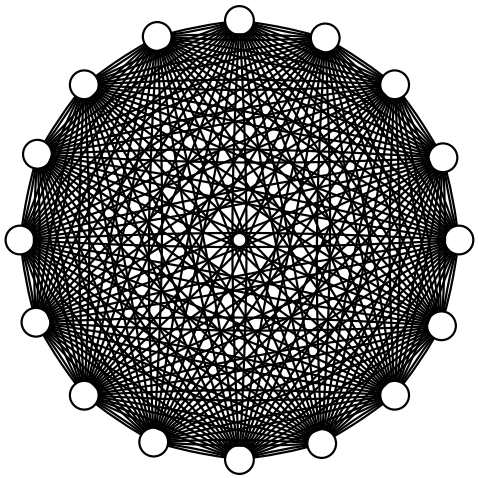
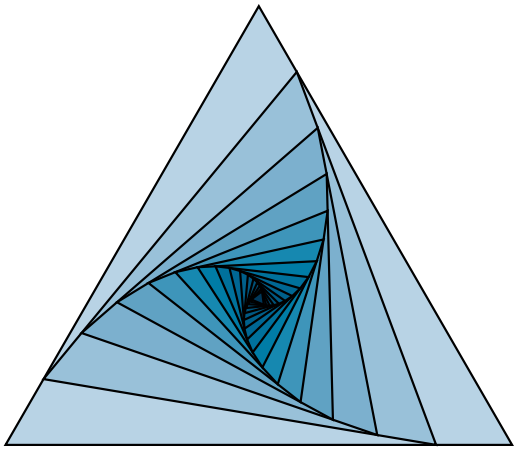
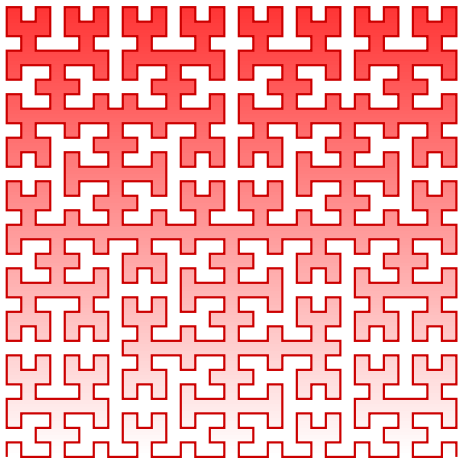
Introduction to LaTeX, *part I*

Graduate Mathematics Association
Department of Mathematics
University of Florida
April 19, 2016
Presentation Written by Jay Pantone

What is LaTeX?

What is LaTeX good for?

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.
- Designing posters.

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.
- Designing posters.

What is LaTeX bad for?

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.
- Designing posters.

What is LaTeX bad for?

- Doing mathematical calculations.

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.
- Designing posters.

What is LaTeX bad for?

- Doing mathematical calculations.
- Running algorithms.

What is LaTeX?

What is LaTeX good for?

- Writing documents, especially those with scientific notation.
- Making presentation slides.
- Designing posters.

What is LaTeX bad for?

- Doing mathematical calculations.
- Running algorithms.
- Analyzing data.

What is LaTeX?

The most important fact about LaTeX:

What is LaTeX?

The most important fact about LaTeX:

> *You can't learn how to use it by watching someone else use it.*

What is LaTeX?

The most important fact about LaTeX:

> *You can't learn how to use it by watching someone else use it.*

The second most important fact about LaTeX:

What is LaTeX?

The most important fact about LaTeX:

> *You can't learn how to use it by watching someone else use it.*

The second most important fact about LaTeX:

> *Google knows everything about it.*

What is LaTeX?

The most important fact about LaTeX:

> *You can't learn how to use it by watching someone else use it.*

The second most important fact about LaTeX:

> *Google knows everything about it.*

So why am I here?

What is LaTeX?

The most important fact about LaTeX:

> *You can't learn how to use it by watching someone else use it.*

The second most important fact about LaTeX:

> *Google knows everything about it.*

So why am I here?

> *To show you what LaTeX can do.*

BACKGROUND AND HISTORY

Background and History

- TEX was written by Donald Knuth in the 70s and 80s so that he could make his books look good.

Background and History

- TEX was written by Donald Knuth in the 70s and 80s so that he could make his books look good.
- LATEX was created by Leslie Lamport to make TEX easier to use.

Background and History

- TEX was written by Donald Knuth in the 70s and 80s so that he could make his books look good.
- LATEX was created by Leslie Lamport to make TEX easier to use.
- There are now many variants of TEX, all with funny names.

Background and History

Background and History

- Overleaf was founded in 2012 as a means to write documents online

Background and History

- Overleaf was founded in 2012 as a means to write documents online
- Pros: Multiple people can edit a document simultaneously, accessible anywhere, comes with built in packages, don't need to install!

Background and History

- Overleaf was founded in 2012 as a means to write documents online
- Pros: Multiple people can edit a document simultaneously, accessible anywhere, comes with built in packages, don't need to install!
- Cons: Requires internet access

Background and History

Background and History

- How do you get around this?

