

LaTeX Tutorial

Jeff Clark

Revised February 26, 2002

Contents

1	Introduction	4
1.1	Introduction to <i>LaTeX</i>	4
1.2	Required Components of a <i>LaTeX</i> Document	4
1.3	Using <i>LaTeX</i> on Elon's Computers	5
1.4	Error Messages	6
1.5	Typing <i>LaTeX</i> Commands	6
2	Document Structure	7
2.1	Page Numbering and Headings	7
2.2	Creating a Title Page	7
2.3	Creating a Title Page, Continued	8
2.4	Sections	8
2.5	Cross-References	8
2.6	Table of Contents	8
2.7	Abstracts	9
3	Mathematical Typesetting	10
3.1	Mathematical Formulas	10
3.2	Greek Letters	10
3.3	Exponents and Subscripts	12
3.4	Above and Below	13
3.5	Fractions	13

3.6	Functions	14
3.7	Sums, Integrals, and Limits	14
3.8	Roots	15
3.9	Text in Math Displays	15
3.10	Operators	16
3.11	Relations	17
3.12	Negated Symbols	17
3.13	More Symbols	17
4	Spacing	19
4.1	Spacing Between Words	19
4.2	Fine-Tuning Spacing in Math-Mode	19
4.3	Double Spacing	19
4.4	Sloppy Line Breaks	19
4.5	Enlarging Pages	20
5	Accents and Font Style	21
5.1	Accents	21
5.2	Hyphenation	21
5.3	The L ^A T _E X Logo	21
5.4	Quotation Marks	22
5.5	Changing the Appearance of Words	22
6	Tables, Arrays, and Lists	23
6.1	Constructing Arrays	23
6.2	Constructing Tables	23
7	Multiline Equations	25
7.1	Multi-line Equations	25
7.2	Accents	25
7.3	Bracket Symbols	26
7.4	Dots	27
7.5	Indenting	28

8	Text Formatting	29
8.1	Centering Text	29
8.2	Special Headers	29
8.3	Extended Quotation	29
8.4	Bulleted Lists	29
8.5	Numbered Lists	30
8.6	Filling a Line	31
8.7	Line Breaks	31
9	Bibliography and Compound Expressions	32
9.1	Bibliographies	32
10	Slides	33
10.1	The Slide Class	33
10.2	How to Use the Slides Class	33
11	Including Graphics in Your Document	34
11.1	Graphic File Formats	34
11.2	Graphics Package	34
11.3	Including Graphics Within Your Document	34
12	Business Letters	35
12.1	The Letter Class	35
12.2	Letter Commands for the Preamble	35
12.3	Commands for Each Letter	35

1 Introduction

1.1 Introduction to *LaTeX*

LaTeX is a family of programs designed to produce publication-quality typeset documents. It is particularly strong when working with mathematical symbols.

The history of *LaTeX* begins with a program called *TeX*. In 1978, a computer scientist by the name of Donald Knuth grew frustrated with the mistakes that his publishers made in typesetting his work. He decided to create a typesetting program that everyone could easily use to typeset documents, particularly those that include formulae, and made it freely available. The result is *TeX*.

Knuth's product is an immensely powerful program, but one that does focus very much on small details. A mathematician and computer scientist by the name of Leslie Lamport wrote a variant of *TeX* called *LaTeX* that focuses on document structure rather than such details.

1.2 Required Components of a *LaTeX* Document

Every *LaTeX* document must contain the following three components. Everything else is optional (even text).

1. `\documentclass{article}`
2. `\begin{document}`
3. `\end{document}`

The first statement tells *LaTeX* what kind of document it is to process, as there are different style rules for different kinds of documents. We will use the *article* document class exclusively in this tutorial. Other possible classes include *report*, *book*, and *letter*.

The default font size for each class is 10 point. You can use 11 point or 12 point fonts by including this information in the `\documentclass` command as `\documentclass[11pt]{article}` or `\documentclass[12pt]{article}`. You could also use `\documentclass[10pt]{article}`, but since this is the default you don't need to type the `[10pt]` part.

In general, required information is included in *LaTeX* commands in braces {}, while optional information is included in square brackets [].

The `\documentclass` command must appear at the very beginning of your *LaTeX* document, before any other *LaTeX* commands, or you will get an error message.

If you have commands for *LaTeX* that will affect the whole document, you should include them in the *preamble*, which is what the space between the `\documentclass` and `\begin{document}` commands is called. The body of the document, where you include all of your text, must occur between the `\begin{document}` and `\end{document}` commands. Any text that comes after the `\end{document}` command will be ignored.

1.3 Using *LaTeX* on Elon's Computers

LaTeX consists of several programs:

1. A program `latex` that processes your input.
2. A program (`yap` on our computers) that previews and prints your work.

In addition, you need an editor that produces plain text without formatting commands as well as a good spell-checker.

