LaTeX

CTAN lion drawing by Duane Bibby

Markus Kuhn

Computer Laboratory, University of Cambridge

https://www.cl.cam.ac.uk/teaching/1718/TeX+MATLAB/

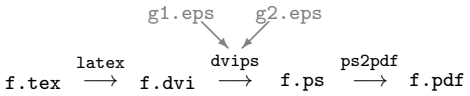
LaTeX – a document formatter

LaTeX: a macro package for the TeX ($\tau\epsilon\chi$, "tech") typesetting system

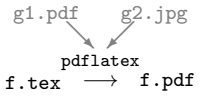
- excellent facilities for mathematical notation
- de-facto standard for preparing scientific publications in mathematical, physical, computing and engineering disciplines

History: Donald Knuth (Stanford CS) developed TeX in mid 1970s in SAIL to typeset his "Art of Computer Programming" books, reimplemented it in Pascal in the mid 1980s (WEB, literate programming), was later ported to C.
Leslie Lamport (SRI, DEC) wrote LaTeX in early 1980s.
Both now community maintained as *TeX Live* open-source distribution.

Classic processing steps:



Modern alternative:

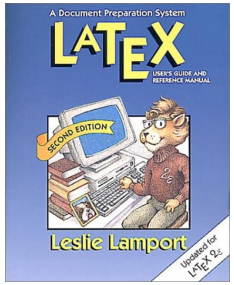


These slides: prepared using the LaTeX `beamer` class.

LaTeX – resources

Recommended introduction

Leslie Lamport: LaTeX – a document preparation system. 2nd ed., Addison-Wesley, 1994. (CL library)



Installation

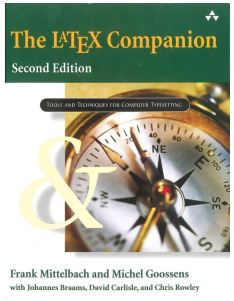
Debian Linux:	`apt-get install texlive`
macOS:	`http://www.tug.org/mactex/`
Windows:	`https://miktex.org/`
	`http://www.tug.org/protext/`

More documentation

Online tutorials:
`http://www.latex-project.org/guides/`

TeX Frequently Asked Questions list:
`http://www.tex.ac.uk/`

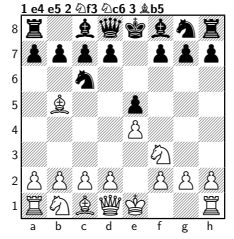
For advanced users:
Mittelbach, et al.: *The LaTeX Companion*.
2nd ed., Addison-Wesley, 2004.



LaTeX – features and benefits

- Most popular mathematical typesetting language
  (subset imitations now also in: Word, MathJax, MediaWiki, etc.)
- Encourages logical markup ⇒ helps to maintain consistent style
- Plain-text source ⇒ easy to collaborate via version-control systems
- Command-line tool ⇒ easy to automate build (make)
- Use any plain-text editor you like (Emacs, vi, TeXworks, Word, etc.)
- Robust for large, complex documents (PhD thesis, books, etc.)
- Highly extensible (a Turing-complete macro programming language)
  ⇒ vast collection of add-on packages for special typesetting needs (figures, logic proofs, pseudo code, circuit diagrams, flow charts, chemical formulae, slides, chess positions, etc.)

```
\usepackage[skaknew]{skak}
\newgame
\mainline{1.e4 e5 2. Nf3 Nc6 3.Bb5}
\showboard
```



- Computer Modern, etc. ⇒ free font families
- Mature, free, portable, open source, used by many science publishers

LaTeX example

```
\documentclass[12pt]{article}
\setlength{\textwidth}{75mm}
\begin{document}
\title{\TeX\ -- a summary}
\author{Markus G.~Kuhn}
\date{30 October 2017}
\maketitle
\thispagestyle{empty}

\section{Introduction}

Mathematical formul\ae\ such as
$e^{i\pi} = -1$ or even
\[ \Phi(z) = \frac{1}{\sqrt{2\pi}}
   \int_0^x e^{-\frac12 x^2}        \]
were a real `pain' to typeset until
\textsc{Knuth}'s text formatter \TeX\
became available \cite{Knuth86}.

\begin{thebibliography}{9}
  \bibitem{Knuth86}Donald E. Knuth:
  The \TeX book. Ad\-dison-Wesley, 1986.
\end{thebibliography}

\end{document}
```

