Making a document with LaTeX is a little like making a computer program. You create a file with commands in it that tell the LaTeX processing system how to make your final document. The input file contains source code, and LaTeX converts this in to an output file, typically a PDF file. Here is the source code of a very simple LaTeX document:

```
\documentclass{article}
% Preamble here.
\begin{document}
Hello world!
\end{document}
```

Here's what each line of source code does:

1. `\documentclass{article}`
   This loads a file that contains a number of definitions that set up how the document as a whole is going to look. A common choice is the `article` class. Every LaTeX file must start with a `\documentclass` command. I have provided a custom document class, `homework`, that you can download from the course web page, but its use is not mandatory. Note that the `\` character has a special meaning, and is used to start the name of a command (in this case the `documentclass` command).

2. `% Preamble here.`
   The start of the source code is known as the preamble. You can put commands here that will be used throughout the document. Nothing in the preamble will cause any marks on the page. The preamble for this simple document has two lines, the first line with the mandatory `\documentclass` command, and a second line that has what is known as a comment. Everything on a line following a `%` sign is a comment. Comments are for the benefit of a person reading the source code, and do not result in any marks on the page. You can delete all the comments from a LaTeX file and it will not change how the final document appears, but it might make it harder for a human to figure out what your source code does.

3. `\begin{document}`
   This signals the end of the preamble and the beginning of the document body. Everything following this point can make marks on the page.

4. `Hello world!`
   Now we're writing stuff. In fact, each letter is an instruction to the LaTeX processor. For example, `H` means add the letter H in the current font to the line of text that is currently being built. LaTeX subsequently uses algorithms to determine where lines should be broken to build paragraphs, and how these lines should be split up to make pages. You can tell LaTeX to start a new paragraph by adding a blank line to your file.

5. `\end{document}`
   This tells LaTeX that we are done describing the document and it can finish up writing the final output. Any text occurring after this point in the source code will be ignored.

LaTeX and mathematics

The reason LaTeX is used nearly universally by mathematicians is that it makes it relatively easy to typeset good looking mathematics. For example:

The side lengths of a right triangle satisfy $a^2 + b^2 = c^2$.

Here's the source code:

```
The side lengths of a right triangle satisfy $a^2+b^2=c^2$.
```

Math mode is started with a $ character and ended with a $ as well. Everything in between is treated as material to by typeset using the rules for math rather than the rules for regular text. Note that to get a superscript, we use the ^ character. Here's a related example:

Recall the exponent rule $x^{a+b} = x^a x^b$.

```
Recall the exponent rule $x^{a+b}=x^a x^b$.
```

To get an exponent with more than one character in it, we need to enclose the contents in a pair of curly braces. Curly braces in LaTeX are used to group material to be treated as a single character. But then, how do you put curly braces on the page? You use the commands \{ and \}. For example:

Let $A = \{x : f(x) = 0\}$.

```
Let $A=\{x: f(x)=0\}$.
```

All the math we've described so far is being typeset is in-line mode. This means that the math is written as part of the sentence without any extra space around it. The other common way to write mathematics is called display mode, where the text beforehand ends a line, and the math is given some extra space and centered in the middle of the page. For example:

Fermat's Theorem states that if n is an integer with $n \geq 3$, then there do not exist any non-zero integers a, b, and c such that

$$a^n + b^n = c^n.$$

```
Fermat's Theorem states that if $n$ is an integer
with $n\ge 3$, then there do not exist any non-zero integers
$a$, $b$, and $c$ such that
\[
a^n+b^n=c^n.
\]
```

The \[and \] commands are used to enclose a single line of display mathematics. Notice that the mathematics is still part of the sentence, and ends with a period. Good mathematical style dictates that everything, even the displayed math, needs to be part of a sentence. The displayed math can even be part of the middle of a sentence; you just keep going with the sentence after the \] command.

Frequently in displayed math, we need to break the math up into lines, perhaps with some alignment. For this, we use the \begin{align*}/\end{align*} commands.