Jim Beuerle has arranged for all of these programs to be installed on Elon's lab PC's, as well as a program `TeXShell` to coordinate them.

You can start `TeXShell` either from a desktop seashell icon (for the machines in Duke 201, 204, and 209) or from the **Start/Courseware/Math** menu (for the lab machines).

To use `TeXShell`, you need to follow the following steps:

1. You should either create a new *LaTeX* file with **File/New** or open an existing *LaTeX* file with **File/Open**.
2. You need to designate the main file that you will be working with, since *LaTeX* files can include other *LaTeX* files and the programs need to know which one to use; a number of secondary files will be created with the same filename but different extensions. Use **File/Main File**.
3. You should use the main editing screen to enter and revise your work.
4. Always spell-check or your teachers will laugh and/or cry at your work: use **Edit/Spell Check**.
5. Press the **TeX** button to run the *LaTeX* program on your file. A small window will open and close during this time; any error messages will show up there; if successful the window will close itself. If not, you will need to type an "x" and hit *Enter* to close it after reading the error message.
6. If you want to see the warnings and error messages later, press the **Log** button.

7. To view your masterpiece, press the **Preview** button. If you are happy with the result, the **Preview** program has a printer icon at the top left.
8. If your masterpiece includes graphics, you will probably want to create a PDF file for printing and sharing; use the **PDFLaTeX** button to create a PDF file (with a file extension *pdf*). Use *Acrobat Reader* to preview and print it; this program is installed on all Elon PC's and will start up if you double click the file's name in any directory listing.

1.4 Error Messages

LaTeX will tell you when it figures out that something is wrong. Often the actual error occurs earlier in your file.

A common error is not to close braces for a command. Another one that occurs frequently is to use math commands outside of math mode (described later).

Since *LaTeX* will stop after any `\end{document}` command, a good strategy for finding errors is to insert `\end{document}` temporarily earlier in the file to see if the error is above its location.

1.5 Typing *LaTeX* Commands

For this tutorial, you will occasionally find yourself having to type *LaTeX* commands as part of your text. How do you do that without *LaTeX* taking them seriously and following them?

Surround any text that you want printed as is with a `\begin{verbatim}` and an `\end{verbatim}` command.

Practice: Create a document that explains what some *LaTeX* commands do.

2 Document Structure

2.1 Page Numbering and Headings

The command `\pagestyle` controls page numbering and headings. It should always go between the `\documentclass{article}` and the `\begin{document}` commands. It can take the following forms:

1. `\pagestyle{plain}` is the default, which puts the page number at the center of the bottom of the page and provides no headings.
2. `\pagestyle{empty}` provides neither page numbers nor headings.
3. `\pagestyle{headings}` will provide page numbers and headings from any `\section's` that you are using.
4. `\pagestyle{myheadings}` will provide page numbers and custom headings.

These commands can also be applied to a single page using `\thispagestyle` instead of `\pagestyle`.

Practice: Prepare two documents entitled *format1.tex* and *format2.tex* containing any text that you want and using the first two different page styles.

2.2 Creating a Title Page

The title, author, and date of your document are information that various *LaTeX* commands can make use of, if you provide it. It is a good habit to get into to provide this information in the preamble of your document. (Remember that the preamble refers to any commands between the `\documentclass` command and the `\begin{document}` command.) The commands are:

1. `\title{yourtitlehere}`
2. `\author{yournamehere}`
3. `\date{currentdate}`

Given that you have provided this information in the preamble, you may or may not want a title heading. If you do, place a `\maketitle` command immediately after the `\begin{document}` command.

Practice: Create a document of three pages with a title heading, using the `\pagestyle{plain}` command.

2.3 Creating a Title Page, Continued

The `\documentclass` command can take a *titlepage* option:

```
\documentclass[titlepage]{article}.
```

Practice: Create a document of three pages with a title page, using the `\documentclass[titlepage]{article}` command.

2.4 Sections

LaTeX is a language for creating *structured* documents. One of the most important ways of creating structure in a document is to split it into logical sections. If your document deals with more than one concept or theme, then each concept should go into its own section.

There are two related commands for creating sections: `\section{sectiontitle}` and `\section*{sectiontitle}`. The first one numbers the sections, while the starred form does not. Both create separate sections with titles in a larger font size; they also provide information to *LaTeX* in case you want to create a Table of Contents.

Practice: Create a document with five numbered sections.

2.5 Cross-References

If you wish to have cross-references in a document with numbered sections, use `\label{name}` to label the point in your document with some mnemonic, and `\Section \ref{name}` to refer to that point. `\ref{name}` will be replaced by the number of the section containing the corresponding `\label` command. As with your bibliography citations, you will need to run *LaTeX* twice to generate these references.

Practice: Create a fake document with many numbered sections and lots of cross references.