TEX – a summary

Markus G. Kuhn

30 October 2017

1 Introduction

Mathematical formulæ such as $e^{i\pi} = -1$ or even

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{1}{2}x^2}$$

were a real 'pain' to typeset until KNUTH's text formatter TEX became available [1].

References

[1] Donald E. Knuth: The TEXbook. Addison-Wesley, 1986.

TEX input syntax

- TEX reads plain-text `*.tex` files (e.g., prepared with emacs)
- no distinction is made between space character and line feed
- multiple spaces are treated like a single space
- empty lines act as paragraph separator (just like the `\par` command)
- command, macro and variable names start with a backslash (\), followed by either a sequence of letters or a single non-letter character (uppercase/lowercase is significant).

  Correct: `\par`, `\item`, `\pagethree`, `\LaTeX`, `\+`, `\\`, `\3`

  Wrong: `\page33`, `\<>`
- space and line-feed characters are ignored if they follow a command/macro/variable name consisting of letters. Use `\␣` to add an explicit space (e.g., `\TeX\ syntax` ⇒ TEX syntax).

Characters with special semantics

In `*.tex` input files, the following are meta characters (part of syntax):

```
# $ % & ~ _ ^ \ { }
```

#$%&_^{} can be included in regular text using the macros

```
\# \$ \% \& \_ \^ \{ \}
```

Otherwise:

%	starts a comment	#	macro parameter
~	no-break space	&	tabulator mark
$	delimits inline equations	\	macro prefix, \\ line separator
^	superscript	_	subscript (in math mode)
{	begin group	}	end group

Comments: All characters between (and including) a % and the next line feed will be ignored. Append % at the end of a line to avoid interpretation of the subsequent line feed as a space.

One could also insert \ and ~ as `\textbackslash` and `\textasciitilde`. But this is rarely ever done. The ASCII characters _~^`{} are not typically used in regular text. They are common in computing-related strings (identifiers, source code, path names, URLs, etc.), for which it is customary to use a fixed-width `typewriter font`. There use the `verbatim` environment or the macro `\verb+...+`, which typeset all ASCII characters in typewriter font, or `\url{...}` for URLs.

Grouping

The curly braces { and } serve two purposes in TEX:

- Lexical scope: a { saves current state on a stack and } restores it, therefore state changes (e.g., font switch, variable assignment) inside a { ... } group last only until the }:

  ```
  This is a {\bfseries bold} statement.
  ```
  ⇓
  This is a **bold** statement.
- Commands and macros read for each argument either a single character or a group enclosed by { and }:

  ```
  Typeset \textbf M in \textbf{boldface}.
  ```
  ⇓
  Typeset **M** in **boldface**.

LaTeX macros can have optional arguments, which are enclosed by [...]:

```
\makebox[80mm][c]{this is centered in a box 80 mm wide}
```

Changing font style manually

Declaring shape, series, family (each independently):

`\mdseries`	Medium series	`\upshape`	Upright shape
`\bfseries`	**Boldface series**	`\itshape`	*Italic shape*
`\rmfamily`	Roman family	`\slshape`	*Slanted shape*
`\sffamily`	Sans-serif family	`\scshape`	Small caps shape
`\ttfamily`	`Typewriter family`	`\normalfont`	Normal style

Corresponding commands that change the style only for their argument:

`\textmd{`*text*`}`	`\mdseries`	`\textup{`*text*`}`	`\upshape`
`\textbf{`*text*`}`	`\bfseries`	`\textit{`*text*`}`	`\itshape`
`\textrm{`*text*`}`	`\rmfamily`	`\textsl{`*text*`}`	`\slshape`
`\textsf{`*text*`}`	`\sffamily`	`\textsc{`*text*`}`	`\scshape`
`\texttt{`*text*`}`	`\ttfamily`	`\textnormal{`*text*`}`	`\normalfont`

