

# A Beginner's Guide to L<sup>A</sup>T<sub>E</sub>X

David Xiao  
dxiao@cs.princeton.edu

September 12, 2005

## 1 Introduction

L<sup>A</sup>T<sub>E</sub>X is *the* standard mathematical typesetting program. This document is for people who have never used L<sup>A</sup>T<sub>E</sub>X before and just want a quick crash course to get started. I encourage all students in mathematics and theoretical computer science to learn L<sup>A</sup>T<sub>E</sub>X so you can use it to typeset your problem sets; your TA's will love you if you do.

For a more comprehensive introduction, check out  
<http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf>.

## 2 How to find and use L<sup>A</sup>T<sub>E</sub>X?

There are Windows packages for L<sup>A</sup>T<sub>E</sub>X, though typically I prefer to use it on a Unix system (e.g. OS X, Linux, Solaris). You will find that most university clusters have L<sup>A</sup>T<sub>E</sub>X installed. If you prefer, you can install it on your home system; it is easy to find via Google or your search engine of choice. I will assume that you are using a Unix system for the rest of the guide.

I recommend using an editor like `emacs` or `vi` to edit L<sup>A</sup>T<sub>E</sub>X. If you are uncomfortable with these, then any plaintext editor will do. `emacs` is particularly nice because it has a built-in L<sup>A</sup>T<sub>E</sub>X-mode that does text highlighting and indentation.

## 3 Basic rules

Basic L<sup>A</sup>T<sub>E</sub>X is just text with typesetting commands. Typesetting commands are usually preceded by “\”, and any arguments are usually placed inside curly braces “{}”.

L<sup>A</sup>T<sub>E</sub>X wraps text in adjacent lines as if they were part of the same paragraph. To start a new paragraph, insert an extra “return”:

Source:  
This is one paragraph.  
  
This is another.

Output:

This is one paragraph. This is another.
--

To get a newline without starting a new paragraph, use `\\`.

To get a comment, use the percent sign `%` at the beginning of a line. The rest of that particular line will be commented out.

## 4 Starting a new document

The most basic (empty) document has only three parts:

```
\documentclass{article}

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

To start a new document, you can just take the L<sup>A</sup>T<sub>E</sub>X file for this document and delete the stuff between the begin and end document commands, leaving the `\maketitle` command (this prints out the title and your name). You will see before the `\begin{document}` that there are commands for the title and author of the document. Change the names between the curly braces to the name of problem set and your name. (The other stuff that precedes `\begin{document}` is useful stuff that you don't need to worry about for now, just leave it as is.) Then, save the file under a different name, for example `problem-set-1.tex`.

## 5 Compiling

Suppose our file is named `mylatexfile.tex`. To compile it, simply invoke `latex mylatexfile.tex` in your Unix shell. This will compile the file, assuming there are no errors.<sup>1</sup> If there are errors, you can quit the compiler by hitting “x” and then enter. Unfortunately L<sup>A</sup>T<sub>E</sub>X compiler errors are very unhelpful in determining the nature of the problem, but they usually correctly point you to the line where the error occurred.

Once it successfully compiles, you will get a file named `mylatexfile.dvi`. Typically we convert this either into a Postscript or PDF file, which may be done by the programs `dvips mylatexfile.dvi` and `dvipdf mylatexfile.dvi`. Sometimes `dvips` requires an extra option: `-o output-file-name` specifying the output file.

## 6 Organization

One important thing to do is to organize your document well.

### 6.1 Sectioning

There are two sectioning commands that will be useful for you: `\section{Name of section}` and `\subsection{Name of subsection}`. Use these to separate different problems or subproblems in the assignment.

### 6.2 Tables

You can put stuff into tables by using the `tabular` environment. For example:

---

<sup>1</sup>You may have to invoke `latex` twice if you are using labels and references. See Section 6.4.

Source:

```
\begin{tabular}{r|cl}
1st column & 2nd column & 3rd column\\
\hline
a & b & c
\end{tabular}
```

Output:

1st column	2nd column	3rd column
a	b	c

Note that the command is called `tabular` and *not* `table`. Important points:

- The “`{r|cl}`” after the tabular `\begin{tabular}` indicate the alignment of the three columns: right, center, and left. This is mandatory as it specifies the layout of the table. For more columns, type more alignment commands, e.g. for a table with 5 columns all aligned to the right, you would use `rrrrr`.
- The vertical bar `|` between the `r` and `c` indicates that a vertical line should be drawn between those columns.’
- The `&` separates the columns in the body of the table.
- A `\\` signifies the end of each line of the table.
- The command `\hline` means that a horizontal line should be inserted.

### 6.3 Lists

You can put stuff into ordered and unordered lists by using the `enumerate` and `itemize` commands, respectively. For example:

Source:

Unordered list:

```
\begin{itemize}
\item This is one item.
\item This is another.
\end{itemize}
```

Ordered list:

```
\begin{enumerate}
\item This is the first item.
\item This is the second.
\end{enumerate}
```

Output:

Unordered list:

- This is one item.
- This is another.

Ordered list:

1. This is the first item.
2. This is the second.

### 6.4 Labels and references

It is useful to refer to the section number at times. This may be done by using the `\label{labelname}` command. Place this right after you start a section. Then, you may refer to the section number by using `\ref{labelname}`. This will also be useful to refer to math equations.

Note that  $\text{\LaTeX}$  creates and uses a bunch of auxiliary files. Thus you will have to invoke it twice to compile a file that has labels and references, or if those labels and references have changed since the last compilation.

## 7 Math

The reason we use L<sup>A</sup>T<sub>E</sub>X is because it is so powerful in typesetting mathematical expressions. It wins hands down versus word processors like Word.

### 7.1 Math mode

Math expressions are separate from text in L<sup>A</sup>T<sub>E</sub>X. To enter a math environment in the middle of text, use the dollar sign \$, for example `$F = ma$` produces  $F = ma$ . Everything between the two \$ signs will be considered math formula.

To type a math expression that is on its own line and centered, use `$$`:

Source:

```
The following is an important equation:  
$$E = mc^2$$
```

Output:

```
The following is an important equation:  

$$E = mc^2$$

```

To give an equation a number and have it referable, use the `equation` environment and use a `\label` command:

Source:

```
The following is an important equation:  
\begin{equation}  
\label{emc}  
E = mc^2  
\end{equation}  
Please memorize Equation \ref{emc}.
```

Output:

```
The following is an important equation:  

$$E = mc^2 \tag{7.1}$$
  
Please memorize Equation 7.1.
```

To typeset several equations together and have them properly aligned, use the `align` environment:

Source:

```
Some important equations:  
\begin{align}  
\label{einstein}  
E & = mc^2 \\  
\label{newton}  
F & = ma \\  
\label{euler}  
e^{i \pi} & = -1  
\end{align}
```

Output:

```
Some important equations:  