Background and History

- How do you get around this?
- You can manually install LaTeX on your computer, which requires the installation of a distributor and an interface

Background and History

- How do you get around this?
- You can manually install LaTeX on your computer, which requires the installation of a distributor and an interface
- Common distributors: MiKTex, MacTeX, TeX Live.
- Common interfaces: Texmaker

Basic Document

A basic LaTeX document has three parts:

Basic Document

A basic LaTeX document has three parts:

- a document class,

Basic Document

A basic LaTeX document has three parts:

- a document class,
- a preamble,

Basic Document

A basic LaTeX document has three parts:

- a document class,
- a preamble,
- a body.

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

- article

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

- article
- report

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

- article
- report
- book

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

- article
- report
- book
- memoir

Document Classes

The first line of a LaTeX file must be

```
\documentclass{[class]}
```

where "`[class]`" is replaced by the type of document you are creating.

Some popular classes are:

- article
- report
- book
- memoir
- beamer

Preamble

The preamble is where you define the style of your document and load any packages you need to use.

Preamble

The preamble is where you define the style of your document and load any packages you need to use.

Set margins:

```
\usepackage[top=2in, bottom=1.5in,
    left=1in, right=1in]{geometry}
```

PREAMBLE

The preamble is where you define the style of your document and load any packages you need to use.

Set margins:

```
\usepackage[top=2in, bottom=1.5in,
    left=1in, right=1in]{geometry}
```

Load packages:

```
\usepackage{graphicx}
```

Body

The body contains all of your content.

Body

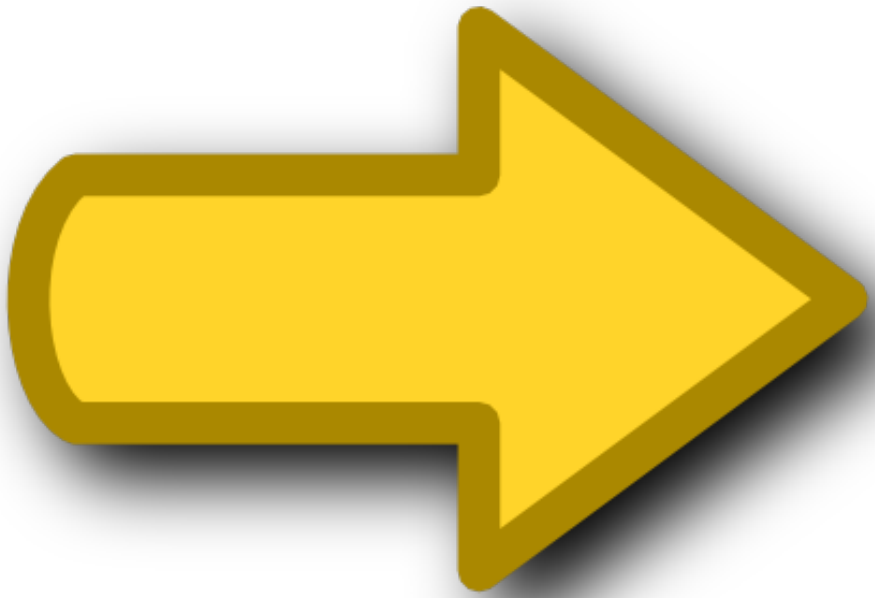
The body contains all of your content.

The body must start with

```
\begin{document}
```

and end with

```
\end{document}.
```



Text Effects

Text can be *italicized* (using `\emph`), **bolded** (using `\textbf`), and <u>underlined</u> (using `\underline`).

Text Effects

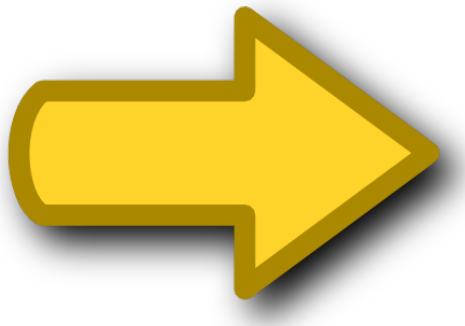
Text can be *italicized* (using `\emph`), **bolded** (using `\textbf`), and underlined (using `\underline`).

You can also use the effects textsf, `texttt`, TEXTSC, or UPPERCASE.

Text Effects

Text can be *italicized* (using `\emph`), **bolded** (using `\textbf`), and underlined (using `\underline`).

You can also use the effects textsf, `texttt`, TEXTSC, or UPPERCASE.



Test #1: Write some text that is bold *and* underlined.

Comments

When you want to add comments that won't appear in the pdf, start the line with a %.

Comments

When you want to add comments that won't appear in the pdf, start the line with a %.

This helps keep your code organized. You can break separate sections by putting a commented line between them.

```
% ========================================
```

Comments

When you want to add comments that won't appear in the pdf, start the line with a %.

This helps keep your code organized. You can break separate sections by putting a commented line between them.

```
% ==========================================
```

In the code that created this slideshow, each slide is separated by:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Spacing

LaTeX treats any number of spaces as a single space.

Spacing

LaTeX treats any number of spaces as a single space.

```
See    Spot   run.
```
$\Longrightarrow$
See Spot run.

Spacing

LaTeX treats any number of spaces as a single space.

```
See    Spot   run.
```
$\Longrightarrow$
See Spot run.