Change type size:

`\tiny`	`\small`	`\large`	`\huge`
`\scriptsize`	`\normalsize`	`\Large`	`\Huge`
`\footnotesize`		`\LARGE`	

For style consistency: use such low-level font commands only exceptionally (e.g., title pages, special notations). Otherwise leave any font adjustments to higher-level semantic commands such as `\emph{`*text*`}` or `\section{`*heading*`}`.

Typewriting versus typesetting

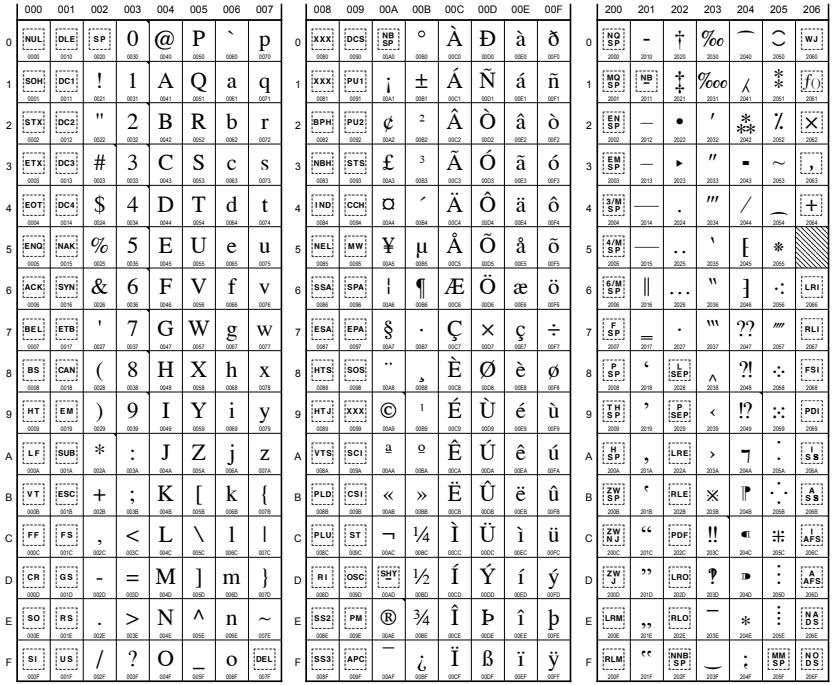
The ASCII (ISO 646) 7-bit character set has only 94 graphic characters:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

They were chosen to cover the character repertoire of US typewriters and teletype printers. The standards committee added a few more symbols ([\]{|}_) in the hope that they will be useful for programming.

TEX defines a number of shortcuts and macros to access the much larger set of of "typographic" characters used by book printers.

These typographic characters still cannot be found on standard PC keyboard layouts, which were designed for 7-bit ASCII.



The Unicode Standard 9.0, Unicode Inc., http://www.unicode.org/charts/

Dashes

ASCII provides only a single combined hyphen-minus character, but typesetters distinguish carefully between several dash characters:

`-`	⇒	-	hyphen	U+002D
`--`	⇒	–	en dash	U+2013
`---`	⇒	—	em dash	U+2014
`$-$`	⇒	−	minus	U+2212

The hyphen (-) is the shortest of these and is used to combine separate words or split words across line-breaks.

The en dash (–) is often used to denote a range of numbers (as in pages 64–128), or – as in this example – as a punctuation dash.

The em dash is used—like this—as a punctuation dash, often without surrounding space, especially in US typography.

The minus (−) is a mathematical operator, whose shape matches the plus (+), unlike the hyphen or dashes. Compare: -+, –+, —+, −+.

Quotation marks

Typewriters and ASCII offer only undirectional 'single' and "double" quotation marks, while typesetters use ‘curly’ and “directed” variants.