2.6 Table of Contents

For a large document, it is a kindness to your reader to provide a Table of Contents. If you have been using `\section` commands throughout your document, then *LaTeX* has all the information that it needs to construct one for you. Place the command `\tableofcontents` after your `\begin{document}` command.

It may be necessary to run *LaTeX* twice on a document with a Table of Contents: the first time, *LaTeX* stores the page numbers for the sections in a separate file, and then the second time *LaTeX* writes this information into the Table of

Contents. If you have question marks instead of page numbers in your Table of Contents, run *LaTeX* again.

Practice: Create a three page document with at least ten sections and a Table of Contents.

2.7 Abstracts

To create an abstract, place your text in an *abstract* environment, i.e., between `\begin{abstract}` and `\end{abstract}` commands. The abstract should come immediately after your `\maketitle` command, but before any `\tableofcontents` command.

Practice: Create an abstract, with and without a separate titlepage.

3 Mathematical Typesetting

3.1 Mathematical Formulas

There are two ways to insert mathematical formulas into your document with *LaTeX*. One is to have it appear in a paragraph with text. In doing so, the formulas will be compressed vertically: limits for integrals and summations will appear to the side instead of on the top and bottom, etc. The other way is to have them appear in a separate paragraph, where there will be more room.

For formulas that appear in a paragraph, surround them with `$`'s. For example,

`$$\alpha$` is the first letter of the Greek alphabet.

becomes

α is the first letter of the Greek alphabet.

To have formulas appear in their own paragraph, use matching `$$`'s to surround them. For example,

`$$`
`\frac{x^n-1}{x-1} = \sum_{k=0}^{n-1} x^k`
`$$`

becomes

$$\frac{x^n - 1}{x - 1} = \sum_{k=0}^{n-1} x^k$$

Practice: Create your own document with both kinds of formulas.

3.2 Greek Letters

- α is `\alpha`
- β is `\beta`
- γ is `\gamma`
- δ is `\delta`
- ϵ is `\epsilon`
- ε is `\varepsilon`

- ζ is `\zeta`
- η is `\eta`
- θ is `\theta`
- ϑ is `\vartheta`
- ι is `\iota`
- κ is `\kappa`
- λ is `\lambda`
- μ is `\mu`
- ν is `\nu`
- ξ is `\xi`
- o is `o`
- π is `\pi`
- ϖ is `\varpi`
- ρ is `\rho`
- ϱ is `\varrho`
- σ is `\sigma`
- ς is `\varsigma`
- τ is `\tau`
- v is `v`
- ϕ is `\phi`
- φ is `\varphi`
- χ is `\chi`
- ψ is `\psi`
- ω is `\omega`
- Γ is `\Gamma`
- Δ is `\Delta`
- Θ is `\Theta`
- Λ is `\Lambda`

- Ξ is `\Xi`
- Π is `\Pi`
- Σ is `\Sigma`
- Υ is `\Upsilon`
- Φ is `\Phi`
- Ψ is `\Psi`
- Ω is `\Omega`

3.3 Exponents and Subscripts

Use the `^` character (shift-6), known as a caret, to create exponents:

`x^2`

produces

x^2

If you have an exponent containing more than one character, group the exponent characters inside braces.

`$x^{21} \neq x_{21}$`

produces

$x^{21} \neq x_{21}$

Similarly, subscripts are created using the `_` (underscore character). Again, for subscripts of more than one character, use braces to indicate where the subscript starts and stops.

`$x_21 \neq x_{21}$`

produces

$x_21 \neq x_{21}$

Practice: Create a document containing formulas using exponents and subscripts.

3.4 Above and Below

It is useful to be able to draw horizontal lines and braces above and below parts of a formula. We can combine the `\overline`, `\overbrace`, `\underline`, and `\underbrace` commands to our heart's content.

```


$$\frac{(m+n)!}{m!n!} = \frac{\overbrace{(m+n)(m+n-1)\cdots(n+1)}^{m \text{ factors}}}{\underbrace{m(m-1)\cdots 1}_{m \text{ factors}}}$$


```

produces

$$\binom{m+n}{m} = \frac{(m+n)!}{m!n!} = \frac{\overbrace{(m+n)(m+n-1)\cdots(n+1)}^{m \text{ factors}}}{\underbrace{m(m-1)\cdots 1}_{m \text{ factors}}}$$

while

```

$$\overline{x+\overline{y}} = \overline{x}+y$$

```

produces

$$\overline{x+\overline{y}} = \overline{x}+y$$

Practice: Construct an equation using all four of these commands.

3.5 Fractions

Fractions can be written in two ways: with a diagonal fraction bar or a horizontal one. Diagonal fraction bars work best in tight places, such as in a text paragraph or when in a larger fraction.

```

$$a/b$$

```

becomes

a/b

The horizontal bar is clearer when you have more room, such as in a formula paragraph. The command is a little more complicated, because the numerator and denominator are often complicated themselves. A horizontal bar fraction is written as `\frac{numerator}{denominator}`.

```
$$  
\frac{a/b-c/d}{e/f-g/h}  
$$
```

becomes

$$\frac{a/b - c/d}{e/f - g/h}$$

Practice: Construct a couple of nested fractions yourself.