Single new lines are treated as if there is no new line.

Spacing

LaTeX treats any number of spaces as a single space.

```
See    Spot   run.
```
$\Longrightarrow$
See Spot run.

Single new lines are treated as if there is no new line.

```
See Spot run.
Spot runs fast.
```
$\Longrightarrow$
See Spot run. Spot runs fast.

Spacing

LaTeX treats any number of spaces as a single space.

```
See    Spot   run.
```
$\Longrightarrow$
See Spot run.

Single new lines are treated as if there is no new line.

```
See Spot run.
Spot runs fast.
```
$\Longrightarrow$
See Spot run. Spot runs fast.

Multiple blank lines are treated as a single new line.

Spacing

LaTeX treats any number of spaces as a single space.

```
See    Spot   run.
```
$\Longrightarrow$
See Spot run.

Single new lines are treated as if there is no new line.

```
See Spot run.
Spot runs fast.
```
$\Longrightarrow$
See Spot run. Spot runs fast.

Multiple blank lines are treated as a single new line.

```
See Spot run.

Spot runs fast.
```
$\Longrightarrow$
See Spot run.
Spot runs fast.

Spacing

To create a space, use \

Spacing

To create a space, use \

You can use \\ or `\newline` to create a new line.

Spacing

You can force horizontal and vertical space using the `\hspace` and `\vspace` commands.

Spacing

You can force horizontal and vertical space using the `\hspace` and `\vspace` commands.

You have to give each command a length: `\hspace{0.1cm}` or `\hspace{10pt}`.

Spacing

You can force horizontal and vertical space using the `\hspace` and `\vspace` commands.

You have to give each command a length: `\hspace{0.1cm}` or `\hspace{10pt}`.

Sometimes LaTeX thinks it would look better without the space, and so it chooses to ignore you. In this case, add an asterisk:

```
\hspace*{0.1cm}
```

Spacing

You can force horizontal and vertical space using the `\hspace` and `\vspace` commands.

You have to give each command a length: `\hspace{0.1cm}` or `\hspace{10pt}`.

Sometimes LaTeX thinks it would look better without the space, and so it chooses to ignore you. In this case, add an asterisk:

```
\hspace*{0.1cm}
```

You can also use negative space to bring things closer together: `\hspace{-0.1cm}`

Spacing

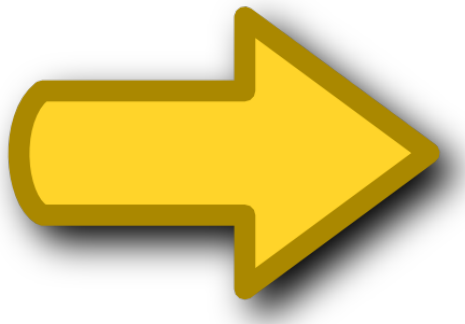
You can force horizontal and vertical space using the `\hspace` and `\vspace` commands.

You have to give each command a length: `\hspace{0.1cm}` or `\hspace{10pt}`.

Sometimes LaTeX thinks it would look better without the space, and so it chooses to ignore you. In this case, add an asterisk:

```
\hspace*{0.1cm}
```

You can also use negative space to bring things closer together: `\hspace{-0.1cm}`



Spacing

To create a newpage, you can use `\newpage`.

Spacing

To create a newpage, you can use `\newpage`.

Usually, this will cause an indentation to appear. If you do not want an indent, use `\noindent`.

Quotes

To type an opening quote (‘), use a backtick (`).

Quotes

To type an opening quote (‘), use a backtick (`).

To type a closing quote (’), use the normal single quote (´).

Quotes

To type an opening quote (‘), use a backtick (`).

To type a closing quote (’), use the normal single quote (´).

If you try to use the normal quote symbol, you get this:

"This is an ugly quote."

Quotes

To type an opening quote (‘), use a backtick (`).

To type a closing quote (’), use the normal single quote (´).

If you try to use the normal quote symbol, you get this:

"This is an ugly quote."

Using `` and ´´, you get this:

“This is a pretty quote!”

Dashes

There are three sizes of dashes you can use.

Dashes

There are three sizes of dashes you can use.

- - (–, hyphen): used for compound words and names

Dashes

There are three sizes of dashes you can use.

- - (−, hyphen): used for compound words and names
- – (−−, en-dash): used for ranges or multiple names

Dashes

There are three sizes of dashes you can use.

