Introduction to LaTeX

Scarlet & Jeremy

Cornell University

hy323 & jlf248

November 8, 2014

Overview

- Basic
  - Setting Up
  - Basic Writing
  - List Structures
  - Math Mode & Symbols
  - Math Paragraphs & Alignment
  - Multicols
- Intermediate
  - Defining & Renewing Commands
  - Boxes & Resizing
  - Tables, Graphs, & Figures
  - Writing Code
- Advanced
  - References
  - Presentations (Very Meta)

What is LaTeX?

- A document preparation and markup language for high-quality typesetting
  - i.e. it takes your confusing looking markup and compiles it to a beautiful PDF
- Widely used in academic research, especially the sciences
  - i.e. it gives off a very professional feel, even when what you're writing makes no sense ‿

Setting Up & Compiling

- Windows: use cygwin
- Mac: https://tug.org/mactex/
  - The instructions are really good
  - The files you have to download/install are huge (on the order of GBs)
- I highly recommend using an editor like TeXshop, TeXworks, Vim, Emacs, or even Sublime Text to write LaTeX with because they all are either built to compile your documents or have well supported plugins that do it all for you.

Basic Document Structure

- `\documentclass[options]{article}`
- Preamble – import external packages, set up page formatting, define your own macros, etc.
- `\begin{document}`
- Some sort of title
- `\section{Section Title}`
- Some content
- `\subsection{Subsection Title}`
- More content
- etc. etc. etc.
- `\end{document}`

Text Paragraphs

- Any time LaTeX sees a blank line, it treats the next line as the start of a new paragraph
- You can force new lines with \\
- Add indentation by \indent or suppress it with \noindent

```
This is the start of a new paragraph.
This is not, even though I'm on a new line.

This is another paragraph. \\
Now I've forced it onto a new line, but not a new
paragraph.
```

Text Style

- There are all the normal text styles you'd expect from a word processor, but for some you have to use external packages.
- For example, to use color you need to add `\usepackage{color}` to your preamble

`\textbf{Hello World}`	**Hello World**
`\emph{Hello World}`	*Hello World*
`\textsc{Hello World}`	Hello World
`\texttt{Hello World}`	`Hello World`
`\underline{Hello World}`	Hello World
`\textcolor{red}{Hello World}`	Hello World

Titles

- There are some shortcuts for you to make nice titles, but you don't *have* to use them
- A title will typically look like

```
\title{This is a Title}
\author{Scarlet \& Jeremy}
\date{November 2014}
\maketitle
```

Sections

- There are some shortcuts for you to make nice section headings (think chapters in a book, questions on a problem set, etc.)
- Sections will typically look like

```
\section{A big section title}
\subsection{A nested section title}
\subsubsection{An even more nested title}
```

Text Size

- There are built in commands for changing text size
- You can use them with `\textsize{your text}`
- Here are some examples

`\tiny`	tiny text
`\scriptsize`	scriptsize text
`\footnotesize`	footnotesize text
`\small`	small text
`\normalsize`	normalsize text
`\large`	large text
`\Large`	Large text
`\LARGE`	LARGE text
`\huge`	huge text
`\Huge`	Huge text

Enumerate

```
\begin{enumerate}
    \item one.
    \item two.
\end{enumerate}
```

1. one.
2. two.

```
\begin{enumerate}[a)]
    \item one.
    \item two.
\end{enumerate}
```

a) one.
b) two.

Itemize

```
\begin{itemize}
  \item one.
  \item two.
\end{itemize}
```

- one.
- two.

```
\begin{itemize}
  \item[Item1] one.
  \item[Item2] two.
\end{itemize}
```

Item1 one.

Item2 two.