3.6 Functions

LaTeX uses italics in math mode for variables to make them stand out, but Roman (non-italic) for function names. How is *LaTeX* to know the difference between “sin” as function name and “sin” as the product of the variables s , i , and n ? Use a backslash in front of “sin” and other function names to let *LaTeX* know that you want the function, not the product of variables.

Here is a list of function names:

<code>\arccos</code>	<code>\arcsin</code>	<code>\arctan</code>	<code>\arg</code>	<code>\cos</code>	<code>\cosh</code>	<code>\cot</code>
<code>\coth</code>	<code>\csc</code>	<code>\deg</code>	<code>\det</code>	<code>\dim</code>	<code>\exp</code>	<code>\gcd</code>
<code>\hom</code>	<code>\inf</code>	<code>\ker</code>	<code>\lg</code>	<code>\lim</code>	<code>\liminf</code>	<code>\limsup</code>
<code>\ln</code>	<code>\log</code>	<code>\max</code>	<code>\min</code>	<code>\Pr</code>	<code>\sec</code>	<code>\sin</code>
<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>			

3.7 Sums, Integrals, and Limits

Summations and integrals both have lower and upper limits, and the commands are similar. Limits usually have text with an arrow placed below them.

```
$$  
\sum_{k=0}^{\infty}\frac{(-1)^k}{k+1} = \int_0^1\frac{dx}{1+x}  
$$
```

produces

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{k+1} = \int_0^1 \frac{dx}{1+x}$$

\$\$

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

\$\$

produces

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Practice: Construct your own document using sums, integrals, and limits.

3.8 Roots

Use the `\sqrt{}` command to produce square roots:

$$\sqrt{\frac{a}{b}}$$

produces

$$\sqrt{\frac{a}{b}}$$

If you need an n th root, use `\sqrt[n]{}` instead.

$$\sqrt[10]{\frac{a}{b}}$$

produces

$$\sqrt[10]{\frac{a}{b}}$$

Practice: Construct a document containing the quadratic formula and also a cube root.

3.9 Text in Math Displays

There will be times when you want to include Roman, i.e., non-italicized words amongst your mathematical symbols. The font isn't the only problem; spacing is different between letters in a word and variables in a formula. Use the command `\mbox{your text here}` to include short phrases in a formula. (If your phrase isn't short, then you should consider embedding your formula in a text paragraph instead of your text in a formula paragraph.)

```


$$\int_0^{2\pi} \cos(mx) dx = 0 \quad \text{if and only if} \quad m \neq 0$$


```

produces

$$\int_0^{2\pi} \cos(mx) dx = 0 \quad \text{if and only if} \quad m \neq 0$$

Practice: Construct a document containing the following three expressions using `\mbox`:

$$\sqrt{x^2 + y^2} = 0 \quad \text{if and only if} \quad x = y = 0$$

$$\frac{a}{b} > 0 \quad \text{implies that} \quad ab > 0$$

$$\sqrt{x} \quad \text{is only defined if} \quad x \geq 0$$

3.10 Operators

You will probably not need most of the binary operators listed here, but it should be a handy reference:

Operator	Command	Operator	Command
±	<code>\pm</code>	×	<code>\times</code>
∓	<code>\mp</code>	÷	<code>\div</code>
·	<code>\cdot</code>	*	<code>\ast</code>
★	<code>\star</code>	†	<code>\dagger</code>
‡	<code>\ddagger</code>	∏	<code>\amalg</code>
∩	<code>\cap</code>	∪	<code>\cup</code>
⊕	<code>\uplus</code>	∩	<code>\sqcap</code>
⊔	<code>\sqcup</code>	∨	<code>\vee</code>
∧	<code>\wedge</code>	⊕	<code>\oplus</code>
⊖	<code>\ominus</code>	⊗	<code>\otimes</code>
◦	<code>\circ</code>	•	<code>\bullet</code>
◊	<code>\diamond</code>	⊘	<code>\oslash</code>
⊙	<code>\odot</code>	◯	<code>\bigcirc</code>
△	<code>\bigtriangleup</code>	▽	<code>\bigtriangledown</code>
◁	<code>\triangleleft</code>	▷	<code>\triangleright</code>
\	<code>\setminus</code>	ℓ	<code>\wr</code>

3.11 Relations

Again, here are more relations than you will ever need. You may want to print this for reference.