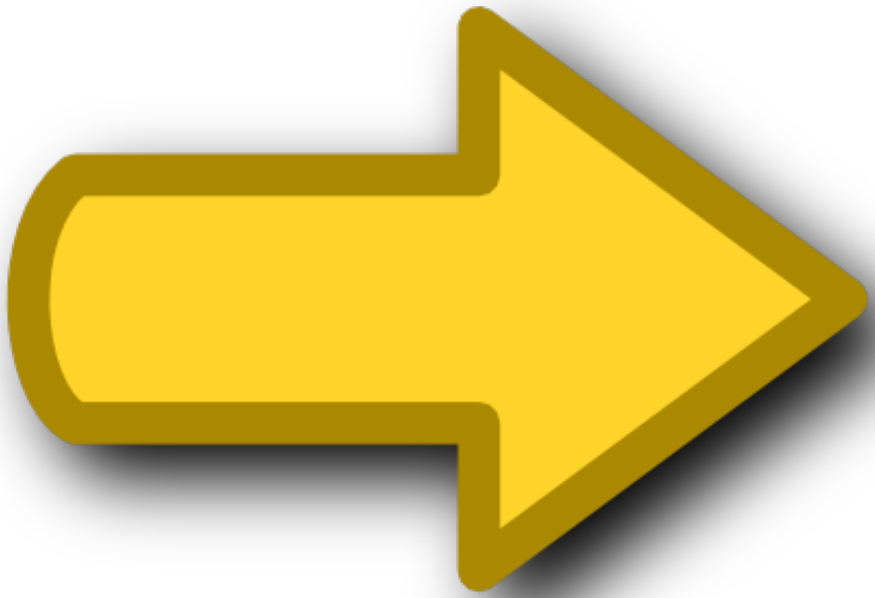
- - (-, hyphen): used for compound words and names
- – (--, en-dash): used for ranges or multiple names
- — (---, em-dash): used for punctuation in sentences

Dashes

There are three sizes of dashes you can use.

- - (-, hyphen): used for compound words and names
- – (--, en-dash): used for ranges or multiple names
- — (---, em-dash): used for punctuation in sentences

In the years 1965–2014, some progress has been made in proving the Birch–Swinnerton-Dyer Conjecture — a quite difficult problem in the field of Number Theory.



Test #2: Write your favorite quote below.

Math Mode

The reason that LaTeX was invented was to format scientific symbols nicely.

Math Mode

The reason that LaTeX was invented was to format scientific symbols nicely.

To type anything in *math mode*, put it between dollar signs:

`$...$`

Math Mode

The reason that LaTeX was invented was to format scientific symbols nicely.

To type anything in *math mode*, put it between dollar signs:

```
$...$
```

In math mode, basic math symbols look better.

```
a+b-c=d
$a+b-c=d$
```

$\Longrightarrow$

a+b-c=d

$a+b-c=d$

Commands and Arguments

Some commands don't need any arguments, like the greek letters.

Commands and Arguments

Some commands don't need any arguments, like the greek letters.

```
\alpha(x) = \beta
```

$$\Longrightarrow \alpha(x) = \beta$$

Commands and Arguments

Some commands don't need any arguments, like the greek letters.

```
\alpha(x) = \beta
```
$\Longrightarrow$ $\alpha(x) = \beta$

Some commands need arguments. The arguments come after the command, surrounded by $\{\cdots\}$.

Commands and Arguments

Some commands don't need any arguments, like the greek letters.

```
\alpha(x) = \beta
```
$\Longrightarrow$ $\alpha(x) = \beta$

Some commands need arguments. The arguments come after the command, surrounded by $\{\cdots\}$.

```
\sqrt{x}
```
$\Longrightarrow$ $\sqrt{x}$

Commands and Arguments

Some commands don't need any arguments, like the greek letters.

```
\alpha(x) = \beta
```
$\Longrightarrow$ $\alpha(x) = \beta$

Some commands need arguments. The arguments come after the command, surrounded by $\{\cdots\}$.

```
\sqrt{x}
```
$\Longrightarrow$ $\sqrt{x}$

```
\frac{a}{b}
```
$\Longrightarrow$ $\frac{a}{b}$

Commands and Arguments

LaTeX lets you combine commands and it still formats them nicely.

Commands and Arguments

LaTeX lets you combine commands and it still formats them nicely.

```
\sqrt{\alpha(x)}=\frac{\sqrt{a}}{b}
```

$$\Longrightarrow \sqrt{\alpha(x)} = \frac{\sqrt{a}}{b}$$

Commands and Arguments

LaTeX lets you combine commands and it still formats them nicely.

```
\sqrt{\alpha(x)}=\frac{\sqrt{a}}{b}
```

$$\Longrightarrow \sqrt{\alpha(x)} = \frac{\sqrt{a}}{b}$$

Sometimes commands have optional arguments, which go between the command and the other arguments and are surrounded by $[\cdots]$.

Commands and Arguments

LaTeX lets you combine commands and it still formats them nicely.

```
\sqrt{\alpha(x)}=\frac{\sqrt{a}}{b}
```

$$\Longrightarrow \sqrt{\alpha(x)} = \frac{\sqrt{a}}{b}$$

Sometimes commands have optional arguments, which go between the command and the other arguments and are surrounded by $[\cdots]$.

```
\sqrt[3]{x}
```

$$\Longrightarrow \sqrt[3]{x}$$

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

```
a_n = n^2 + 1
```

$$\Longrightarrow \quad a_n = n^2 + 1$$

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

```
a_n = n^2 + 1
```
$\Longrightarrow$ $a_n = n^2 + 1$

When the subscript or superscript is more than one character, you must wrap it in $\{\cdots\}$ to group it together.

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

```
a_n = n^2 + 1
```
$$\Longrightarrow a_n = n^2 + 1$$

When the subscript or superscript is more than one character, you must wrap it in $\{\cdots\}$ to group it together.

```
f(n,k) = n^{2k+1}
```
$$\Longrightarrow f(n,k) = n^{2k+1}$$

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

```
a_n = n^2 + 1
```
$\Longrightarrow$ $a_n = n^2 + 1$

When the subscript or superscript is more than one character, you must wrap it in $\{\cdots\}$ to group it together.

```
f(n,k) = n^{2k+1}
```
$\Longrightarrow$ $f(n,k) = n^{2k+1}$

Nesting is allowed.