Nested Lists

```
\begin{enumerate}
  \item enumerate1
    \begin{itemize}
      \item item1
      \item item2
    \end{itemize}
  \item enumerate2
\end{enumerate}
```

1. enumerate1
   - item1
   - item2
2. enumerate2

Math Mode

- Enter math mode using $ around texts

  `$ math mode $` *mathmode*

- Close $ to go back to text mode

  `$math$ text $math$` *math* text *math*

- Or use `\text` to temporarily go into text mode

  `$math \text{ text } math$` *math* text *math*

Commonly used symbols

`\geq \leq \neq`	$\geq \leq \neq$
`\oplus \times`	$\oplus \times$
`\cup \cap`	$\cup \cap$
`\vee \wedge`	$\vee \wedge$
`\forall \exists \in`	$\forall \exists \in$
`\because \therefore`	$\because \therefore$
`\rightarrow \leftarrow`	$\rightarrow \leftarrow$
`\Rightarrow \Leftarrow`	$\Rightarrow \Leftarrow$
`\equiv \approx \sim`	$\equiv \approx \sim$
`\LaTeX`	LaTeX

- `{ } _ ^ # & $ % ~` Needs to be escaped to be used in text mode.
- Use `http://detexify.kirelabs.org/classify.html`

Powers and Subscripts

- Powers, subscripts, fractions are written in math mode!

`a^b  a^{multichar}`	a^b $a^{multichar}$
`a_b  a_{multichar}`	a_b $a_{multichar}$
`\frac{a}{b}`	$\frac{a}{b}$

Math Paragraphs

- simple line of math can be written in `\[ ... \]`
- Automatically in math mode

```
\[
  \{(pk,sk) \leftarrow Gen(1^n) :
    (pk,Enc_{pk} (m_0))\}_n
\]
```

$$\{(pk, sk) \leftarrow Gen(1^n) : (pk, Enc_{pk}(m_0))\}_n$$

- But you can't linebreak!

Alignment

- Automatically in math mode
- `align` has equation numbers by default, `align*` doesn't

```
\begin{align}
  f(x) = 5x^2 + 4x + 3\\
  g(x) = 6x + 7
\end{align}
```

$$f(x) = 5x^2 + 4x + 3 \quad (1)$$
$$g(x) = 6x + 7 \quad (2)$$

```
\begin{align*}
  f(x) = 5x^2 + 4x + 3\\
  g(x) = 6x + 7
\end{align*}
```

$$f(x) = 5x^2 + 4x + 3$$
$$g(x) = 6x + 7$$

Alignment

- Align equations using anchors &
- Can be followed by anything, not just =

```
\begin{align*}
  f(x) &= (x + 2) - 2 \\
       &= x \\
       &\rightarrow wow!
\end{align*}
```

$$\begin{aligned} f(x) &= (x + 2) - 2 \\ &= x \\ &\rightarrow wow! \end{aligned}$$

More Alignment

- LaTeX assumes that each equation consists of two parts separated by anchor &

```
\begin{align*}
  x&=y          &  w &=z                  &  a&=b+c\\
  2x&=-y        &  3w&=\frac{1}{2}z       &  a&=b\\
  -4 + 5x&=2+y  &  w+2&=-1+w              &  ab&=cb
\end{align*}
```

$$\begin{aligned} x &= y & w &= z & a &= b + c \\ 2x &= -y & 3w &= \frac{1}{2}z & a &= b \\ -4 + 5x &= 2 + y & w + 2 &= -1 + w & ab &= cb \end{aligned}$$

Multicols

```
\begin{multicols}{2}
  \begin{enumerate} % chunk1
    \item A-one
    \item A-two
  \end{enumerate}
  \begin{enumerate} % chunk2
    \item B-one
    \item B-two
  \end{enumerate}
\end{multicols}
```

1. A-one
2. A-two

1. B-one
2. B-two

Multicols

```
\begin{multicols}{2}
  Latex is beautiful! It is easy to learn! \\
\columnbreak          % forces column break
  Computer Science is great!
  Hope this session is helping!
\end{multicols}
```

Latex is beautiful! It is easy to learn!

Computer Science is great! Hope this session is helping!

