

How To LaTeX

Adam Blank

February 12, 2013

Contents

1	What is \LaTeX?	3
2	Installation and Editors	3
2.1	Windows	3
2.2	Mac	3
2.3	Linux	3
3	Math Mode	3
4	Some \LaTeX Quirks and Misconceptions	5
4.1	Whitespace	5
4.2	Escape Characters?	6
4.3	“Quotes”	6
4.4	\imath \jmath	6
5	How Do I Make that Symbol?	7
5.1	Basic Math	7
5.2	Logical Symbols	8
5.3	Set Symbols	8
5.4	Sequences, Summations, and Products	8
5.5	Text Styles	9
5.6	Piecewise Functions	9
5.7	Still Can’t Find It?	9
6	Some Useful Environments	10
6.1	Centering Text	10
6.2	Tables	10
6.3	Math Align	11
6.4	a, b, c, 1, 2, 3	11
6.5	Bullet Points	12
6.6	Custom Lists	12
6.7	Matrices	13
6.8	Code and WYSIWYG	13
7	Using the Homework Template	14

1 What is \LaTeX ?

\LaTeX is a typesetting language that is often used in technical fields like Computer Science, Mathematics, Chemistry, and Physics. It is particularly useful to easily typeset math (like $\emptyset \subseteq \mathbb{N}$) and chemical formulae (like H_2O).

Unfortunately, \LaTeX has a non-trivial learning curve associated with it. This document is intended to lessen your pain in learning how to easily typeset beautiful documents for the rest of your technical career. It is this author's belief that once you learn how to use \LaTeX , editors like Microsoft Word become obsolete, even for documents like English papers.

2 Installation and Editors

2.1 Windows

You should download MiKTeX from <http://miktex.org>. TeXnicCenter is a good editor to use to write \LaTeX documents. You can download it here: <http://www.texniccenter.org/>.

2.2 Mac

You should download MacTeX from <http://www.tug.org/mactex/2012/>. An editor comes with the distribution which you can use; alternatively, most of the common editors for OSX will support writing \LaTeX documents.

2.3 Linux

If you're using Debian or Ubuntu, try `sudo apt-get install texlive`. Otherwise, you most likely know your system well enough to figure it out.

3 Math Mode

One of the most useful things that \LaTeX offers is the so-called "Math Mode." Let's just begin with an example. If you type the following snippet of code:

```
1 Let  $M$  be the number of moons in the Universe, and let  $E$  be the number of
2 moons Earth has. We know from Elementary School that  $E=1$ , but what can we
3 say about  $M$ ? Since Earth is a part of the Universe, we know that  $M \geq E$ .
4 Furthermore, Jupiter has at least one moon; so,  $M > E$ .
```

\LaTeX parses it as:

Let M be the number of moons in the Universe, and let E be the number of moons Earth has. We know from Elementary School that $E = 1$, but what can we say about M ? Since Earth is a part of the Universe, we know that $M \geq E$. Furthermore, Jupiter has at least one moon; so, $M > E$.

Isn't that neat? The $\$$'s allow us to seamlessly switch between using Math and English. Sometimes, however, we want to isolate the math to display how important it is. This \LaTeX mode is called

“Display Mode” and is just as easy to use. Instead of a single \$, just double them up.¹ Here’s an example:

```
1 Now that we have determined that $M > E$, we would like to find a simple
2 unification result for physics. To do this, we must recall Newton’s famous
3 formula: $$F = ma$$
4 You see, once we realize its significance, the rest is all simple arithmetic.
```

And the output:

Now that we have determined that $M > E$, we would like to find a simple unification result for physics. To do this, we must recall Newton’s famous formula:

$$F = ma$$

You see, once we realize its significance, the rest is all simple arithmetic.

By using display mode, we made several things happen: (1) we entered math mode and (2) we asked L^AT_EX to make sure the content was spaced nicely. You should be careful with math mode and display mode, because, while L^AT_EX will try its best to not let the math run off the side of the page, there are some times when it just doesn’t know what else to do:

```
$$a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s +
t + u + v + w + x + y + z + 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 +
12 + 13 + 14 + 15$$
```

```
a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+x+y+z+0+1+2+3+4+5+6+7+8+9+10+11+
```

In such a situation, the correct thing to do is use the `align*` environment which is discussed below.

