LaTeX

Joy Allen

Newcastle University

March 12, 2015

Why use LaTeX?

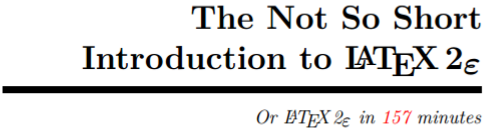
- Extremely useful tool for writing scientific papers, presentations and PhD Thesis which are heavily laden with mathematics
- You get lovely pretty documents
- Extremely satisfying
- Free software
- Lots of help on the internet

Why use LaTeX?

- Extremely useful tool for writing scientific papers, presentations and PhD Thesis which are heavily laden with mathematics
- You get lovely pretty documents
- Extremely satisfying
- Free software
- Lots of help on the internet
- Word and Powerpoint are painful...

Useful resources

- The Not So Short Introduction to LaTeX
  https://tobi.oetiker.ch/lshort/lshort.pdf



The Not So Short Introduction to LaTeX 2_{ε}

Or LaTeX 2_{ε} in 157 minutes

- A shorter, snappier tutorial:
  http://ece.uprm.edu/ caceros/latex/introduction.pdf

The Preamble

LaTeX needs to know the type of document the author wants to create. This is specified with the

```
\documentclass
```

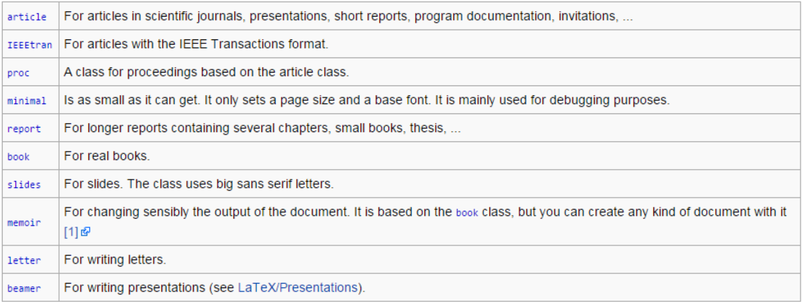
command.

```
\documentclass[options]{class}
```

is declared at the very beginning of every document. The ‘class’ specifies the type of document being created.

Document class types

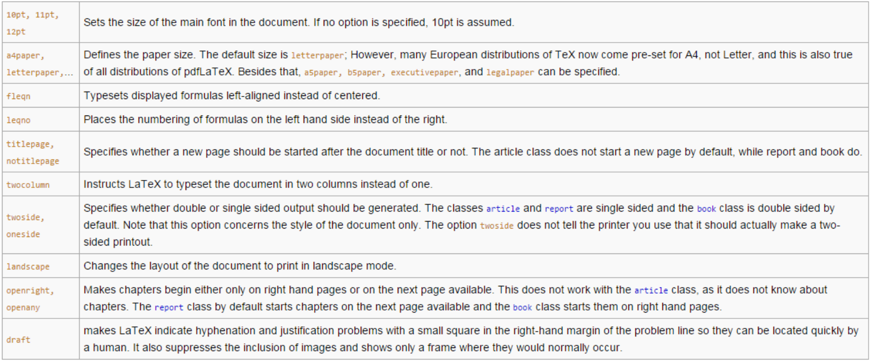
```
\documentclass[options]{class}
```



article	For articles in scientific journals, presentations, short reports, program documentation, invitations, ...
IEEEtran	For articles with the IEEE Transactions format.
proc	A class for proceedings based on the article class.
minimal	Is as small as it can get. It only sets a page size and a base font. It is mainly used for debugging purposes.
report	For longer reports containing several chapters, small books, thesis, ...
book	For real books.
slides	For slides. The class uses big sans serif letters.
memoir	For changing sensibly the output of the document. It is based on the book class, but you can create any kind of document with it [1]
letter	For writing letters.
beamer	For writing presentations (see LaTeX/Presentations).

Document class options

```
\documentclass[options]{class}
```



10pt, 11pt, 12pt	Sets the size of the main font in the document. If no option is specified, 10pt is assumed.
a4paper, letterpaper,...	Defines the paper size. The default size is letterpaper; However, many European distributions of TeX now come pre-set for A4, not Letter, and this is also true of all distributions of pdfLaTeX. Besides that, a5paper, b5paper, executivepaper, and legalpaper can be specified.
fleqn	Typesets displayed formulas left-aligned instead of centered.
leqno	Places the numbering of formulas on the left hand side instead of the right.
titlepage, notitlepage	Specifies whether a new page should be started after the document title or not. The article class does not start a new page by default, while report and book do.
twocolumn	Instructs LaTeX to typeset the document in two columns instead of one.
twoside, oneside	Specifies whether double or single sided output should be generated. The classes article and report are single sided and the book class is double sided by default. Note that this option concerns the style of the document only. The option twoside does not tell the printer you use that it should actually make a two-sided printout.
landscape	Changes the layout of the document to print in landscape mode.
openright, openany	Makes chapters begin either only on right hand pages or on the next page available. This does not work with the article class, as it does not know about chapters. The report class by default starts chapters on the next page available and the book class starts them on right hand pages.
draft	makes LaTeX indicate hyphenation and justification problems with a small square in the right-hand margin of the problem line so they can be located quickly by a human. It also suppresses the inclusion of images and shows only a frame where they would normally occur.