Notice that

$$\begin{aligned}\log(4c^3) &= \log(4) + \log(c^3)\\ &= \log(2^2) + \log(c^3)\\ &= 2\log(2) + 3\log(c).\end{aligned}$$

```
Notice that
\begin{align*}
\log(4c^3) &= \log(4)+\log(c^3) \\
&= \log(2^2) + \log(c^3) \\
&= 2\log(2) + 3\log(c).
\end{align*}
```

Each line of math is terminated by a \\ command. The & character is used to create alignment. The lines are typeset so that the positions of the &'s on each line of math all fall on the same vertical line. For alignment with binary relations, like = or < or ≤, it is traditional to put the alignment character just before the relation. Even for displayed math split over several lines, the material all forms part of a sentence that ends with a period. It could be read out loud as "Notice that $\log(4c^3)$ is equal to $\log(4) + \log(c^3)$, which is equal to $\log(2^2) + \log(c^3)$, which is equal to $2\log(2) + 3\log(c)$". It's a bit of a run-on sentence, but still legitimate. The displayed math split into lines results in an equivalent sentence that is a lot easier to read.

The previous example also introduces the command \log. Many standard functions in mathematics are written in Roman font rather than in italics, and have special spacing rules. The trig functions sine and cosine are examples, as are the logarithms. Each of these functions can be typeset using an associated command: \sin, \cos, \log, etc.

Sometimes you want to number an equation so that you can refer to it later.

A common error is writing

$$\log(a + b) = \log(a) + \log(b). \tag{1}$$

We can see that equation (1) cannot be correct by taking $a = b = 1$. Then

$$\log(a + b) = \log(2) \approx 0.7$$

but

$$\log(a) + \log(b) = \log(1) + \log(1) = 0 + 0 = 0.$$

```
A common error is writing
\begin{equation}\label{logmistake}
  \log(a+b) = \log(a)+\log(b).
\end{equation}
We can see that equation \eqref{logmistake} cannot be correct
by taking $a=b=1$.  Then
\[
\log(a+b)=\log(2)\approx 0.7
\]
but
\[
\log(a)+\log(b)=\log(1)+\log(1)=0+0=0.
\]
```

To get a displayed equation with a number, use `\begin{equation}` and `\end{equation}` rather than `\[` and `\]`. A mnemonic label `logmistake` is given in this example using the `\label` command so that your can refer to the equation number in your text without happening to know what its exact number is. You refer to the equation number using the `\eqref` command, which typesets the equation number surrounded by parentheses. If the equation number happens to change in a later revision, the text generated by `\eqref` will be kept up to date. However, LaTeX frequently needs to process a document twice to get all the labels up to date. There are old historical reasons for this, and it's a bit of a hassle in the modern era. Just keep in mind that if your labels look wrong, process the document again and the problems will likely go away. In some cases a third processing might be in order.

You might have wondered what the * in `align*` meant. It indicates that the lines in the display should not have numbers. If you use `\begin{align}`/`\end{align}` instead, each line in the display will be given a number. You can label a line using `\label` just as you would a normal equation. And you can indicate that a specific line should have no number using the `\nonumber` command. This business with * is a common convention in LaTeX. In fact, the commands `\[` and `\]` we saw earlier to generate a single line of un-numbered display math are shorthand for `\begin{equation*}` and `\end{equation*}`.

With a sequence of aligned equations, you sometimes want to put some justification for each step to the side of that step. For example:

Note that

$$\begin{aligned}(m+n)p &= p(m+n) && \text{by commutivity}\\ &= pm+pn && \text{by distributivity}\\ &= mp+np && \text{by commutivity.}\end{aligned}$$

```
Note that
\begin{align*}
(m+n)p &=  p(m+n)     &&\text{by commutivity}\\
&= pm+pn         &&\text{by distributivity}\\
&= mp+np         &&\text{by commutivity}.
\end{align*}
```

The command `\text` is used to insert some material typeset according to text rules into a math environment. The extra &'s are used to create an additional alignment along the right side of the display.

Typesetting homework

A typical homework will consist of a set of problems. For this class, the `HWDemo.tex` file on the course web page can be used as a template for your solutions. This file contains some commands that make the process of writing solutions easier. Here's an example:

1. Suppose p is even and q is any integer. Then pq is even.

Solution:
Since p is even, there exists an integer k such that $p = 2k$. Hence

$$\begin{aligned} pq &= (2k)q \\ &= 2(kq). \end{aligned}$$

Since $kq \in \mathbb{Z}$, we conclude that $2 \mid pq$ and hence pq is even.

```
\begin{problems}
\problem
Suppose $p$ is even and $q$ is any integer. Then $pq$ is even.
\solution
Since $p$ is even, there exists an integer $k$ such that
$p=2k$.  Hence
\begin{align*}
  pq &= (2k)q\\
     &= 2(kq).
\end{align*}
Since $kq\in\mathbb{Z}$, we conclude that
$2 \mid pq$ and hence $pq$ is even.
\end{problems}
```

The set of all problems goes between a `\begin{problems}`/`\end{problems}` pair. The individual problem statements goes after a `\problem` command and the solution follows a `\solution` command.