Occasionally, typesetting something requires being in math mode, but we don’t want *alltheletterstoruntogether* or be in the math *font*. One good example of this is chemical formulae. They require us to typeset superscripts and subscripts, but we want normal letters. How do we fix this?² We use the `text` command! Here’s an example:

```
1 Little Johnny was a chemist.\\
2 He isn’t any more.\\
3 For what he thought was $\text{H}_2\text{O}$ was $\text{H}_2\text{SO}_4$.
```

And the output:

Johnny was a little chemist.
He isn’t any more.
For what he thought was H₂O was H₂SO₄.

That completes the basics of typesetting math in L^AT_EX ; now, we move on to some quirks that are often “gotchas” to people using L^AT_EX for the first time.

1 The L^AT_EX purist might yell at me here. Using \$\$ is not actually the accepted way to enter and leave display mode. It is in fact deprecated, and, instead, you can delimit it by using `\[` and `\]`.
2 It is worthy of note that `H$_2$SO$_4$` would also work, but `text` is also useful in other places.

4 Some L^AT_EX Quirks and Misconceptions

One of the common misconceptions that people using L^AT_EX for the first time have is that they think of it as more like Microsoft Word than Java. L^AT_EX is, first and foremost, a programming language, and you should think of it that way. The way L^AT_EX works is that you describe how you want your output to look and L^AT_EX figures out what it thinks will look best given your constraints.

4.1 Whitespace

As a direct result of L^AT_EX's methodology, it is not very sensitive to whitespace. Think about when the last time Java cared if you put one or two spaces between `public` and `void` was. Take a look:

```

1 I      like      my words  to have      varied              spaces
2 between them.  I also
3 like
4     lots
5         of
6             new
7             lines.

```

Note how L^AT_EX ignores my tastes in spacing:

```
I like my words to have varied spaces between them. I also like lots of new lines.
```

The good news is that if you are really insistent, L^AT_EX will let you do it:

```
A\;B\;\;\;C\quad D\qquad E
```

```
A B C D E
```

```
You can\\force new lines\\ as well.
```

```
You can
force new lines
as well.
```

If you put a blank line between two paragraphs, L^AT_EX assumes that they are paragraphs and will correctly indent the next one. Lastly, to tell L^AT_EX you want a new page use the `newpage` command. That is, type `\newpage`.

4.2 Escape Characters?

Like most programming languages, \LaTeX has a bunch of characters that must be “escaped” to be written literally, like $\backslash n$ in Java or C. Here is a list of common characters, their uses in \LaTeX and how to write them literally:

Character	Usage in LaTeX	Literal
\backslash	Signifies that what is coming next is a command	$\text{\textbackslashbackslash}$
$\$$	Enter/exit math mode	$\text{\textbackslash\$}$
$\#$	Macro arguments	\textbackslash\#
$\&$	Delimits cells in tabulars	\textbackslash\&
$\%$	Comments out the rest of the line after it	\textbackslash\%
$-$	Displaying subscripts	\textbackslash_
\wedge	Displaying superscripts	$\text{\textbackslashverb!^!}$
$\{$	Delimits various environment and macro bodies	$\text{\textbackslash\{}$
$\}$	Delimits various environment and macro bodies	$\text{\textbackslash\}}$

4.3 “Quotes”

This oddity is easier to show than explain:

```
Using "double quotes" and 'single quotes' will not come out like you expect.
```

And the output:

```
Using "double quotes" and 'single quotes' will not come out like you expect.
```

Instead, to get the right output, use backticks, which are on the key to the left of ‘1’, to open quotes and single quotes to close them.

4.4 \imath \jmath

If you are seeing \imath or \jmath randomly in your output, you probably meant to use $<$ or $>$. To display either of those you must be in math mode.

5 How Do I Make that Symbol?

This section is intended to be a reference for common symbols and constructs that you will probably use fairly often.

