label

  itemize bullets
  enumerate automatic numbering
  description labelling

The `pifont` package includes a variant of the `itemize` command that will replace the usual 'bullet' by a Zapf Dingbat symbol of your choice

```
\begin{dinglist}{43}
\item First
\item Second
\end{dinglist}
```

- ☞ First
- ☞ Second

and a variant of the `enumerate` command that given an initial Zapf Dingbat symbol will increment the symbol for each item.

```
\begin{dingautolist}{172}
\item First
\item Second
\end{dingautolist}
```

① First

② Second

4.3 Tabular

Tabular output is supported. When you create the environment you specify how many columns to have and how the contents are to be aligned (use `l, c` or `r` to represent each column with either left, center or right alignment) and where you want vertical lines (use |). The contents of the columns are separated by a '&' and rows by `\\`. To draw a full horizontal line, use `\hline` otherwise draw a line across selected columns using `\cline`. The `\multicolumn` command allows items to span columns. It takes as its first argument the number of columns to span. The following, rather complicated example shows how to use most of these facilities.

```
\begin{tabular}{||l|lr||} \hline
\textbf{Veg}  & \multicolumn{2}{|c||}{\textbf{Detail}}\\\hline
carrots    & per pound & \pounds 0.75 \\ \cline{2-3}
           & each      &         20p  \\ \hline
mushrooms  & dozen     &         86p  \\ \cline{1-1} \cline{3-3}
toadstools & pick your own & free     \\ \hline
\end{tabular}
```

Veg	Detail	
carrots	per pound	£0.75
	each	20p
mushrooms	dozen	86p
toadstools	pick your own	free

Tables won't continue on the next page if they're too long. The `longtable` commands are needed to do this. Type `handout latex2e_longtable` for details and examples.

If the text in a column is too wide for the page, LaTeX won't automatically text-wrap. Using `p{5cm}` instead of `c`, `l` or `r` in the `tabular` line will wrap-around the text in a 5 cm wide column.

There are various packages to assist with table creation. The `array` package adds some helpful features, including the ability to add formatting commands that control a whole column at a time, like so

```
\begin{tabular}{>{\ttfamily}l>{\scshape}c>{\Large}r}
      Text  & More Text & Large Text\\
      Left  & Centred   & Right
\end{tabular}
```

Text	More Text	Large Text
Left	Centred	Right

The `rotating` package is useful if you have a wide table that you want to display in landscape mode. You need to put your table inside `\begin{sideways}` and `\end{sideways}`.

4.4 Array

The `array` environment is similar to the `tabular` but must be within a *math* environment. This

```
\begin{math}
\left(
\begin{array}{clrr}
      a+b+c & uv & x-y & 27 \\
       x+y  & w  & +z  & 363
\end{array}
\right)
\end{math}
```

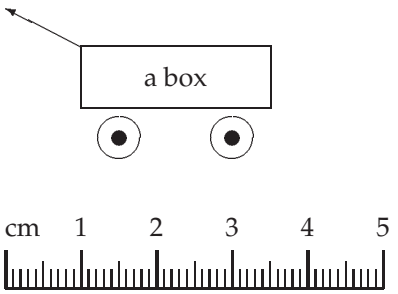
produces $\left(\begin{array}{clrr} a+b+c & uv & x-y & 27 \\ x+y & w & +z & 363 \end{array} \right)$

4.5 Pictures

LaTeX has some graphics capabilities. It's much better to use `xfig`. See the appropriate document (*LaTeX Maths and Graphics*) for more details.

```
\newcounter{cms}
\setlength{\unitlength}{1mm}
\begin{picture}(50,39)
\put(0,7){\makebox(0,0)[bl]{cm}}
\multiput(10,7)(10,0){5}{\addtocounter
   {cms}{1}\makebox(0,0)[b]{\arabic{cms}}}
\put(15,20){\circle{6}}
\put(30,20){\circle{6}}
\put(15,20){\circle*{2}}
```

```
\put(30,20){\circle*{2}}
\put(10,24){\framebox(25,8){a box}}
\put(10,32){\vector(-2,1){10}}
\multiput(1,0)(1,0){49}{\line(0,1){2,5}}
\multiput(5,0)(10,0){5}{\line(0,1){3,5}}
\thicklines
\put(0,0){\line(1,0){50}}
\multiput(0,0)(10,0){6}{\line(0,1){5}}
\end{picture}
```



4.6 Maths

Maths is dealt with in a separate handout (*LaTeX maths and graphics*). Here are some examples

```
\begin{math}
 \lim_{n \rightarrow \infty}x = 0 \\
x^{2y} \\
x_{2y} \\
x^{2y}_{1} \\
\frac{x+y}{1 + \frac{1}{n+1}} \\
\end{math}
```

produces $\lim_{n \rightarrow \infty} x = 0$
x^{2y}
x_{2y}
x^{2y}_{1}
$\frac{x+y}{1+\frac{1}{n+1}}$

4.7 Figures

To leave a space for graphics, use the `figure` environment. The arguments to figure specify where the space will be made, preferentially

- `h` here
- `t` top of page
- `b` bottom of page
- `p` on a page with no text

```
\begin{figure}[htbp]
   \vspace{1cm}
   \caption{1 cm of space}
\end{figure}
```

Figure 1: 1 cm of space

If you have a label defined in the caption, LaTeX is likely to give an error message. `\label` is a *fragile* command (see the LaTeX book for details) so you'll need to do something like

```
...
\caption{1 cm of space\protect\label{EMPTY}}
```

or simply put the `\label` command after the caption. Note that if you put the `\label` *before* the caption, the resulting reference will be the section number and not the figure number.