TEX input files use the single quotation mark (') and the grave accent (`) to encode these, as well the mathematical ‘prime’ marker and the French accents:

`` ` ``	⇒	‘	left quote	U+2018
'	⇒	’	right quote	U+2019
`` ``	⇒	“	left doublequote	U+201C
''	⇒	”	right doublequote	U+201D
$'$	⇒	′	prime	U+2032
\'u	⇒	ú	acute accent	U+00B4
\`u	⇒	ù	grave accent	U+0060

The apostrophe (it’s) is identical to the right single quotation mark.

In some older terminal fonts (especially of US origin), the ` and ' characters have a compromise shape somewhere between the quotation marks ‘’ and the accents `´.

Non-ASCII Symbols

¡	!`	Å	\AA	¶	\P
¿	?`	ø	\o	†	\dag
œ	\oe	Ø	\O	‡	\ddag
Œ	\OE	ł	\l	©	\copyright
æ	\ae	Ł	\L	£	\pounds
Æ	\AE	ß	\ss	...	\ldots
å	\aa	§	\S		

Combining characters

ó	\'o	ō	\=o	o͡o	\t{oo}
ò	\`o	ȯ	\.o	o̧	\c{o}
ô	\^o	ŏ	\u{o}	ọ	\d{o}
ö	\"o	ǒ	\v{o}	o̲	\b{o}
õ	\~o	ő	\H{o}		

Space – the final frontier

Traditional English typesetting inserts a larger space at the end of a sentence. TEX believes any space after a period terminates a sentence, unless it is preceded by an uppercase letter. Parenthesis are ignored.

This works often: J. F. Kennedy’s U.S. budget. Look!
But not always: E.g. NASA. Dr. K. Smith et al. agree.

To correct failures of this heuristic, use

~	⇒	no-break space
\␣	⇒	force normal space
\@	⇒	following punctuation ends sentence

as in

```
E.g.\ NASA\@. Dr.~K. Smith et al.\ agree.
```

⇓

E.g. NASA. Dr. K. Smith et al. agree.

Or disable the distinction of spaces with `\frenchspacing`.

Structure of a LATEX document

First select a document class and its options, e.g. with

```
\documentclass[12pt,a4paper]{article}
```

Standard classes: `article`, `report`, `book`, `letter`, `slides`.

Publishers often provide authors with their own class as a `*.cls` file. Appendix A of *The LATEX Companion* explains how to write new class files. A popular class for presentation slides: `beamer`

Environment: group delimited by `\begin{`*name*`}` and `\end{`*name*`}`, e.g.

```
\begin{document}
...
\end{document}
```

Common environments: `abstract`, `center`, `verbatim`, `itemize`, `enumerate`, `quote`, `tabular`, `equation`, ...

Mark headings with

```
\section{...}            \subsubsection{...}
\subsection{...}         \paragraph{...}
```

and LATEX will take care of font sizes, numbering, and table of contents.

LaTeX list environments

```
\begin{itemize}
  \item Mammals
  \begin{itemize}
    \item Dogs
    \item Ponys
  \end{itemize}
  \item Insects
  \item[+] \ldots
\end{itemize}
```

- Mammals
  - Dogs
  - Ponys
- Insects

+ ...

```
\begin{enumerate}
  \setcounter{enumi}{-1}
  \item Fruits
  \begin{enumerate}
    \item Apples
    \item Cucumbers\label{c}
  \end{enumerate}
  \item Veggies
        (see also \ref{c})
  \item \ldots
\end{enumerate}
```

0. Fruits
   (a) Apples
   (b) Cucumbers
1. Veggies (see also 0b)
2. ...

Package `enumitem` adds many configuration options to these environments. (`texdoc enumitem`)

Verbatim text: quoting source code

Lines of source code: the `verbatim` environment disables all meta characters, uses a typewriter font and preserves space and line feed:

```
\begin{verbatim}
$initial = substr($record->{'name'}, 0, 1);
\end{verbatim}
```

Do not indent a `verbatim` block (prints *all* whitespace). Keep `\end{verbatim}` on its own line.