5.1 Basic Math

Description	Example	L ^A T _E X Code
Composition	\circ	<code>\circ</code>
Superscript	x^{n+3}	<code>x^{\{n+3\}}</code>
Subscript	y_{n+4}	<code>y_{\{n+4\}}</code>
Fraction	$\frac{a+b}{c+d}$	<code>\frac{\{a+b\}\{c+d\}}</code>
Binomial Coefficient	$\binom{n}{k}$	<code>\binom{\{n\}\{k\}}</code>
Greek Letter	δ, Δ	<code>\delta</code> , <code>\Delta</code>
Greek Letters (Styled)	ε	<code>\varepsilon</code>
Large Parentheses	$d\left(\frac{a-b}{c}\right)$	<code>d\left(\frac{\{a-b\}\{c\}\right)</code>
Modular Equivalence	$x \equiv 1 \pmod{5}$	<code>x \equiv 1 \pmod{5}</code>
Infinity	∞	<code>\infty</code>
\leq	$x \leq y$	<code>x \leq y</code>
\geq	$x \geq y$	<code>x \geq y</code>
\neq	$x \neq y$	<code>x \neq y</code>
\langle, \rangle	$\langle 1, 2 \rangle$	<code>\langle 1, 2 \rangle</code>
\lfloor, \rfloor	$\left\lfloor \frac{1}{2} \right\rfloor$	<code>\left\lfloor \frac{\{1\}\{2\}}{\right\rfloor</code>
\lceil, \rceil	$\left\lceil \frac{1}{2} \right\rceil$	<code>\left\lceil \frac{\{1\}\{2\}}{\right\rceil</code>
\implies	$x \implies y$	<code>x \implies y</code>
\iff	$x \iff y$	<code>x \iff y</code>
\times	$A \times B$	<code>A \times B</code>
\cdot	$A \cdot B$	<code>A \cdot B</code>

5.2 Logical Symbols

Description	Example	L ^A T _E X Code
And	$p \wedge q$	<code>p \wedge q</code>
Equivalence	$p \Leftrightarrow q$	<code>p \Leftrightarrow q</code>
Implication	$p \Rightarrow q$	<code>p \Rightarrow q</code>
Not	$\neg p$	<code>\neg p</code>
Or	$p \vee q$	<code>p \vee q</code>
Quantifiers	\forall, \exists	<code>\forall , \exists</code>

5.3 Set Symbols

Description	Example	L ^A T _E X Code
Blackboard Style	\mathbb{N}	<code>\mathbb{N}</code>
Difference	$A \setminus B$	<code>A \setminus B</code>
Element Of	$x \in A$	<code>x \in A</code>
Empty Set	\emptyset	<code>\emptyset</code>
Intersection	$A \cap B$	<code>A \cap B</code>
Not Element Of	$x \notin A$	<code>x \notin A</code>
Subset	$A \subseteq B$	<code>A \subseteq B</code>
Superset	$B \supseteq A$	<code>B \supseteq A</code>
Union	$A \cup B$	<code>A \cup B</code>

5.4 Sequences, Summations, and Products

Description	Example	L ^A T _E X Code
Centered Ellipses	$1 + 2 + \cdots + n$	<code>1 + 2 + \cdots + n</code>
Ellipses	$1, 2, \dots, n$	<code>1, 2, \dots, n</code>
Product	$\prod_{k=0}^n k^2$	<code>\prod_{k=0}^n k^2</code>
Summation	$\sum_{k=0}^n k^2$	<code>\sum_{k=0}^n k^2</code>

5.5 Text Styles

Description	Example	L ^A T _E X Code
Blackboard Style	\mathbb{N}	<code>\mathbb{N}</code>
Bold	OMG	<code>\textbf{OMG}</code>
Code	<code>int counter</code>	<code>\texttt{int counter}</code>
Italics	<i>OMG</i>	<code>\emph{OMG}</code>
Script Style	\mathcal{R}	<code>\mathcal{R}</code>
Truth Values	T, F	<code>\textsf{T}</code> , <code>\textsf{F}</code>

5.6 Piecewise Functions

```


$$\text{abs}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$


```

$$\text{abs}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

5.7 Still Can't Find It?

There is a great tool online called detexify. You draw the symbol you are looking for, and it gives you the L^AT_EX command that it thinks most closely represents it. You can find it at <http://detexify.kirelabs.org/classify.html>.