Relation	Command	Relation	Command
\leq	<code>\le</code>	\geq	<code>\ge</code>
\neq	<code>\ne</code>	\sim	<code>\sim</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>
\doteq	<code>\doteq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>
\approx	<code>\approx</code>	\asymp	<code>\asymp</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\cong	<code>\cong</code>	\smile	<code>\smile</code>
\equiv	<code>\equiv</code>	\frown	<code>\frown</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>
\propto	<code>\propto</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>
\models	<code>\models</code>	\perp	<code>\perp</code>
\parallel	<code>\parallel</code>	\mid	<code>\mid</code>

3.12 Negated Symbols

Operator	Command	Operator	Command
$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>
$\not\leq$	<code>\not\le</code>	$\not\geq$	<code>\not\ge</code>
$\not=$	<code>\not=</code>	$\not\equiv$	<code>\not\equiv</code>
$\not\prec$	<code>\not\prec</code>	$\not\succ$	<code>\not\succ</code>
$\not\preceq$	<code>\not\preceq</code>	$\not\succeq$	<code>\not\succeq</code>
$\not\sim$	<code>\not\sim</code>	$\not\simeq$	<code>\not\simeq</code>
$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>
$\not\subseteq$	<code>\not\subseteq</code>	$\not\supseteq$	<code>\not\supseteq</code>
$\not\approx$	<code>\not\approx</code>	$\not\cong$	<code>\not\cong</code>
$\not\sqsubseteq$	<code>\not\sqsubseteq</code>	$\not\sqsupseteq$	<code>\not\sqsupseteq</code>
$\not\asymp$	<code>\not\asymp</code>	\notin	<code>\notin</code>

3.13 More Symbols

Here are some more symbols:

Symbol	Command
\aleph	<code>\aleph</code>
\emptyset	<code>\emptyset</code>
∇	<code>\nabla</code>
∂	<code>\partial</code>
\forall	<code>\forall</code>
\exists	<code>\exists</code>
\neg	<code>\neg</code>
\angle	<code>\angle</code>
\therefore	<code>\therefore</code>
\mathbb{N}	<code>\mathbb{N}</code>
\mathbb{Q}	<code>\mathbb{Q}</code>
\mathbb{R}	<code>\mathbb{R}</code>
\mathbb{Z}	<code>\mathbb{Z}</code>

For `\therefore` you will need to include the line `\usepackage{amssymb}` in your preamble. Similarly, for using `\mathbb{Z}`, etc., you will need to include the line `\usepackage{amssymb}` in your preamble.

4 Spacing

4.1 Spacing Between Words

LaTeX controls the spacing of your document, trying hard to break lines in places that are pleasing to the eye. As a consequence,

- One blank space is the same as a million blank spaces.
- Tabs are treated like blank spaces.
- Blanks at the end of a line are ignored.
- A single “Enter” is treated like a blank space.
- More than one “Enter” marks the beginning of a new paragraph.

4.2 Fine-Tuning Spacing in Math-Mode

It is possible to adjust the spacing that *LaTeX* uses in *math mode*. (I usually add a little space with `\,` before the differential when I write an integral.)

1. `\,` produces a small space
2. `\:` produces a medium space
3. `\;` produces a large space
4. `\!` produces a small negative space

Try them all out in math mode before continuing.

4.3 Double Spacing

There will be times when you will need to submit a draft that is double-spaced, to permit a grader or editor to make comments. *LaTeX* does not explicitly support doing this, because, well, it looks ugly.

Still: to double-space a paper, put `\renewcommand{\baselinestretch}{2}` in your paper’s preamble. Give it a try.

4.4 Sloppy Line Breaks

LaTeX works very hard to find an optimal line break for each line of your document. If you are not happy with its result, surround the offending paragraph with `\begin{sloppypar}` and `\end{sloppypar}` commands. Then *LaTeX* will not break words up but rather will allow more spacing between words in the given paragraph.

4.5 Enlarging Pages

LaTeX works very hard to find the best place to break between pages. If you are unhappy with the result, you can change it with the following two commands:

1. `\newpage` will force the start of a new page.
2. `\enlargethispage{size}` will increase the number of lines added to a page, where *size* is a measurement with units, such as `1in` or `2cm`.

5 Accents and Font Style

5.1 Accents

LaTeX can produce the following accents. (The letter “u” is only used for the purposes of this example. The accents work with any letter.)

<code>\`{u}</code>	<code>\' {u}</code>	<code>\^ {u}</code>	<code>\~ {u}</code>
<code>\" {u}</code>	<code>\= {u}</code>	<code>\. {u}</code>	<code>\u {u}</code>
<code>\v {u}</code>	<code>\H {u}</code>	<code>\t {uu}</code>	<code>\c {u}</code>
<code>\d {u}</code>	<code>\b {u}</code>	<code>\r {u}</code>	

Practice: Try typing some of these accents yourself before proceeding.

5.2 Hyphenation

There are four different variations on hyphens in *LaTeX*: -, --, ---, and \$-\$.