Source code with math mode: `alltt` (`\usepackage{alltt}`) is similar to `verbatim`, but keeps the meta characters `\{}`, so you can still switch fonts and typeset mathematical expressions:

```
\begin{alltt}
for \(i:=1,\ldots,n\)
  print \(x\sb{i}\sp{2}\)
\end{alltt}
```

```
for i := 1,...,n
  print x_i^2
```

Inline strings: use `\verb+`*text*`+` to quote *text* inside a paragraph, where + is any character that does not occur in *text*. This also disables meta characters, preserves whitespace, and switches to a typewriter font.

You can't use `\verb` in command arguments, use `\texttt{`*text*`}` there instead.
The `\verb*+`*text*`+` variant prints spaces as ␣. Fix single quotation mark: `\usepackage{upquote}`.

Tweaking and extending LaTeX

LaTeX behaviour can be changed by overwriting predefined variables and macros. This can be done

- in the *preamble* (between the `\documentclass{...}` and `\begin{document}` lines) ⇒ for the entire document
- anywhere in the document ⇒ the effect will last only until the end of the current group (i.e., the next } or `\end{...}`)

Packages

A huge collection of extension packages exists for LaTeX. Some merely define additional macros and environments, others rewrite parts of LaTeX's internal machinery. For example, adding to the preamble

```
\usepackage{hyperref}
```

loads all the macros and settings defined in the `hyperref.sty` package.

`hyperref` adds new marcros, such as `\url{...}` for typesetting URLs, but also automatically turns every reference to a page, section, or bibliographic entry into a hyperlink.

Documentation: `texdoc` *packagename* e.g. `texdoc geometry`

Example: changing page layout geometry

Adjust margins manually, via numerous length variables:

```
\setlength{\oddsidemargin}{-0.4mm} % 25 mm left margin
\setlength{\evensidemargin}{\oddsidemargin}
\setlength{\textwidth}{160mm}      % 25 mm right margin
\setlength{\topmargin}{-5.4mm}     % 20 mm top margin
\setlength{\headheight}{5mm}
\setlength{\headsep}{5mm}
\setlength{\footskip}{10mm}
\setlength{\textheight}{237mm}     % 20 mm bottom margin
```

More comfortable:

```
\usepackage[vmargin=20mm,hmargin=25mm]{geometry}
```

The `geometry.sty` package automatically recalculates any dimensions not specified.

Make paragraphs not indented at the first line, but spaced apart slightly:

```
\setlength{\parindent}{0mm}
\setlength{\parskip}{\medskipamount}
```

Or just:

```
\usepackage{parskip}
```

Mathematical typesetting

In TEX, mathematical formulas are formatted in a completely different mode from that used for normal text.

Inline formulas such as a_n (`$a_n$`) that appear as part of a normal paragraph have to be surrounded with `$...$` or `\(...\)`, while `\[...\]` produces a displayed formula, such as

$$F_n = F_{n-1} + F_{n-2} \qquad \texttt{\textbackslash[F\_n = F\_\{n-1\} + F\_\{n-2\} \textbackslash]}$$

In math mode

- space characters are ignored; TEX adds its own space around operators based on heuristics; manually add thinspace with "`\,`"
- a special *math italic* font is used, with different inter-character spacing, designed for *single-letter variables* concatenated in products
- many additional macros for special symbols are defined

Math italic is very $different$ and not suitable for writing words or units! Use `\mathrm{...}` around words, as in `$v_\mathrm{diff}$` $\rightarrow v_{\mathrm{diff}}$.

Macros for common function symbols (constants: upright font!):

$\max_{x\in\mathbb{R}}\{\log_2(\cos x)\}$ `$\max_{x\in\mathbb{R}}\{\log_2(\cos x)\}$`

Macros for neatly aligning multiple equations: `\usepackage{amsmath}`, see `texdoc amsldoc`.