Example

Let's begin our document

```
\documentclass[a4paper, 11pt]{article}

\begin{document}

  Some writing here.

\end{document}
```

Example

Let's begin our document

```
\documentclass[a4paper, 11pt]{article}

\begin{document}

  Some writing here.

\end{document}
```

- Save this as a .tex file (e.g. example.tex)
- Make sure the editor compiler is pdfLaTeX
- Compile! (Press F1 to build and view)
- Congratulations on your first ever LaTeX document! The output is automatically saved as a .pdf file (example.pdf).

Getting Started

Most characters on the keyboard, such as letters and numbers, have their usual meaning. However the characters

\ { } $ ^ _ % ~ # &

are used for special purposes within LaTeX. Thus typing one of these characters will not produce the corresponding character in the final document. In particular, there are sequences of characters which begin with a ‘backslash’ which are used to produce mathematical symbols and Greek letters and to accomplish tasks such as changing fonts. These are known as control sequences.

Special characters

$	The special character $ is used when embedding mathematical expressions in paragraphs of ordinary text in order to change into and out of ‘mathematics mode’.
\	Most control sequences consist of a backslash followed by a string of (upper or lower case) letters. e.g `$\delta$` gives the greek letter delta $\delta$
{	The ‘braces’ { and } are used for grouping: the characters they enclose are treated as a single ‘group’, which can be specified as an ‘argument’ of a control sequence. E.g. `{\bf{Hello World}}` gives **Hello World**.
%	The special character % is used to introduce ‘comments’ into the input file that do not appear in the final document: all characters occuring after % on any line of the input file are ignored by LaTeX.

Special characters cont.

$\wedge$ and _	The special characters $\wedge$ and _ are used in mathematical expressions to produce superscripts and subscripts respectively.
#	The special character # is used to specify arguments in definitions of control sequences.
&	The special character & is used when typesetting tables in order to separate entries in different columns within the tabular or eqnarray.
$\sim$	The special character $\sim$ is used to create spaces.

In order to generate these characters within your document, you can combine them with a backslash to create a control sequence, however in a few cases you may need to include their 'name'. This must be done within mathmode.

```
\$, \backslash, \{, \%, \wedge, \_, \#, \&, \sim
```

$\$, \backslash, \{, \%, \wedge, \_, \#, \&, \sim$

Changing font style

Adding color

```
\documentclass[a4paper, 11pt]{article}

\begin{document}

 \textcolor{red}{Some writing here.}

\end{document}
```

Changing font style

Adding color

```
\documentclass[a4paper, 11pt]{article}

\begin{document}

 \textcolor{red}{Some writing here.}

\end{document}
```

When we try to compile this, we get an error. If you want to include graphics, colored text or source code from a file into your document, you need to enhance the capabilities of LaTeX. Such enhancements are called packages. Some packages come with the LaTeX base distribution. Others are provided separately. Modern TeX distributions come with a large number of packages pre-installed.

Packages

We link LaTeX to the packages we wish to use in the preamble.

```
\documentclass[a4paper, 11pt]{article}

\usepackage{xcolor}

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

Packages

We link LaTeX to the packages we wish to use in the preamble.

```
\documentclass[a4paper, 11pt]{article}

\usepackage{xcolor}

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

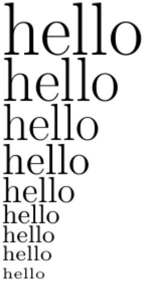
2829 packages in the MiKTeX repository: http://miktex.org/packages
You can see which packages are installed on your computer and what all is availabile by looking at the MiKTeX Package Manager via:

<Win> and r button.
open ‘mpm’

MiKTeX automatically installs the packages required if they are called for in the preamble.

Fonts

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



hello
hello
hello
hello
hello
hello
hello
hello
hello

Font

- Italic text

  ```
  \emph{text}
  ```

  text

- Bold Text

  ```
  {\bf{text}}
  ```

  text

- Underlining text

  ```
  \underline{text}
  ```

  text

Text Alignment

```
\begin{center}
  Text written in here will be centered.
\end{center}

\begin{flushleft}
  Text here will be left justified
  this default so no need to specify
  unless you have changed the default.
\end{flushleft}

\begin{flushright}
   Text here will be justified to the right.
\end{flushright}

\justifying
 Text following this will be fully justified.
 This command requires the package
 ragged2e to be installed.
```