6 Some Useful Environments

6.1 Centering Text

```
\begin{center}[\langle options \rangle] \langle environment contents \rangle \end{center}
```

This environment is used for centering a section of your document.

```
\begin{center}
  This text is centered. I am a silly example.\\
  She sells sea shells by the sea shore.
\end{center}
```

This text is centered. I am a silly example. She sells sea shells by the sea shore.
--

6.2 Tables

```
\begin{tabular}[\langle options \rangle] \langle environment contents \rangle \end{tabular}
```

options	usage
column horizontal alignment	c,l,r separated by for grid lines

This environment nicely typesets tables. It allows you to choose where (and if) you want grid lines in your table.

```
\begin{tabular}{l|c|r}
First Column & Second Column & Third Column\\
\hline
Cell 1,0 & Cell 1,1 & Cell 1,2\\
Cell 2,0 & Cell 2,1 & Cell 2,2
\end{tabular}
```

First Column	Second Column	Third Column
Cell 1,0	Cell 1,1	Cell 1,2
Cell 2,0	Cell 2,1	Cell 2,2

6.3 Math Align

```
\begin{align*}[\langle options \rangle] \langle environment contents \rangle \end{align*}
```

This environment allows you to write out aligned columns of text and math. It is particularly useful for writing out proofs with justifications to the right. Note the *** on the end of the name. Also note that each row of the *align** can have an arbitrary number of *&* signs which are alignment points.

```
\begin{align*}
\neg ((p \vee \neg q) \vee (q \vee \neg p)) &\iff & (\neg p \wedge q) \wedge (\neg q \wedge p) \\
&\iff & (\neg q \wedge \neg p) \wedge (\neg p \wedge q) \quad \&\& \text{[DeMorgan's Laws]} \\
&\iff & \neg p \wedge \neg q \wedge p \wedge q \quad \&\& \text{[Associativity and Commutativity]} \\
&\iff & \text{F} \quad \&\& \text{[Any value of } p \text{ will lead to F]} \\
\end{align*}
```

$\neg((p \vee \neg q) \vee (q \vee \neg p))$	\iff	$(\neg p \wedge q) \wedge (\neg q \wedge p)$	[DeMorgan's Laws]
	\iff	$\neg p \wedge p \wedge \neg q \wedge q$	[Associativity and Commutativity]
	\iff	F	[Any value of p will lead to F]

6.4 a, b, c, 1, 2, 3

```
\begin{enumerate}[\langle options \rangle] \langle environment contents \rangle \end{enumerate}
```

options	default	usage
item display	1.	Use this argument to change the appearance of the item. You can use 1, i, a, etc.

This environment allows you to typeset numbered or lettered lists. Make sure to use `\usepackage{enumerate}` if you want to change the look of the *item display*.

```
\usepackage{enumerate} %Put this line at the beginning of your document
\begin{enumerate}[(a)]
\item This is the first item.
\item This is the second item.
\item This is the third item.
\end{enumerate}
```

- (a) This is the first item.
- (b) This is the second item.
- (c) This is the third item.

6.5 Bullet Points

```
\begin{itemize}[\langle options \rangle] \langle environment contents \rangle \end{itemize}
```

This environment allows you to typeset bullet pointed lists.

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

- This is the first item.
- This is the second item.
- This is the third item.

6.6 Custom Lists

```
\begin{description}[\langle options \rangle] \langle environment contents \rangle \end{description}
```

This environment allows you to typeset lists with completely custom items displays. This is really nice for induction proofs.

```
\begin{description}
\item[First Item.] This is the first item.
\item[Second Item.] This is the second item.
\item[Third Item.] This is the third item.
\end{description}
```

First Item. This is the first item.

Second Item. This is the second item.

Third Item. This is the third item.

6.7 Matrices

```
\begin{bbmatrix}[\langle options \rangle] \langle environment contents \rangle \end{bbmatrix}
```

This environment allows you to typeset matrices. Note that the matrix must be in math mode!

```
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

6.8 Code and WYSIWYG

```
\begin{verbatim}[\langle options \rangle] \langle environment contents \rangle \end{verbatim}
```

This environment allows you to typeset words in a WYSIWYG fashion. Note that it will ignore all tabs, so you will want to use spaces instead.

```
\begin{verbatim}
  I want this text
    to look
      exactly as I put it.
def hello():
  return "Goodbye"
\end{verbatim}
```

```
  I want this text
    to look
      exactly as I put it.
def hello():
  return "Goodbye"
```

7 Using the Homework Template

```
\begin{question}[\langle options \rangle] \langle environment contents \rangle \end{question}
```

options	usage
question name	The name of the question being answered.

You should use this environment to typeset the answers to the questions we assign you. Each answer should be wrapped in this environment.