Mathematical symbols

Greek letters

Γ	`\Gamma`	δ	`\delta`	π	`\pi`
Δ	`\Delta`	ϵ	`\epsilon`	ϖ	`\varpi`
Θ	`\Theta`	ε	`\varepsilon`	ρ	`\rho`
Λ	`\Lambda`	ζ	`\zeta`	ϱ	`\varrho`
Ξ	`\Xi`	η	`\eta`	σ	`\sigma`
Π	`\Pi`	θ	`\theta`	ς	`\varsigma`
Σ	`\Sigma`	ϑ	`\vartheta`	τ	`\tau`
Υ	`\Upsilon`	ι	`\iota`	υ	`\upsilon`
Φ	`\Phi`	κ	`\kappa`	ϕ	`\phi`
Ψ	`\Psi`	λ	`\lambda`	φ	`\varphi`
Ω	`\Omega`	μ	`\mu`	χ	`\chi`
α	`\alpha`	ν	`\nu`	ψ	`\psi`
β	`\beta`	ξ	`\xi`	ω	`\omega`
γ	`\gamma`	o	`o`		

Mathematical symbols

Binary operators

$\pm$	`\pm`	$\lhd$	`\lhd`	$\oplus$	`\oplus`
$\mp$	`\mp`	$\cap$	`\cap`	$\ominus$	`\ominus`
$\setminus$	`\setminus`	$\cup$	`\cup`	$\otimes$	`\otimes`
$\cdot$	`\cdot`	$\uplus$	`\uplus`	$\oslash$	`\oslash`
$\times$	`\times`	$\sqcap$	`\sqcap`	$\odot$	`\odot`
$\ast$	`\ast`	$\sqcup$	`\sqcup`	$\dagger$	`\dagger`
$\star$	`\star`	$\wr$	`\wr`	$\ddagger$	`\ddagger`
$\diamond$	`\diamond`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\circ$	`\circ`	$\rhd$	`\rhd`	$\unlhd$	`\unlhd`
$\bullet$	`\bullet`	$\vee$	`\vee`	$\unrhd$	`\unrhd`
$\div$	`\div`	$\wedge$	`\wedge`		

$\triangleleft$	`\triangleleft`	$\bigtriangleup$	`\bigtriangleup`
$\triangleright$	`\triangleright`	$\bigtriangledown$	`\bigtriangledown`

Mathematical symbols

Relations

$\leq$	`\leq`	$\geq$	`\geq`	$\equiv$	`\equiv`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\ll$	`\ll`	$\gg$	`\gg`	$\asymp$	`\asymp`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\smile$	`\smile`	$\mid$	`\mid`	$\doteq$	`\doteq`
$\frown$	`\frown`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`	$\bowtie$	`\Join`

$\not<$	`\not<`	$\not=$	`\not=`	$\not>$	`\not>`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	...	

Mathematical symbols

Arrows

$\leftarrow$	\leftarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\Leftarrow$	\Leftarrow	$\longmapsto$	\longmapsto
$\rightarrow$	\rightarrow	$\hookrightarrow$	\hookrightarrow
$\Rightarrow$	\Rightarrow	$\rightharpoonup$	\rightharpoonup
$\leftrightarrow$	\leftrightarrow	$\rightharpoondown$	\rightharpoondown
$\Leftrightarrow$	\Leftrightarrow	$\leadsto$	\leadsto
$\mapsto$	\mapsto	$\uparrow$	\uparrow
$\hookleftarrow$	\hookleftarrow	$\Uparrow$	\Uparrow
$\leftharpoonup$	\leftharpoonup	$\downarrow$	\downarrow
$\leftharpoondown$	\leftharpoondown	$\Downarrow$	\Downarrow
$\rightleftharpoons$	\rightleftharpoons	$\updownarrow$	\updownarrow
$\longleftarrow$	\longleftarrow	$\Updownarrow$	\Updownarrow
$\Longleftarrow$	\Longleftarrow	$\nearrow$	\nearrow
$\longrightarrow$	\longrightarrow	$\searrow$	\searrow
$\Longrightarrow$	\Longrightarrow	$\swarrow$	\swarrow
$\longleftrightarrow$	\longleftrightarrow	$\nwarrow$	\nwarrow