Subscripts and Superscripts

Subscripts and superscripts in math mode are formed using the _ and ^ characters.

```
a_n = n^2 + 1
```
$\Longrightarrow$ $a_n = n^2 + 1$

When the subscript or superscript is more than one character, you must wrap it in $\{\cdots\}$ to group it together.

```
f(n,k) = n^{2k+1}
```
$\Longrightarrow$ $f(n,k) = n^{2k+1}$

Nesting is allowed.

```
n^{n^{n^n}} - k^{k^{k^k}}
```
$\Longrightarrow$ $n^{n^{n^n}} - k^{k^{k^k}}$

Display Style

There are two styles of math mode. We've already seen the inline mode, where math is enclosed in `$...$`.

Display Style

There are two styles of math mode. We've already seen the inline mode, where math is enclosed in `$...$`.

In display mode, math is put on it's own line and centered. The old way to write math in display mode is with `$$...$$`. The newer way is to use `\[...\]`.

Display Style

There are two styles of math mode. We've already seen the inline mode, where math is enclosed in `$...$`.

In display mode, math is put on it's own line and centered. The old way to write math in display mode is with `$$...$$`. The newer way is to use `\[...\]`.

```
$\int_0^1 x^{-x}dx = \sum_{n=1}^\infty n^{-n}$
```

$$\Longrightarrow \int_0^1 x^{-x}dx = \sum_{n=1}^{\infty} n^{-n}$$

Display Style

There are two styles of math mode. We've already seen the inline mode, where math is enclosed in `$...$`.

In display mode, math is put on it's own line and centered. The old way to write math in display mode is with `$$...$$`. The newer way is to use `\[...\]`.

```
$\int_0^1 x^{-x}dx = \sum_{n=1}^\infty n^{-n}$
```

$$\int_0^1 x^{-x}dx = \sum_{n=1}^{\infty} n^{-n}$$

```
\[\int_0^1 x^{-x}dx = \sum_{n=1}^\infty n^{-n}\]
```

$$\int_0^1 x^{-x}dx = \sum_{n=1}^{\infty} n^{-n}$$

Other Math Commands

Some commands have weird formats.

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Some symbols have weird names.

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Some symbols have weird names.

```
\otimes \doublecap \coprod
```

$$\Longrightarrow \otimes \Cap \coprod$$

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Some symbols have weird names.

```
\otimes \doublecap \coprod
```

$\Longrightarrow \otimes \Cap \coprod$

So how can you learn them all?

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Some symbols have weird names.

```
\otimes \doublecap \coprod
```

$\Longrightarrow \otimes \Cap \coprod$

So how can you learn them all?

- Google

Other Math Commands

Some commands have weird formats.

```
{n \choose k} = \frac{n!}{k!(n-k)!}
```

$$\Longrightarrow \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Some symbols have weird names.

```
\otimes \doublecap \coprod
```

$\Longrightarrow \otimes \Cap \coprod$

So how can you learn them all?

- Google
- http://detexify.kirelabs.org/

Reserved Characters

Some characters have special meanings. These are called *reserved characters*.

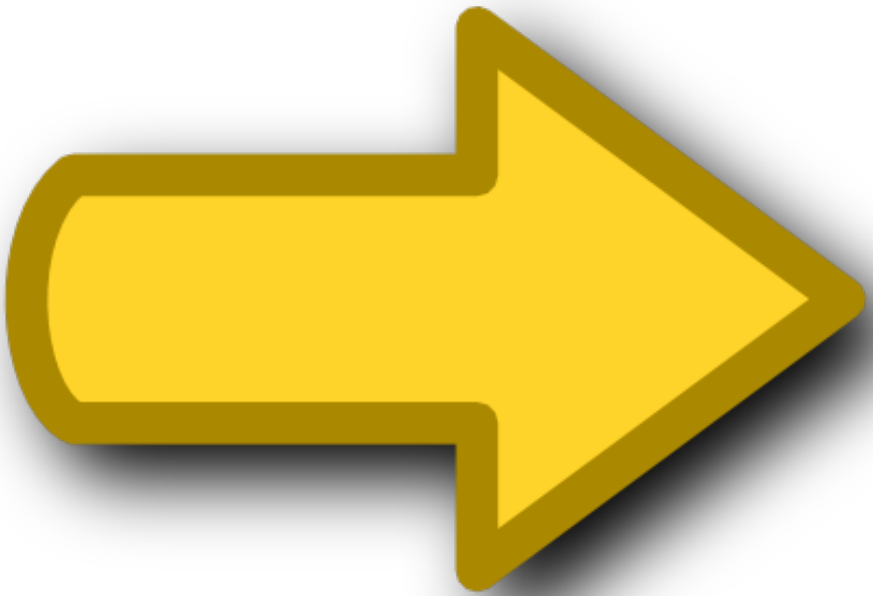
Reserved Characters

Some characters have special meanings. These are called *reserved characters*.

