

L^AT_EX intro — enough to get you started

“Down the rabbit hole”

Joel Fenwick

7th April 2014

Outline

- 1 L^AT_EX?
 - What?
 - Why?
 - Basic document
 - Special characters
- 2 Floats
 - graphics
 - tabular
- 3 Maths
 - Examples
 - Displaymath examples
- 4 Demo
- 5 Misc bits
 - Misc

What is it?

L^AT_EX is:

- a computer *typesetting* system,
- ... which takes as plain text input,
- ... a mixture of content(text) and *markup*.

L^AT_EX is **not**:

- ... for desktop publishing or pixel precise layout.
- ... WYSIWYG
- ... an editor
- ... wonderfully communicative in its error messages

Why use it?

- Overall look of the end result.
- handles large documents well (eg `\include` and `\includeonly`).
- Maths notation
 - Ease of editing
 - Beautiful layout
 - treated as part of the text (with different rules).
 - macros
- Encourages separation of content from presentation.
 - less distraction
 - Flexible styling: change classes and macros in one place to change whole document.
- plain text
 - editable “anywhere”.
 - plays nice with version control software

Things you need

- A text editor.
- A L^AT_EX install. (eg: MikTeX on windows, MacTeX on MacOS)
- A way to view pdf files.

optional A GUI environment with buttons for the above.

Windows TexWorks — minimalistic but fine.

Windows TeXnic Center — more buttons but I don't like it as much

MacOS MacTex

Linux Lots. kile is the best one I've found.

optional “Latex Cheat Sheet” by Winston Chang.

There are typically two commands to run L^AT_EX: `latex` and `pdflatex`. Always use `pdflatex` (image format flexibility, and you most likely want to make a pdf anyway).

“Minimal” document

```
\documentclass[a4paper]{article}
% This is the preamble
\begin{document}
Hello
    World.
\end{document}
```

“Minimal” document

```
\documentclass[a4paper]
{article}
% This is the preamble
\begin{document}
Hello
    World.
\end{document}
```

`\` indicates a macro. “article” is the type of document we are creating.

`%` means the rest of the line is a comment.

`\begin` indicates the start of an “environment”. You must have a document environment.

Because latex won't create a completely empty document.

All environments must be ended properly.

Document classes

There are a number of standard document classes: `report`, `book`, `letter`, `article`.

Publishers and professional societies (Eg IEEE computer society) may have their own classes.

The document class sets the defaults for styling as well as determine some of the commands which are available.

Eg: letters do not have `\subsubsections`.

books have `parts`, `chapters`, `sections`, `...`, but not `abstracts`.

Special characters

The following have special meanings, be careful:

Symbols	Meaning	How to write them
\	start of a macro name	<code>\textbackslash</code>
%	start of a comment	<code>\%</code>
#	macro parameter	<code>\#</code>
{ }	Grouping text and markup to be treated as a unit. Compulsory params for macros.	<code>\{ \}</code>
\$	Starts/ends <i>math mode</i>	<code>\\$</code>
< >	Fine in math mode. Looks odd outside of math mode	<code>\$\$<\$</code> or <code>\textless</code> <code>\$\$>\$</code> or <code>\textgreater</code>
~	non-breaking space	

For the programmers out there, note that:

\ is **not an “escape character”**. \\ does not render as \ . nor does \< render as < .

Math mode Special characters

The following have special meaning in math mode.

- _ Indicates that the next thing should be subscripted.

Eg: x_2 is written as `x_2`.

If you want to have an underscore in your text, use `_` (or `\usepackage{underscore}`).

- ^ Indicates that the next thing should be superscripted.

Eg: x^2 is written as `x^2`.