1. - (a single dash) is for hyphenating words.
2. -- (two dashes) is for ranges of numbers.
3. --- (three dashes) is for an honest-to-goodness dash between words.
4. \$-\$ is a minus sign in math mode.

```
My cousin-in-law lived in Germany in 1995--6;  
he speaks French---really, he does. His favorite number is $-2$.
```

produces

My cousin-in-law lived in Germany in 1995–6.

He speaks French—really, he does.

His favorite number is −2.

Notice the difference in appearance of the four variations.

Practice: Try using all of them yourself before continuing.

5.3 The L^AT_EX Logo

You can typeset the L^AT_EX logo with the `\LaTeX` command. As with most commands, it consumes any space behind it, so if it isn’t at the end of a sentence, use `\LaTeX\` instead.

Practice: Try using the logo.

5.4 Quotation Marks

Beginning and ending quotation marks differ. In LaTeX, use ‘ ‘ (usually on the left side of the keyboard) to begin a quotation and ’ ’ (usually on the right side of the keyboard) to end a quotation:

She said ‘ ‘three’ ’.

produces

She said “three”.

Practice: Try using quotation marks before going on.

5.5 Changing the Appearance of Words

There are many ways of changing the appearance of words to add emphasis, such as underlining, boldfacing, and italicizing. When over-used, these changes can make a document hard to read, so they should always be used sparingly.

LaTeX provides underlining, boldfacing, and italicizing, but studies have shown that italicizing is most effecting in stressing without distracting.

Use

`\underline{phrase}`

to underline a phrase,

`\textbf{phrase}`

to print a phrase in boldface, and

`\emph{phrase}`

to italicize a phrase.

Practice: Try all three methods out now.

6 Tables, Arrays, and Lists

6.1 Constructing Arrays

To construct an array, surround the entries with a `\begin{array}{justification}` command and an `\end{array}` command. The *justification* should consist of *l* for left justification, *c* for centered justification, or *r* for right justification. Separate column entries by an `&`, and end each line with a `\\`. If your array is a matrix, you can surround it with large parentheses `\left(` and `\right)`. For example:

```
$$
\left(
\begin{array}{rcl}
\alpha&\beta&\gamma\\
\delta&\epsilon&\zeta\\
\eta&\theta&\iota
\end{array}
\right)
$$
```

produces

$$\begin{pmatrix} \alpha & \beta & \gamma \\ \delta & \epsilon & \zeta \\ \eta & \theta & \iota \end{pmatrix}$$

Practice: Create your own document with an array with a mix of left, center, and right-justified columns.

6.2 Constructing Tables

To construct a table, surround the entries with a `\begin{tabular}{justification}` command and an `\end{tabular}` command. The *justification* should consist of *l* for left justification, *c* for centered justification, or *r* for right justification. Separate column entries by a `&`, and end each line with a `\\`. Use `\hline` to construct a horizontal line, and separate the *l*, *c*, and *r*'s by a `|` wherever you want a vertical line. For example:

```
\begin{tabular}{|r|c|l|}
\hline
Right & Center & Left\\
\hline
```

```
alpha&beta&gamma\\
delta&epsilon&zeta\\
eta&theta&iota\\
\hline
\end{tabular}
```

produces

Right	Center	Left
alpha	beta	gamma
delta	epsilon	zeta
eta	theta	iota

Practice: Create your own document with a table with a mix of left, center, and right-justified columns.

7 Multiline Equations

7.1 Multi-line Equations

Often, in a derivation, you will want to have a series of equations or inequalities aligned together. Surround the equations by `\begin{eqnarray*}` and `\end{eqnarray*}`. (The same command without the asterisk generates equation numbers automatically.) Surround the equals sign or inequality with `&`'s, and end each line with `\\`. Note: you do not need to use `$$`'s with this environment. For example,

```
\begin{eqnarray*}
1+2+\dots+n &=& \frac{1}{2}((1+2+\dots+n)+(n+\dots+2+1))\\
&=& \frac{1}{2}\underbrace{(n+1)+(n+1)+\dots+(n+1)}_{\mbox{$n$ copies}}\\
&=& \frac{n(n+1)}{2}
\end{eqnarray*}
```

produces

$$\begin{aligned} 1 + 2 + \dots + n &= \frac{1}{2}((1 + 2 + \dots + n) + (n + \dots + 2 + 1)) \\ &= \frac{1}{2} \underbrace{(n + 1) + (n + 1) + \dots + (n + 1)}_{n \text{ copies}} \\ &= \frac{n(n + 1)}{2} \end{aligned}$$

Practice: Produce your own aligned set of 5 equations.

7.2 Accents

We use several different kinds of accents in mathematics: a hat, bar, dot, and arrow over a variable all have different meanings. *LaTeX* uses commands that surround the variable:

```
 $\hat{a}$ ,  $\dot{a}$ ,  $\ddot{a}$ ,  $\tilde{a}$ ,  $\bar{a}$ ,  $\vec{a}$ 
```

yields

```
 $\hat{a}$ ,  $\dot{a}$ ,  $\ddot{a}$ ,  $\tilde{a}$ ,  $\bar{a}$ ,  $\vec{a}$ 
```

Practice: Try each of these commands out before continuing.