The reserved characters are:

```
# $ % ^ & _ { } ~ \
```

If you want to use them in your document, you have to *escape* them, normally by adding a backslash before them. (Sometimes, it's harder to escape them, just Google it!)



Test #3: Write the following equations in math mode:

$$y = 3x + \frac{2}{3} \tag{1}$$

$$y = \cos^{-1}(x) - \sin^{-1}(x) \tag{2}$$

$$x' = \frac{\beta x \theta^n}{\theta^n + x^n} - \gamma x \tag{3}$$

Tables

LaTeX has many *environments* for displaying data.

Tables

LaTeX has many *environments* for displaying data.

In text mode, the `tabular` environment is used to make tables.

Tables

LaTeX has many *environments* for displaying data.

In text mode, the `tabular` environment is used to make tables.

```
\begin{tabular}{||c|c|c||}
   \hline\hline
   fruit & quantity & price \\
   \hline
   apple & 6 & \$4.00\\
   orange & 12 & \$3.00\\
   banana & 5 & \$3.50\\
   \hline\hline
\end{tabular}
```

Tables

LaTeX has many *environments* for displaying data.

In text mode, the `tabular` environment is used to make tables.

```
\begin{tabular}{||c|c|c||}
   \hline\hline
   fruit & quantity & price \\
   \hline
   apple & 6 & \$4.00\\
   orange & 12 & \$3.00\\
   banana & 5 & \$3.50\\
   \hline\hline
\end{tabular}
```

fruit	quantity	price
apple	6	$4.00
orange	12	$3.00
banana	5	$3.50

Tables

With a little work (and using some packages), we can make fancy tables.

Tables

With a little work (and using some packages), we can make fancy tables.

name	*foo*			
Models	A	B	C	D
Model X	X1	X2	X3	X4
Model Y	Y1	Y2	Y3	Y4

(`http://tex.stackexchange.com/questions/94032/fancy-tables-in-latex`)

Arrays

Arrays are like tables, but in math mode.

Arrays

Arrays are like tables, but in math mode.

```
$\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}$
```

Arrays

Arrays are like tables, but in math mode.

```
$\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}$
```

$$\Longrightarrow \begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

Arrays

Arrays are like tables, but in math mode.

```
$\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}$
```

$$\Longrightarrow \begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

Add sides to the array by wrapping with `\left( ··· \right)`

Arrays

Arrays are like tables, but in math mode.

```
$\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}$
```

$$\Longrightarrow \begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

Add sides to the array by wrapping with `\left( ··· \right)`

```
$\left(\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}\right)$
```

Arrays

Arrays are like tables, but in math mode.

```
$\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}$
```

$$\Longrightarrow \begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

Add sides to the array by wrapping with `\left( ··· \right)`

```
$\left(\begin{array}{rrr}
   1 & -1 & 0 \\
   0 & 1 & 0 \\
   2 & 0 & 1
\end{array}\right)$
```

$$\Longrightarrow \left(\begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{array} \right)$$

Arrays

Arrays can have dividing lines, just like tables.

Arrays

Arrays can have dividing lines, just like tables.

```
$\left(\begin{array}{r|r|r}
   1 & -1 & 0 \\ \hline
   0 & 1 & 0 \\ \hline
   2 & 0 & 1
\end{array}\right)$
```

Arrays

Arrays can have dividing lines, just like tables.

```
$\left(\begin{array}{r|r|r}
   1 & -1 & 0 \\ \hline
   0 & 1 & 0 \\ \hline
   2 & 0 & 1
\end{array}\right)$
```

$$\Longrightarrow \left(\begin{array}{r|r|r} 1 & -1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 2 & 0 & 1 \end{array}\right)$$

Align

The `align` environment makes multiline equations look nice.

Align

The `align` environment makes multiline equations look nice.

```
\begin{align}
   C_n &= \frac{1}{n+1}{2n \choose n} \\
   &= \frac{(2n)!}{(n+1)!n!} \\
   &= \prod_{k=2}^n \frac{n+k}{k}.
\end{align}
```

Align

The `align` environment makes multiline equations look nice.

```
\begin{align}
   C_n &= \frac{1}{n+1}{2n \choose n} \\
   &= \frac{(2n)!}{(n+1)!n!} \\
   &= \prod_{k=2}^n \frac{n+k}{k}.
\end{align}
```

$\Longrightarrow$

$$C_n = \frac{1}{n+1}\binom{2n}{n} \tag{4}$$

$$= \frac{(2n)!}{(n+1)!n!} \tag{5}$$

$$= \prod_{k=2}^{n} \frac{n+k}{k}. \tag{6}$$

Align

The `align` environment makes multiline equations look nice.

```
\begin{align}
   C_n &= \frac{1}{n+1}{2n \choose n} \\
   &= \frac{(2n)!}{(n+1)!n!} \\
   &= \prod_{k=2}^n \frac{n+k}{k}.
\end{align}
```

$\Longrightarrow$

$$C_n = \frac{1}{n+1}\binom{2n}{n} \tag{4}$$

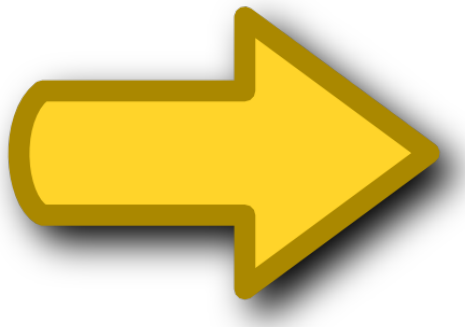
$$= \frac{(2n)!}{(n+1)!n!} \tag{5}$$

$$= \prod_{k=2}^{n} \frac{n+k}{k}. \tag{6}$$

You can hide the line numbers by using `align*` instead of `align`.