$$E = mc^2 \tag{7.2}$$
  

$$F = ma \tag{7.3}$$
  

$$e^{i\pi} = -1 \tag{7.4}$$

```

The equations are aligned along the `&` and each line is terminated by `\\`. To suppress the equation numbering (i.e. if the equations won't be referred to) use `align*` instead of `align`.

### 7.2 Writing math expressions

I will only go over a few common mistakes and hard-to-find expressions regarding how to write math expressions. It is quite intuitive otherwise, you can figure most things out quickly with trial and error. All expressions in math mode may be nested within each other arbitrarily.

- Superscript and subscript are done using `^` and `_` characters. Note that if you want multiple characters in the super/subscript then you need to surround them with curly braces: `$e^i\pi = -1$` gives  $e^i\pi = -1$  whereas `$e^{i\pi} = -1$` gives  $e^{i\pi} = -1$ .

- Fractions are done using  $\frac{1}{2}$  which gives  $\frac{1}{2}$ .
- To do a binomial coefficient, use  $\binom{n}{k}$  which gives  $\binom{n}{k}$ .
- Modular arithmetic can be written using the  $\pmod{n}$  and  $\bmod{n}$  commands. The first puts parentheses and a lot of space around the mod and the second does not.
- $\forall$  and  $\exists$  are written as  $\forall$  and  $\exists$ .
- $\neq$ ,  $\geq$ , and  $\leq$  are  $\neq$ ,  $\geq$ , and  $\leq$ .
- $\cdot$  (e.g. for multiplication) is  $\cdot$ .
- $\circ$  is  $\circ$ .
- $\cup$  and  $\cap$  are  $\cup$  and  $\cap$ .
- $\oplus$  is written with  $\oplus$ .
- Large  $\cup$ ,  $\cap$ ,  $\oplus$  signs that behave like summations (see below for summations) are written as  $\bigcup$ ,  $\bigcap$ ,  $\bigoplus$ .
- $\leftarrow_{\mathbb{R}}$  is produced with  $\leftarrow_{\mathbb{R}}$ .
- $\mathbb{Z}$ ,  $\mathbb{R}$ , etc. are produced using  $\mathbb{Z}$ ,  $\mathbb{R}$ , etc.
- $\mathbb{E}$  is produced with  $\mathbb{E}$ .
- $\mathbb{P}$ ,  $\mathbb{NP}$ , etc. are produced using  $\mathbb{P}$ ,  $\mathbb{NP}$ , etc.
- $\mathcal{P}$  is  $\mathcal{P}$ .
- $\ell$  (as opposed to  $l$ ) is produced with  $\ell$ .
- $\{ \}$  are done with  $\{$  and  $\}$ .
- $\approx$  is produced with  $\approx$ .
- $\hat{x}$  and  $\bar{x}$  are done with  $\hat{x}$  and  $\bar{x}$ . A longer bar may be written using  $\overline{\text{SAT}}$ , which produces  $\overline{\text{SAT}}$ .
- $\varepsilon$  (as opposed to  $\epsilon$ ) may be written with  $\varepsilon$ .
- $\in$  is written as  $\in$ .
- $\notin$  is written as  $\notin$ .
- Negations may be done with  $\text{not}$ , for example
- $\{0, 1\}$  is abbreviated as  $\{0, 1\}$ .  $\text{not} \geq$  gives  $\not\geq$ .
- The probability sign is defined as  $\Pr$ , i.e.  $\Pr$ .
- Encryption and decryption are  $\text{Enc}$  and  $\text{Dec}$ , which give  $\text{Enc}$  and  $\text{Dec}$ .
- $\text{poly}$  is written as  $\text{poly}$ .
- To draw parentheses that grow to match the contents, use  $\left$  to precede the left parenthesis and  $\right$  to precede the right:  
Source:

`$$\Pr\left[\sum_{i=1}^k X_i > c \right] \leq 2^{-\Omega(c^2 k)}$$`

Output:

$$\Pr \left[ \sum_{i=1}^k X_i > c \right] \leq 2^{-\Omega(c^2 k)}$$

- Arrays are like tables, except they must be used in place of tables when in math mode: instead of using `\begin{tabular}` and `\end{tabular}` use `\begin{array}` and `\end{array}`. Again, you must give a column specification for how the columns are to be laid out.
- Spacing is very different in math mode so text in the middle of a formula is set strangely. If you want to have text in the middle of the formula, use the `\text{some text}` command. For example, `$$\mathbf{P} \neq \mathbf{NP} \text{ implies that } \mathbf{SAT} \notin \mathbf{P}$$` produces  $\mathbf{P} \neq \mathbf{NP}$  implies that  $\mathbf{SAT} \notin \mathbf{P}$ .
- Summations and products are done using `\sum` and `\prod` respectively. Parameters can be given for the summation/product as well:

Source:

`$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$$`

Output:

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$$

- Piecewise functions may be defined using the `piecewise` environment:

Source:

`$$f(x) = \begin{cases} 1 & x = 0 \\ 0 & \text{else} \end{cases}$$`

Output:

$$f(x) = \begin{cases} 1, & x = 0 \\ 0, & \text{else} \end{cases}$$

- You may define new commands using `\newcommand{\commandname}{definition}`. This is essentially a macro, so that whenever `\commandname` appears, the text of `definition` is inserted in its place. If `\commandname` is already taken, either use a different name or use `\renewcommand{...}{...}` to overwrite the old definition.
- There are many many other symbols available. You can search for “latex symbols” online and come up with the references.