7.3 Bracket Symbols

Brackets, such as braces and parentheses, are used to group expressions. Without them it would be a good deal more difficult to understand complicated mathematical expressions.

When working with complicated expressions, it is important for the brackets to expand to match the size of whatever they contain. In *LaTeX*, the way to do that is with matching `\left(` and `\right)` commands. (You can use `|`, `{`, `}`, `[`, and `]` instead of parentheses. Remember that since braces are used to group in *LaTeX*, we have to use `\{` and `\}`.) Since every `\left` needs a matching `\right`, it is important to have null commands `\left.` and `\right.` which do nothing. Here are some examples:

```
$$
\left(
\begin{array}{cc}
1&2\\
3&4\\
\end{array}
\right)
$$
```

produces

$$\left(\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right)$$

while

```
$$
|x| =
\left\{
\begin{array}{lr}
-x&x\le 0\\
x&x\ge 0
\end{array}
\right.
$$
```

produces

$$|x| = \left\{ \begin{array}{ll} -x & x \leq 0 \\ x & x \geq 0 \end{array} \right.$$

Practice: Construct a document containing the following expression:

$$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x$$

7.4 Dots

The continuation dots ... are known as an ellipsis. They occur frequently enough in mathematics for *LaTeX* to have four commands to typeset them with the right spacing. They are

1. `\cdots` for center height dots.
2. `\ddots` for diagonal dots, which occur in matrices.
3. `\ldots` for lower height dots.
4. `\vdots` for vertical dots.

```
$a_1, \ldots, a_n$
```

produces

a_1, \dots, a_n

```
$$  
\left(  
\begin{array}{ccc}  
a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{array} \\ \right)  
$$
```

produces

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Practice: Construct a document using at least two different forms of dots.

7.5 Indenting

The default for a *LaTeX* document is to indent new paragraphs unless the paragraph follows a section heading. If you want to change the indentation, use the `\indent` and `\noindent` commands respectively, at the beginning of the paragraph in question.

If you wish to choose the amount of indentation for some reason, then use the command `\setlength{\parindent}{size of indentation with unit}`. (I only do this to set the indentation to 0in when I want no indentation in my documentation.) Since this is a command that affects the whole document, it should go in the preamble, between the `\documentclass` and `\begin{document}` commands.

8 Text Formatting

8.1 Centering Text

By default, *LaTeX* will start all text at the left margin. If you want to center a title, a table, etc., surround what you want centered with the commands `\begin{center}` and `\end{center}`.

Practice: Create a document containing text that is and isn't centered.

8.2 Special Headers

A header is the text automatically included at the top of each document. If you use `\pagestyle{myheadings}`, then you will need some way to indicate what your heading is. The command `\markright{Your Header Text Here}` will do the job for you.

The name of the command `\markright` requires a little explanation. An option that we will not use in this tutorial is `\documentclass[twoside]{article}`, which produces pages formatted as in a book, i.e., with a left page and a right page. Using this option it is possible to produce different headings for the left and right pages. When using the default of one-sided pages, all pages are thought of as right pages, and we use `\markright` to mark our headings on these right-sided pages.

Practice: Produce a three-page document with the name of this course as your heading.

8.3 Extended Quotation

If you are going to include an extended quotation from another source, it is important to indicate the difference between the quotation and your words. The least obtrusive way to do so is to indent. In *LaTeX*, surround the quotation with `\begin{quote}` and `\end{quote}`.

Practice: Create a document containing regular text and containing a quotation of several paragraphs.

8.4 Bulleted Lists

To create a bulleted list, surround the information with a `\begin{itemize}` and an `\end{itemize}`, and begin each item with an `\item`.

For example,

```

\begin{itemize}
\item A bulleted item.
\item Another bulleted item.
\begin{itemize}
\item A nested bulleted item.
\end{itemize}
\item You get the idea.
\end{itemize}

```

produces

- A bulleted item.
- Another bulleted item.
 - A nested bulleted item.
- You get the idea.

Practice: Create a document creating your own bulleted list. Have one of the items in your list itself consist of a bulleted list.

8.5 Numbered Lists

To create a numbered list, surround the information with a `\begin{enumerate}` and an `\end{enumerate}`, and begin each item with an `\item`.

For example,

```

\begin{enumerate}
\item A numbered item.
\item Another numbered item.
\begin{enumerate}
\item A nested numbered item.
\end{enumerate}
\item You get the idea.
\end{enumerate}

```

produces

1. A numbered item.
2. Another numbered item.
 - (a) A nested numbered item.
3. You get the idea.

Practice: Create a document creating your own numbered list. Have one of the items in your list itself consist of a numbered list.