Making your own commands

In mathematics, the real numbers are typically denoted by the letter R written in a distinctive font. Sometimes it is written as a bold letter, and on the blackboard we add some decoration to the letter to make it seem bold. This has become so customary that we now also have fonts that simulate the blackboard bold. In LaTeX we obtain this using the `\mathbb` command. For example, `$\mathbb{R}$` results in $\mathbb{R}$. Since the real numbers are frequently used, it would be nice to have a command for this. LaTeX does not provide one by default, but it is easy enough to add one. In the preamble, place

```
\newcommand\Reals{\mathbb{R}}
```

Then, in your text you can obtain $\mathbb{R}$ simply by typing `$\Reals$`. I've provided a list of some helpful commands in `math215extras.tex`, which you can include in your files by using `\input{math215extras.tex}`. You can add your own definitions to this file, or you can put one-time use definitions in a document's preamble.

Sometimes you want a command to depend on some input. For example, consider the absolute value operator, $|\cdot|$. To obtain a simple absolute value, you could just use `$|x|$`, which results in $|x|$. This looks fine, but the output of `$|\int x^2|$` is not so nice:

$$|\int x^2|.$$

We really want the absolute value signs to be large enough to enclose the exponent and the integral sign. To do this, LaTeX provides `\left`/`\right` commands that work with delimiters such as `|`, `(`, `[` and so forth to make the delimiters as large as needed. The appropriate input is `\left|\int x^2\right|`, which results in

$$\left|\int x^2\right|.$$

But you might not want to type all that every time you need an absolute value sign. A nice solution is the following:

```
\newcommand{\abs}[1]{\left|#1\right|}
```

This makes a new command, `\abs`, that takes one input. The input replaces the `#1` in the command definition, so `$\abs{\int x^2}$` is the same thing as `$\left|\int x^2\right|$`.

Some basic tasks

Task	Command	Example	Result
Fractions	`\frac`	`$\frac{1}{1-x}$`	$\frac{1}{1-x}$

Infinity	`\infty`	`$\infty}$`	∞
Integrals	`\int`	`$\int_a^b$`	$\int_a^b$
Limits	`\lim`	`$\lim_{x\rightarrow a} f(x)$`	$\lim_{x\to a} f(x)$
Right arrow	`\rightarrow`	`$x\rightarrow a$`	$x \to a$
Sums	`\sum`	`$\sum_{k=1}^N$`	$\sum_{k=1}^{N}$
Greek	`\alpha`, `\beta`, etc.	`$\gamma(t)$`	$\gamma(t)$
Uppercase Greek	`\Theta`, `\Xi`, etc.	`$\Phi(n)$`	$\Phi(n)$
Greater than or equal	`\ge`	`$\pi \ge e$`	$\pi \geq e$
Less than or equal	`\le`	`$e \le \pi$`	$e \leq \pi$
Element of	`\in`	`$5\in\mathbb{Z}$`	$5 \in \mathbb{Z}$
Not an element of	`\not\in`	`$\pi\not\in\mathbb{Z}$`	$\pi \notin \mathbb{Z}$
Divides relation	`\mid`	`$3 \mid 15$`	$3 \mid 15$
Does not divide	`\mid`	`$4 \not\mid 15$`	$4 \nmid 15$
Binomial coefficients	`\binom`	`$\binom{5}{3}$`	$\binom{5}{3}$
A thin math space	`$\;$`	`$\int_0^\pi \sin(x)\;dx$`	$\int_0^\pi \sin(x)\,dx$
Bold text	`\textbf`	`Hello \textbf{world}!`	Hello **world**!
Italic text	`\textit`	`Hello \textit{world}!`	Hello *world*!
Larger font	`\large`	`{\large Hello world!}`	Hello world!
Smaller font	`\small`	`{\small Hello world!}`	Hello world!

Resources

There is an online text at `http://en.wikibooks.org/wiki/LaTeX`.

You can find a large list of mathematical symbols at `http://amath.colorado.edu/documentation/LaTeX/Symbols.pdf`.