If you want to write `^` in your text, you will need `\^{\}`

floats

A float is something which can be moved out of position in the document to help with layout. They can be `\label`-ed and `\ref`-ed elsewhere. There are two in latex:

- figure
- table

The both make use of `\caption`. There are macros `\listoffigures` and `\listoftables` to create table of contents for those things. (There is also `\tableofcontents`).

figure

```
\begin{figure}
%
\caption{About the figure}\label{fig:example}
\end{figure}
```

You will need `\usepackage{graphics}` in your preamble to load graphics.

```
\begin{figure}
\scalebox{0.5}{\includegraphics{picture.png}}
\caption{About the figure}\label{fig:example}
\end{figure}
```

Note that the `\label` must come after (or inside) `\caption` or the numbers will be wrong.

actually making a table

```

\begin{tabular}{|c|p{6cm}|c|}\hline
\textbf{Num} & \textbf{Text} & \textbf{Num2} \\ \hline
1 & \ldots & 5 \\ \hline
2 & Middle column wraps because of p column & 6 \\ \hline
3 & short text & - \\ \hline
\end{tabular}

```

Num	Text	Num2
1	...	5
2	Middle column wraps because of p column	6
3	short text	-

table environment

```
\begin{table}
\begin{tabular}{|c|p{6cm}|c|}\hline
\textbf{Num} & \textbf{Text} & \textbf{Num2} \\ \hline
...
\end{tabular}
\caption{An example of a table}\label{tab:ex}
\end{table}
```

Num	Text	Num2
1	...	5
2	Middle column wraps because of p column	6
3	short text	-

Table: An example of a table

Num	Text	Num2
1	...	5
2	Middle column wraps because of p column	6
3	short text	-

Table: An example of a table

For an example of a table, see `Table~\ref{tab:ex}`.

For an example of a table, see Table 2.

You may need to run Latex more than once to get these correct — see demo

Maths examples:

xyz \$xyz\$

\$n\log_n \sum_0^1\$

\$A^{b^c}\$

\$x \in \{1\ldots \pi\}\$

$\frac{\partial y}{\partial x}$

\$\int_a^b F(x) dx\$

xyz xyz

$n \log_n \sum_0^1$

A^{b^c}

$x \in \{1 \dots \pi\}$

$\frac{\partial y}{\partial x}$

$\int_a^b F(x) dx$

Display Maths examples:

```
\[\int_a^b\int_c^d \]
```

```
\begin{equation}
```

```
\tau = 2\cdot\pi
```

```
\end{equation}
```

$$\int_a^b \int_c^d$$

$$\tau = 2 \cdot \pi$$

(1)

Display Maths examples:

```
\begin{eqnarray*}
F(x) &=& x^2+2xy+y^2 \\
&=& (x+y)^2
\end{eqnarray*}
```

$$\begin{aligned} F(x) &= x^2 + 2xy + y^2 \\ &= (x + y)^2 \end{aligned}$$

Demo

fragile - the irregular verbs of latex

Some macros are *fragile* this means they may have problems if they are placed in a *moving argument*. eg: in a section title. You can fix this with `\protect` if really needed.

spaces after macros

Suppose the following is in the preamble:

```
\newcommand{\python}{\texttt{Python}}
```

You would expect:

```
\python is a programming language.
```

to look like this:

```
Python is a programming language.
```

Instead you will see:

```
Pythonis a programming language.
```

This is because latex is using the space between `\python` and `is` as the end of the macro name. If the a macro name is followed by punctuation this is not a problem. You could also write:

```
\python\ is
```

or

```
\python{} is
```

grouping with $\{ \}$

$\widehat{A}bc$ $\widehat{A}bc$

$\sqrt{7}abc$ $\sqrt{7}abc$

Normal text $\emph{special text}$ normal text

Normal text $\emph{special text}$ normal text

$\widehat{A}bc$ $\widehat{A}bc$

$\sqrt{7}abc \neq \sqrt{7}abc$

Normal text *special text* normal text

Normal text *special text* normal text

Accents, en/em-dash and quote marks

Accents:

```
\ "A \ "u \ ~o \ `e \ 'e
```

Ä ü ã è é

Dashes:

- - : co-ordinates (hyphen).
- -- : 1–12 (range) (en-dash)
- --- : that is — a chicken (em-dash)

Quotes:

```
`Abc' ``Abc" ``Abc''
```

‘Abc’ “Abc” “Abc”

NOT "Abc"

verbatim and verb

For producing listings, consider the listings package. But for simple stuff:

```
\verb|\documentclass{article}| →  
\documentclass{article}
```

```
\begin{verbatim}  
\documentclass{article}  
\begin{document}  
\end{document}  
\end{verbatim}
```

```
\documentclass{article}  
\begin{document}  
\end{document}
```


Not appearing in this film

BibTeX

badness in warnings