On the other hand, if you want to reference line numbers, after the equation type `\label{*}`, and in your body use *\ref*{*}.

On the other hand, if you want to reference line numbers, after the equation type `\label{*}`, and in your body use *\ref*{*}.



Test #4: Your three friends Anna, Bob, and Cathy all got different grades on the midterm and final. Make a table to display their grades, then display the values in a matrix.

Images

The easiest way to include images in your document is to use the `graphicx` package.

Images

The easiest way to include images in your document is to use the `graphicx` package.

The file types available depend on what you're using to compile. If you're compiling using `pdflatex` (recommended), then you can use `jpg`, `png`, `pdf`, or `eps` files.

Images

The easiest way to include images in your document is to use the `graphicx` package.

The file types available depend on what you're using to compile. If you're compiling using `pdflatex` (recommended), then you can use `jpg`, `png`, `pdf`, or `eps` files.

Place the file in the same directory as your `tex` file, and use the `\includegraphics` command.

Images

```
\includegraphics{gator}
```

Images

Images

```
\includegraphics[scale=0.5]{gator}
```

Images

Images

```
\includegraphics[width=5in]{gator}
```

Images

Images

```
\includegraphics[width=\linewidth]{gator}
```

Images

Images

```
\includegraphics[width=.5\linewidth,
                 angle=135]{gator}
```

Images

Images

```
\includegraphics[width=\linewidth,angle=135,
                 trim=12cm 22cm 12cm 5cm, clip]{gator}
```

Images

Deciphering Errors

Sometimes LaTeX is really helpful!

```
Underfull \hbox (badness 10000) in paragraph at lines 94--95


Underfull \hbox (badness 10000) in paragraph at lines 96--97

./demonstration.tex:98: Missing } inserted.
<inserted text>
                }
l.98 ...ts: $\sqrt{x} + \sqrt{y \neq \sqrt{x + y}$
                                                  .\\
?
```

Deciphering Errors

Sometimes LaTeX is really helpful!

```
Underfull \hbox (badness 10000) in paragraph at lines 94--95


Underfull \hbox (badness 10000) in paragraph at lines 96--97

./demonstration.tex:98: Missing } inserted.
<inserted text>
                }
l.98 ...ts: $\sqrt{x} + \sqrt{y \neq \sqrt{x + y}$
                                                  .\\
?
```

Other times...

```
./demonstration.tex:40: LaTeX Error: Something's wrong--perhaps a missing \item
```

Deciphering Errors

You'll see this a lot:

```
Runaway argument?
{\headcommand {\beamer@subsectionentry {0}{5
./Latex_2014_Slides_1.tex:63: File ended while scanning use of \@writefile.
<inserted text>
                \par
l.63 \begin{document}

?
```

Deciphering Errors

You'll see this a lot:

```
Runaway argument?
{\headcommand {\beamer@subsectionentry {0}{5
./Latex_2014_Slides_1.tex:63: File ended while scanning use of \@writefile.
<inserted text>
                \par
l.63 \begin{document}

?
```

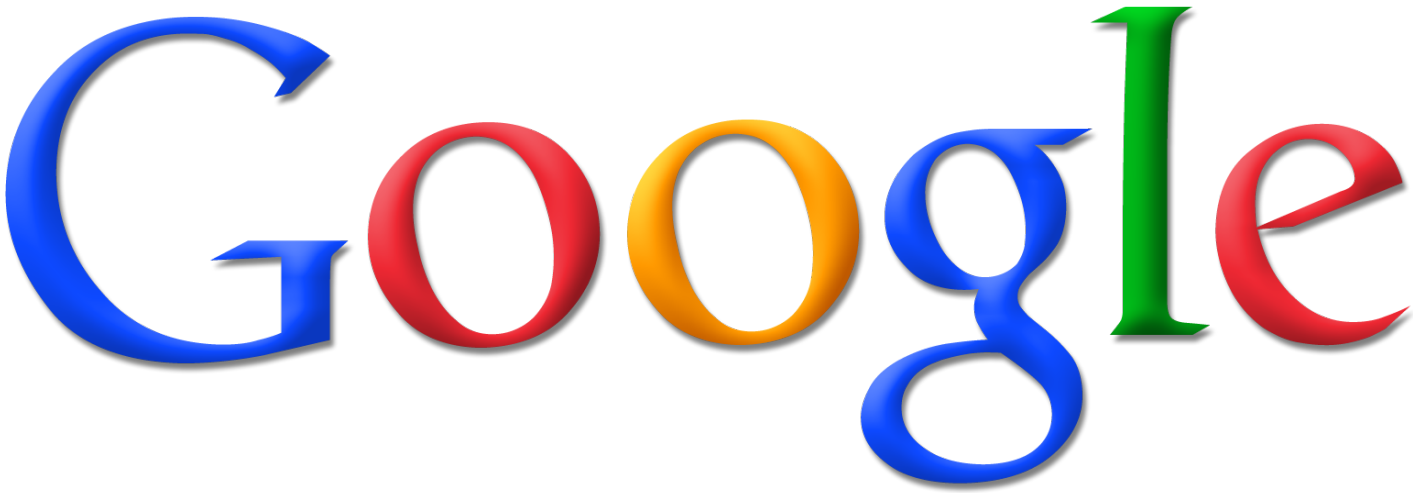
To fix it, clear your auxiliary files.