Document Structure

```
\title{The title of my overall document}
\author{My name}
\date{} %Leave blank if you do not want a date added
        %Either specify the date you want
        %Or don't include at all  if you want todays date.

\begin{document}

 \maketitle or \titlepage

 \chapter{}  % only works in report or book class

 \section{Section Heading 1}
 Some writing in section one.

 \subsection{Subsection Heading 1.1}
 \subsubsection{Subsubsection Heading 1.1.1}

 Placing \tableofcontents after \begin{document} results
 in an autormatically generated table of contents
```

Math mode

Example equation:

```
\begin{equation}
 (i - \gamma)\hbar \frac {\partial \phi \bfrt}
 {\partial t} = \left( -\frac{\hbar ^2}{2m}\nabla^2
     + g|\phi\bfrt|^2 - \mu \right)\phi \bfrt \;.
\end{equation}
```

$$(i-\gamma)\hbar\frac{\partial\phi(\mathbf{r},t)}{\partial t} = \left(-\frac{\hbar^2}{2m}\nabla^2 + g|\phi(\mathbf{r},t)|^2 - \mu\right)\phi(\mathbf{r},t)\;.$$

Math mode, eqn evironment

Packages you need:

```
\usepackage{amsmath}
\usepackage{amssymb}
```

Equation environments, LaTeX automatically numbers your equations

```
\begin{equation}
    y = mx + c
\end{equation}
```

$$y = mx + c \tag{1}$$

If you don't want an equation number:

```
\begin{equation*}
y = mx + c
\end{equation*}
```

$$y = mx + c$$

Short for this is:

```
$$ y = mx + c$$
```

Math mode cont. Greek letters and eqn alignment

```
\begin{eqnarray}
A &=& \pi r^2 \nonumber \\
C &=& \pi d
\end{eqnarray}
```

$$\begin{aligned} A &= \pi r^2 \\ C &= \pi d \end{aligned} \tag{2}$$

We can do fractions, operator signs, bold text, roman text, large brackets...

Importing graphics into a LaTeXdocument

Need the package:

```
\usepackage{graphicx}
```

If compiling with pdfLaTeX

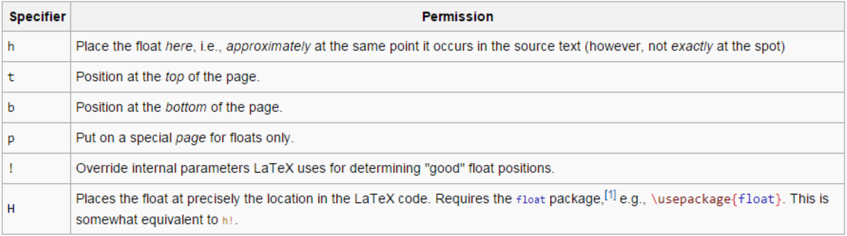
- .png .jpeg .pdf
- .eps may work if you follow with epstopdf - depending on your installation.

If compiling with latex (need to follow with dvipslatex ps2pdf latex).

- .eps

Figures in Latex

```
\begin{figure}[placement specifier]
\includegraphics[size]{name of figure}
\end{figure}
```



Specifier	Permission
h	Place the float *here*, i.e., *approximately* at the same point it occurs in the source text (however, not *exactly* at the spot)
t	Position at the *top* of the page.
b	Position at the *bottom* of the page.
p	Put on a special *page* for floats only.
!	Override internal parameters LaTeX uses for determining "good" float positions.
H	Places the float at precisely the location in the LaTeX code. Requires the `float` package,[1] e.g., `\usepackage{float}`. This is somewhat equivalent to `h!`.

Useful commands

- Taking a new line

  \\

- Bullet points

```
    \begin{itemize}
       \item  Item 1
       \item  Item 2
    \end{itemize}
```

Useful commands continued - Multicolumns

- Multicolumns

```
\usepackage{multicol}
\begin{document}
\begin{multicols}{2}
The text will be split over two columns.
When the first column is filled, the text
will continue into the second.
\end{multicols}
\end{document}
```

The text will be split over two columns. When the first column is filled, the text will continue into the second.

Useful commands continued - Minipages

```
\begin{minipage}{0.45\linewidth}
  Multicolumns are useful for a full document,
  but what if I am making slides and I know
  where I want to split my text?
\end{minipage}
\begin{minipage}{0.45\linewidth}
Or I want a figure on one side of
the page, and writing on the other? \\
\textcolor{mypurple}{Use minipages.}
\end{minipage}
```

Multicolumns are useful for a full document, but what if I am making slides and I know where I want to split my text?

Or I want a figure on one side of the page, and writing on the other?

Use minipages.

And much, much more

There is so much more you can do, you can customise LaTeX to suit your needs as you gain more experience. You can define your own colors, regularly used commands, change margins, put in movies....

More useful packages you may need..

```
\usepackage[a4paper]{geometry}
\usepackage{fancybox}
\usepackage{multimedia}
```