```
\begin{question}{This is a test question}
This is a solution to the test question
\end{question}
```

0: This is a test question
This is a solution to the test question

```
\begin{part}[\langle options \rangle] \langle environment contents \rangle \end{part}
```

You should use this environment to surround the individual parts of questions we assign you.

```
\begin{question}{This is a test question}
\begin{part}
This is the solution to the first part
\end{part}
\begin{part}
This is a solution to the second part
\end{part}
\end{question}
```

1: This is a test question
(a) This is the solution to the first part
(b) This is a solution to the second part

We complete this tutorial with a real-world example:

```

1 \begin{question}{Little Gauss' Formula}
2   \begin{part}
3     We use small examples to attempt to derive a formula for the sum of
4     first  $n$  numbers. Listing out the first few examples, we see the
5     following:
6     \begin{center}
7       \begin{tabular}{|c|r|c|}
8         \hline
9          $n$  & \multicolumn{2}{c|}{Sum} \\ \hline
10        0 & 0 & 0 \\ \hline
11        1 & 0+1 & 1 \\ \hline
12        2 & 0+1+2 & 3 \\ \hline
13        3 & 0+1+2+3 & 6 \\ \hline
14        4 & 0+1+2+3+4 & 10 \\ \hline
15      \end{tabular}
16    \end{center}
17    Clearly, this increases more quickly than a linear function; so,
18    we guess that it might be quadratic. A function that matches
19    these values and is quadratic is  $\frac{n(n+1)}{2}$ . That is,
20    we guess that the following equation holds:
21    
$$\sum_{i=0}^n i = \frac{n(n+1)}{2}$$

22  \end{part}
23  \begin{part}
24    Now, we prove the equality using induction as follows:
25    \begin{description}
26      \item[Base Case.]  $n=0$ :  $0 = \frac{0(1)}{2} = 0$ .
27      \item[Induction Hypothesis.] Assume the equality holds for some
28         $k \geq 0$ .
29      \item[Induction Step.] For  $n=k+1$ , we have
30        
$$\sum_{i=1}^{k+1} i = (k+1) + \sum_{i=1}^k i$$

31        Using the induction hypothesis, we can substitute for the
32        second term on the right-hand side:
33        \begin{eqnarray*}
34          \sum_{i=1}^{k+1} i &=& (k+1) + \frac{k(k+1)}{2} \\
35          &=& \frac{2k+2 + k(k+1)}{2} \\
36          &=& \frac{k^2 + 3k + 2}{2} \\
37          &=& \frac{(k+1)(k+2)}{2}
38        \end{eqnarray*}
39      \end{description}
40      The last line shows that when  $n=k+1$ , the equality that we guessed
41      holds. Since we proved the base case, and we've shown that if it
42      holds for  $n=k$  it must hold for  $n=k+1$ , it must hold for all  $n$ .
43    \end{part}
44  \end{question}

```

2: Little Gauss' Formula

(a) We use small examples to attempt to derive a formula for the sum of first n numbers. Listing out the first few examples, we see the following:

n	Sum
0	0
1	0 + 1
2	0 + 1 + 2
3	0 + 1 + 2 + 3
4	0 + 1 + 2 + 3 + 4

Clearly, this increases more quickly than a linear function; so, we guess that it might be quadratic. A function that matches these values and is quadratic is $\frac{n(n+1)}{2}$. That is, we guess that the following equation holds:

$$\sum_{i=0}^n i = \frac{n(n+1)}{2}$$

(b) Now, we prove the equality using induction as follows:

Base Case. $n = 0$: $0 = \frac{0(1)}{2} = 0$.

Induction Hypothesis. Assume the equality holds for some $k \geq 0$.

Induction Step. For $n = k + 1$, we have

$$\sum_{i=1}^{k+1} i = (k+1) + \sum_{i=1}^k i$$

Using the induction hypothesis, we can substitute for the second term on the right-hand side:

$$\begin{aligned} \sum_{i=1}^{k+1} i &= (k+1) + \frac{k(k+1)}{2} \\ &= \frac{2k+2 + k(k+1)}{2} \\ &= \frac{k^2 + 3k + 2}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

The last line shows that when $n = k + 1$, the equality that we guessed holds. Since we proved the base case, and we've shown that if it holds for $n = k$ it must hold for $n = k + 1$, it must hold for all n .