New Commands

- Define environments using `\newcommand{\your_command_name}{what it does}`
- Here's an example:

```
\newcommand{\scarlet}[1][Jeremy]
  {$S_cA^rL_eT$ and {#1}}
```

- Now you can use `\scarlet` in your document to produce this: $S_cA^rL_eT$ and Jeremy.
- Or you can use `\scarlet[Scarlet]` to produce this: $S_cA^rL_eT$ and Scarlet.

New Environments

- An environment is anything you use with `\begin{env} contents \end{env}`.
- Define new environments using `\newenvironment{name}[num]{before}{after}`
- Here's an example:

```
\newenvironment{newitemize}
  {Here goes nothing! \begin{itemize}}
  {\end{itemize} And so ends the list!}
```

- Now you can use it in your document to produce this: Here goes nothing!
  - item 1
  - item 2

  And so ends the list!

Renewing Commands

We can also overwrite existing commands (or ones that we made) using `\renewcommand` with the same syntax as `\newcommand`
A common example would be redefining how sections and subsections look using the following

```
\renewcommand{\thesection}
  {\Large Exercise \arabic{section}.}
\renewcommand{\thesubsection}
  {\normalsize (\alph{subsection})}
```

This will give you Exercise 1 as the section titles and (a) for the subsections.

Boxes & Resizing

- simple box around an equation

```
\begin{equation}
  \boxed{a = b}
\end{equation}
```

$$\boxed{a = b} \tag{3}$$

- For more capabilities, use empheq package

Tables

- Most of the time you won't want to do this by hand
- There are lots of online tools that make it easier
  - http://truben.no/table/
  - http://www.tablesgenerator.com/
- The basic gist is

```
\begin{table}
  \begin{tabular}{|l|r}
  h1   & h2   \\ \hline
  blah & blah \\
  blah & blah \\
  \end{tabular}
\end{table}
```