8.6 Filling a Line

You can insert an arbitrary amount of space into a line with the `\hspace{length}` command. Here the length must include a unit, such as 1.5in or 2.3cm.

If you want a spacing in a line that will push the surrounding words to the left and right margins, use the `\hfill` command. If instead of spacing, you want either dots or a line, use `\dotfill` or `\hrulefill`, respectively.

Practice: Try all of these commands out now.

8.7 Line Breaks

LaTeX works very hard to find optimal places to split lines of text in making paragraphs. You can help it by indicating when it should avoid a line break. Use a `~` for a space that should not be used to break a line.

When shouldn't you break a line?

1. Don't break a line between a title such as Mr., Ms., Dr., etc., and the name that follows it.
2. Don't break a line between a number and the units that follow it.
3. Don't break a line between the words in a name.

LaTeX inserts more space at the end of a sentence than between its words. If you use an abbreviation like Dr. in the middle of a sentence, then you need to let *LaTeX* know that the period is not the end of a sentence, with a `~` (if the line should not be broken there) or a `\` followed by a space (if the line could be broken there).

If you need to force a line to break at a given point, use `\\`.

9 Bibliography and Compound Expressions

9.1 Bibliographies

For large documents requiring a good deal of revision, it can be difficult to coordinate references in the body of the document with the bibliography at its end. *LaTeX* provides a mechanism for automatically linking citations with items in the bibliography. Surround the bibliography with `\begin{thebibliography}{9}` and `\end{thebibliography}`. For each entry in the bibliography, start with `\bibitem{label}`, where *label* is some mnemonic for the reference.

With the bibliography in place, a citation in the body of the document is made with `\cite{label}`, where *label* is the same as what occurs in the corresponding `\bibitem{label}`.

In order to keep track of new references that have been added, you will often need to run *LaTeX* twice before previewing when using `\cite`.

What is the 9 in `\begin{thebibliography}{9}` for? It is a dummy number indicating how many digits to leave space for in the numbering of the bibliography. If you have 10-99 references, use `\begin{thebibliography}{99}` instead.

Practice: Create a document with a bibliography of five fake works, and cite each one at least once in your document.

10 Slides

10.1 The Slide Class

LaTeX does not want to be *Microsoft PowerPoint*. On the other hand, if you have mathematical formulae to display on transparencies, *LaTeX* is there to help with the *slides* document class.

The slide class uses a larger font that is designed to be legible at a distance.

10.2 How to Use the Slides Class

The *slides* class is easy to use.

1. Start with `\documentclass{slides}`.
2. Surround the document with `\begin{document}` and `\end{document}` commands.
3. Surround the text that you want to appear on each slide with `\begin{slide}` and `\end{slide}` commands.
4. Preview the slides to see where best to break the material between slides.

11 Including Graphics in Your Document

11.1 Graphic File Formats

There are a number of graphics formats out there, such as:

1. bmp
2. eps
3. gif
4. jpg
5. pdf
6. ps

LaTeX works best with the postscript formats (eps, ps) which were around when the program was first created. For the other formats, *LaTeX* may or may not work; you would be safer trying *PDFLaTeX*, which will produce a PDF document.

11.2 Graphics Package

If you are going to include graphics in your document, you will need to ask *LaTeX* to use a package of graphics commands: place `\usepackage{graphicx}` in the preamble.

11.3 Including Graphics Within Your Document

You use the `\includegraphics{graphicfile}` command to include your graphic file in your document.

If you wish to control the size of the document, you can also specify the height and width: `\includegraphics[height=2in, width = 3in]{graphicfile}`.

12 Business Letters

12.1 The Letter Class

Aside from the *article* class, *LaTeX* provides a *letter* class for formal letters. A given file can be use to generate several letters simultaneously. To use the *letter* class,

1. Start with a `\documentclass{letter}`.
2. Include the commands that apply to all letters in the file.
3. Begin with a `\begin{document}` command.
4. Include the commands for each letter.
5. End with a `\end{document}` command.

12.2 Letter Commands for the Preamble

The following commands apply to each letter in the file:

1. `\address{youraddress}` for your return address.
2. `\signature{yournameandtitle}` for your printed name in the signature block.
3. `\date{letterdate}` if you want to fix the date on the letter; otherwise the date will default to the current date when the letter is printed.

12.3 Commands for Each Letter

1. Start with `\begin{letter}`.
2. On the next line, type the addressee's address in braces. Separate lines with `\\`'s.
3. Put your opening greeting in `\opening{dearjohndoe}`.
4. Put the text of your letter.
5. Put your closing in `\closing{sincerely}`.
6. If their are carbon copies, use `\cc{names}`.
7. If their are enclosures, use `\encl{docs}`.
8. If their is a postscript, use `\ps`.
9. End with `\end{letter}`.

Practice: Type a short letter.