4.8 Tabbing

Within this environment tabs can be set by `\=` and the next tab moved to by using `\>`.

```
\begin{tabbing}
if \= it's raining\\      % set tab here, after the 'if'
   \> get an umbrella \=\\ % go to the defined tab and set a new one
else\\
   \> get wet \> * next tab is here\\
endif
\end{tabbing}
```

if it's raining
  get an umbrella
else
  get wet * next tab is here
endif

4.9 Verbatim

Within this environment things come out unformatted. It's useful for showing examples of typed input and provides a way of printing characters that have a special meaning for LaTeX.

```
\begin{verbatim}
caret is ^, tilde is ~ and backslash is \
\end{verbatim}
```

produces

```
caret is ^, tilde is ~ and backslash is \
```

If you just want to quote a few characters, use `\verb|`*quoted text*`|`. The characters delimiting the quote can be anything as long as they are the same.

4.10 Quote, abstract

These widen the margins and change the font. The `abstract` environment also adds a title.

4.11 Letter

See `letter.tex` in `/export/Examples/LaTeX2e`

5 Customising

5.1 Macros

At the top of the source of this file is

```
\def\xdt{$\cal X\!\!$\texttt{.desktop}}
```

which defines `\xdt` to be $\mathcal{X}$.`desktop`. Using such constructions can make your document much tidier, and saves on typing.

5.2 New Commands

A new command can be created using

`\newcommand{\`*commandname*`[`*number of arguments*`]{`*command text, using* `#1`, `#2` *etc to denote arguments*`}`

For example,

```
\newcommand{\ve}[1]{\(#1_1 ... #1_n\)}
\ve{x}
```

produces as output $x_1 \ldots x_n$

A problem with this example is that it should change to math mode if LaTeX is already in that mode. A better try would be

```
\newcommand{\ve}[1]{\ensuremath{#1_1 ... #1_n}}
```

which will only change to math mode if it's necessary.

A new environment is just as easily created. The following provides a variant of the itemize command.

```
\newenvironment{emlist}{\begin{itemize} \em}{\end{itemize}}
\begin{emlist}
\item first comment
\item second comment
\end{emlist}
The end of the environment ends the scope of the emphasis.
```

- *first comment*
- *second comment*

The end of the environment ends the scope of the emphasis.

5.3 Packages

There are many features and options not mentioned in this handout. See the *Local LaTeX Guide* for more details. You can often find out what you want by looking at the files in `/export/latex2e/inputs`. If your file begins

```
\documentclass[12pt]{article}
```

then various macro files are read when you run `latex`, namely (`article.cls`, `art12.sty` and `latex.tex`). Some of these files are well-enough commented to be useful documentation.

5.4 An Example

The following example uses features mentioned in `latex.tex`.

`minipage` creates a miniature page complete with its own footnotes etc. It can be created of any set width. Footnotes within a minipage use a different counter to other footnotes[2]. This example ensures that all the numbers follow on in sequence. The footnotes in minipages are marked by letters rather than numbers, so here the type is changed to arabic.

```
\begin{minipage}{\textwidth}
% Set the minipage footnote counter
\setcounter{mpfootnote}{\value{footnote}}
% Redefine the command that produces the footnote number
\renewcommand{\thempfootnote}{\arabic{mpfootnote}}
\begin{tabular}{|ll|}\hline
one & two\footnote{A minipage footnote}\\\hline
\end{tabular}
\setcounter{footnote}{\value{mpfootnote}}
\end{minipage}
```

one	two[3]

[3] A minipage footnote

You might find this trick useful if you want footnotes in tables. They don't come out otherwise.[4]

Here's an example of using a package. The `multicol` file lets one change the number of columns easily. To switch into 3-column text use

```
\begin{multicols}{3}{
Put the text here. Maths, tables, pictures etc are all ok, but not
figures. But you have to remember to load in the \texttt{multicol}
package at the top of your document.
}
\end{multicols}
```

Put the text here. Maths, tables, pictures etc are all ok, but not figures. But you have to remember to load in the `multicol` package at the top of your document.

[2] like this one

[4] Not for me anyway

LaTeX uses the value of the environmental variable `TEXINPUTS` to decide where to look for `sty` files. If you create a directory called (say) `inputs`, copy a system `*.sty` file into it and do

```
export TEXINPUTS=inputs:/export/latex2e/inputs
```

then your copy will be read in preference to the system one and you can customise easily.

6 More Information

- See the `uk.tex` newsgroup.
- Use the TeX Archive `ftp.tex.ac.uk`
- Use `gopher` on `club` to access a database of TeX info. Select `Services` to find the option. Suppose we wanted to find out how to put italian into LaTeX. Type `latex italian` in reply to the search prompt and relevant files will be presented to you for viewing or copying.
- The LaTeX 2_ε handouts (with source) are online. Just type '`handouts LaTeX2e`'.
- See the examples in `/export/Examples/LaTeX2e`.
- Borrow a LaTeX manual from the machine room.
- Look at the files in `/export/latex2e/inputs`, especially, `latex.tex`.