h1	h2
blah	blah
blah	blah

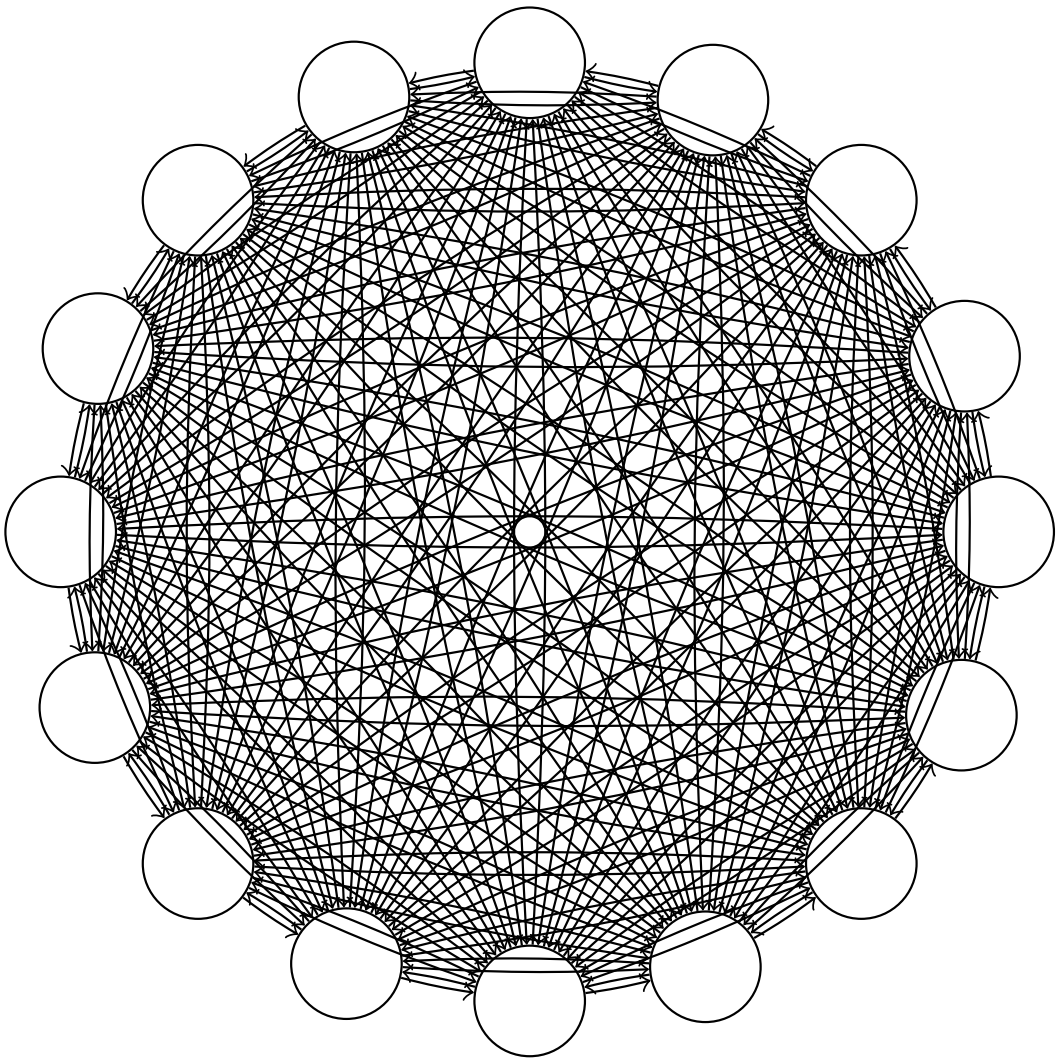
Tables

- The `{|l|r}` after `\begin{tabular}` specifies the alignment of each column. `{|l|r}` means the first column should be left aligned and the second column should be right aligned. The other option is using `c` for center alignment.
- We can put lines between columns by adding `|`. For example, this `{|c|c|c|}` will add lines around every column.
- The `&` separate the items in a row.
- We can add lines between rows by using `\\ \hline` instead of just `\\`.

Graphs

- For simple state machines/graphs, I recommend trying out this tool before jumping into the heavyweight stuff: http://madebyevan.com/fsm/.
- If that doesn't work, then use `tikz` by declaring `\usepackage{tikz}` in your preamble. And look up examples that are similar to what you want to do on http://www.texample.net/tikz/examples/.
- Some people get pretty crazy...

Graphs



Figures & Graphics

- Include the `graphicsx` package using `\usepackage{graphicsx}` in your preamble.
- Here's an example

```
\begin{figure}[h]
  \includegraphics[width=0.3\textwidth]
    {./awesomesauce.jpg}
  \caption{This is an amazing picture}
\end{figure}
```

Figure : This is an amazing picture

Figures & Graphics

- include path and extensions in preamble for ease

```
\graphicspath{ {./images/hw4/} }
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

- Then you can just put

```
\includegraphics[width=8cm]{awesomesauce}
```

Verbatim

- Write things within verbatim

```
\begin{verbatim.}
  public static void main(String[] args)
  {
    System.out.println("Hello World");
  }
\end{verbatim.}
```

Verbatim

- for short verbatim words, use `\verb`
- `\verb` turns words that are within the two separators into verbatim.
- Separater is the letter folling the `\verb`

```
\verb+Everything Here is Verbatim+
```

References

- You can reference nearly anything in LaTeX: equations, figures, graphs, tables, etc.
- Give the equation a `\label{name}` and then get a reference to it (i.e. Equation 1) using `\ref{name}`
- Here's an example

$$c = \sqrt{a^2 + b^2} \tag{4}$$

- This is a reference to Equation 4, and I didn't even have to keep track of the numbers; it's all done for you!

Presentations (Very Meta)

- Use beamer with `\documentclass{beamer}`
- There are a lot of combinations of themes and layouts (use this handy tool)
- Slides are defined using

```
\begin{frame}
  \frametitle{title}
  ...content...
\end{frame}
```

- And `\section`, etc. are used to break up slides so you can get neat labels like the ones up top!