Mathematical symbols

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists
$\imath$	\imath	∇	\nabla	$\neg$	\neg
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat
ℓ	\ell	$\top$	\top	$\natural$	\natural
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp
$\Re$	\Re	$\|$	|	$\clubsuit$	\clubsuit
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit
∂	\partial	$\triangle$	\triangle	$\heartsuit$	\heartsuit
∞	\infty	$\backslash$	\backslash	$\spadesuit$	\spadesuit
$\Box$	\Box	$\Diamond$	\Diamond		

$\ldots$	\ldots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots

Use \ldots in $a, b, \ldots, z$, but \cdots in $a + b + \cdots + z$. Never write ...

Large operators and delimiters

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge		

These appear smaller inline than in displayed equations: $\prod_0^{n-1}$ vs $\prod_0^{n-1}$

$[$	\lbrack	$]$	\rbrack
$\lfloor$	\lfloor	$\rfloor$	\rfloor
$\lceil$	\lceil	$\rceil$	\rceil
$\{$	\lbrace	$\}$	\rbrace
$\langle$	\langle	$\rangle$	\rangle
$[\![$	[\![	$]\!]$	]\!]
$\langle\!\langle$	\langle\!\langle	$\rangle\!\rangle$	\rangle\!\rangle

\left(and \right) grow delimiters to the height of what they enclose:

$\left(\sum_{i=0}^\infty x^i \right)$ \left(\sum_{i=0}^\infty x^i \right)

Mathematical symbols

Alternative names

$\neq$	\ne	$\{$	\{	$\ni$	\owns	$\vert$	\vert
$\neq$	\neq	$\}$	\}	$\wedge$	\land	$\Vert$	\Vert
$\leq$	\le	$\to$	\to	$\vee$	\lor		
$\geq$	\ge	$\leftarrow$	\gets	$\neg$	\lnot		

Stacking things

a^b	a^{b}	a_b	a_{b}
$\overline{a-b}$	\overline{a-b}	$\overbrace{a-b}$	\overbrace{a-b}
$\underline{a-b}$	\underline{a-b}	$\underbrace{a-b}_c$	\underbrace{a-b}_c

$$= \begin{cases} a^{2^{2^2}}, & a \geq 0 \\ -\frac{1}{a}, & a < 0 \end{cases}$$

```
=\left\{\begin{array}{cl}
        a^{2^{2^2}}, & a \ge 0 \\
        -\frac{1}{a}, & a <  0
      \end{array}\right.
```

Including graphics

DVI only supports characters and filled rectangles, but `dvips` and `pdftex` also understand embedded "special" instructions that provide more.

Embedded PostScript (EPS) vector graphics:

Normal PostScript files (*.ps) produce a sequence of pages. An EPS file describes only an image and is meant to be included into a PostScript page. EPS files lack instructions to output paper, but define a rectangular "bounding box", using special `%%BoundingBox:` comments.

Load the `graphicx` extension of LaTeX by adding

```
\usepackage{graphicx}
```

to the preamble. Then write

```
\includegraphics{filename.eps}
```

wherever you want to include the graphics file into your text.

In `pdflatex`, the `graphicx` package allows you to include graphics from PDF (vector graphics), JPEG (photos) and PNG (bitmap) files:

```
\includegraphics{filename.pdf}
```


Postscript/PDF graphics facilities

Applying coordinate transforms:

The `graphicx` package provides access to the geometric transform capabilities of the PostScript and PDF languages:

```
\scalebox{0.8}{\includegraphics{diagram.pdf}}
\includegraphics[height=60mm]{screenshot.png}
\includegraphics[width=0.9\linewidth]{photo.jpg}
\resizebox{190mm}{60mm}{becomes 19 cm $\times$ 6 cm large}
\resizebox{190mm}{!}{this becomes 19 cm wide}
\rotatebox{180}{this is upside down!}
```

Changing colours:

The `color` package also uses Postscript/PDF special commands:

```
This text is \textcolor{red}{printed in red} if ...
```

This text is printed in red if you include `\usepackage{color}`.