Deciphering Errors

What if you can't figure out how to fix an error?

Deciphering Errors

What if you can't figure out how to fix an error?



Types of Commands

We've previously mentioned some commands...

Types of Commands

We've previously mentioned some commands...

- Some have no arguments: `alpha` $\Longrightarrow$ α

Types of Commands

We've previously mentioned some commands...

- Some have no arguments: `alpha` $\Longrightarrow$ α
- Some have arguments: `frac{a}{b}` $\Longrightarrow$ $\frac{a}{b}$

Types of Commands

We've previously mentioned some commands...

- Some have no arguments: `alpha` $\Longrightarrow$ α
- Some have arguments: `frac{a}{b}` $\Longrightarrow$ $\frac{a}{b}$
- Some have optional arguments: `sqrt[n]{x}` $\Longrightarrow$ $\sqrt[n]{x}$

User-Defined Commands

You can define your own commands (also known as *macros*) in the preamble using the `newcommand` command.

User-Defined Commands

You can define your own commands (also known as *macros*) in the preamble using the `newcommand` command.

Format:

```
newcommand{yourcommandname}{[what it does]}
```

User-Defined Commands

You can define your own commands (also known as *macros*) in the preamble using the `newcommand` command.

Format:

```
newcommand{yourcommandname}{[what it does]}
```

Stupid example:

```
newcommand{me}{Bob}
```

Now, we can use `me` $\Longrightarrow$ Bob.

Less Stupid Example

The symbol $\setminus$ is created by `smallsetminus`, which is a lot of typing. We need a shortcut.

Less Stupid Example

The symbol $\setminus$ is created by `smallsetminus`, which is a lot of typing. We need a shortcut.

```
newcommand{ssm}{smallsetminus}
```

Less Stupid Example

The symbol $\setminus$ is created by `smallsetminus`, which is a lot of typing. We need a shortcut.

```
newcommand{ssm}{smallsetminus}
```

```
A ssm B
```
$\Longrightarrow$
AB

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Let's break the parts down:

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Let's break the parts down:

- [1] is the number of arguments

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Let's break the parts down:

- [1] is the number of arguments
- {#1} inserts the given argument

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Let's break the parts down:

- [1] is the number of arguments
- {#1} inserts the given argument
- ; adds a small space

Useful Example

User-defined commands can take arguments. This command shortens the name of `xrightarrow` and adds spacing.

```
newcommand{xto}[1]{xrightarrow{;;{#1};;}}
```

Let's break the parts down:

- [1] is the number of arguments
- {#1} inserts the given argument
- ; adds a small space

```
s_n xrightarrow{ntoinfty} 0
s_n xto{ntoinfty} 0
```

$\Longrightarrow$

$$s_n \xrightarrow{n\to\infty} 0$$
$$s_n n \to \infty 0$$

Really Useful Example

Arrays take a lot of typing.

Really Useful Example

Arrays take a lot of typing.

```
$\left(\begin{array}{rr}
   1 & -1 \\
   2 & 0
\end{array}\right)$
```

$$\Longrightarrow \left(\begin{array}{rr} 1 & -1 \\ 2 & 0 \end{array} \right)$$

Really Useful Example

Arrays take a lot of typing.

```
$\left(\begin{array}{rr}
   1 & -1 \\
   2 & 0
\end{array}\right)$
```

$$\Longrightarrow \left(\begin{array}{rr} 1 & -1 \\ 2 & 0 \end{array} \right)$$

Let's make a macro.

Really Useful Example

Arrays take a lot of typing.

```
$\left(\begin{array}{rr}
   1 & -1 \\
   2 & 0
\end{array}\right)$
```

$$\Longrightarrow \left(\begin{array}{rr} 1 & -1 \\ 2 & 0 \end{array} \right)$$

Let's make a macro.

```
\newcommand{\arr}[4]{
    \left(\begin{array}{rr}
        {#1} & {#2}
        {#3} & {#4}
    \end{array}\right)
}
```

Really Useful Example

Let's make a macro.

```
newcommand{arr}[4]{
    left(begin{array}{rr}
        {#1} & {#2}
        {#3} & {#4}
    end{array}right)
}
```

Really Useful Example

Let's make a macro.

```
newcommand{arr}[4]{
    left(begin{array}{rr}
        {#1} & {#2}
        {#3} & {#4}
    end{array}right)
}
```

Now we can make arrays much quicker.

`$arr{pi}{e}{gamma}{1}$` $\Longrightarrow \pi e \gamma 1$