Default: `\definecolor{red}{rgb}{1,0,0}`

Figures and references

Larger diagrams interfere with page breaking. They are best placed into a `figure` environment, such that LaTeX can move them around. Example:

```
\begin{figure}
  \includegraphics[width=0.6\linewidth]{photo.jpg}
  \caption{This photograph shows the experimental setup.}
  \label{fig:expsetup}
\end{figure}
```

The automatically assigned figure number can be quoted as in:

```
See also Figure~\ref{fig:expsetup}
(page~\pageref{fig:expsetup}).
```

The `\label{...}` command can also be used after `\section{...}`, `\subsection{...}`, etc. and inside `\begin{equation}` ... `\end{equation}` to assign symbolic names to section and equation numbers, which can then be resolved via `\ref{...}` or `\pageref{...}`.

No need to manually renumber figures, sections, or equations!

Build tools for LaTeX

To make sure `\label` references and tables of contents use the correct numbers, it may be necessary to call `latex` twice. It will output "Rerun to get cross-references right" in this case.

The following implicit Makefile rule takes care of this:

```
.DELETE_ON_ERROR:

%.pdf %.aux %.idx: %.tex
        pdflatex $<
        while grep 'Rerun to get ' $*.log ; do pdflatex $< ; done
```

An alternative is the "`latexmk`" tool, which automatically determines dependencies (e.g. from `\includegraphics`) and recompiles LaTeX documents where file modification timestamps indicate that this is necessary.

Graphics editor `xfig`

- Its `*.fig` files have a simple plain-text format that can be edited manually, script generated, and leads to useful diffs.
- Can export `*.eps` or `*.pdf` files
- Can also produce figures in which LaTeX is used to fill in all the text. This provides math mode, macros, symbols, references, fonts that match the main text, etc.

  Ask `xfig` to export a `*.pstex` + `*.pstex_t` file pair. The `*.pstex` file lacks the text parts of the figure. The `*.pstex_t` file contains LaTeX commands that first load the `*.pstex` image, and then add all the text in the figure. Select the "special text" mode in xfig to enable LaTeX metacharacters. Use `\include{*.pstex_t}` to add such a figure in your document. (PDF equivalent: `*.pdftex` + `*.pdftex_t`)

- Command-line export tool (e.g., for Makefile): `fig2dev`

```
%.eps: %.fig
        fig2dev -L eps $< $@
%.pstex %.pstex_t: %.fig
        fig2dev -L pstex_t -p $*.pstex $< $*.pstex_t
        fig2dev -L pstex $< $*.pstex
%.pdftex %.pdftex_t: %.fig
        fig2dev -L pdftex_t -p $*.pdftex $< $*.pdftex_t
        fig2dev -L pdftex $< $*.pdftex
```

Other graphics tools: TikZ, pnmtops, Inkscape, MATLAB, R, gnuplot, Python+matplotlib

Exercise 1: Copy file `example.tex` from slide 9, run "`pdflatex example`" twice (why?), and then "`okular example.pdf &`" (Linux) to see the output.

Exercise 2: Read pages 1–64 of the LaTeX book, then write your CV with LaTeX, convert the result into PDF, and put it onto your homepage.

For information on how to set up homepages locally:
`https://help.uis.cam.ac.uk/website-resources/web-application-development/ds-web`
`http://www.srcf.ucam.org/`

Exercise 3: In a job interview for a position as a subeditor of a technical journal, your skills in spotting typographic mistakes made by LaTeX beginners are tested with this example text:

> The -7 dB loss ($\pm 2dB$) shown on pp. 7-9 can be attributed to the f(t)= $sin(2\pi ft)$signal , where t is the the time and f =48Khz is the "sampling frequency".

Can you spot all 14 mistakes? Write down both the probable original incorrect LaTeX source text, as well as a corrected version.