How do LaTeX?

https://github.com/ekrause/LaTeX-Presentation

Eric Krause

Portland State University
M.S. ECE, 2013

February 19, 2014

Why use LaTeX?

- High quality output
- Unparalleled math/equation typesetting
- Powerful bibliography management
- Handles massive documents with ease
- Free and OS agnostic
- You get to use your favorite text editor
- Highly extensible
- **Focus on content, not formatting**

Don't use LaTeX if...

- Never used a computer before
- Can't spare a few hours of practice in exchange for a life changing skill
- Never need to create documents (why are you here?)
- Afraid of the command line
- Weak, lazy, other personal flaws

Downloading and Using LaTeX

Downloading

- **Linux:** Check your software repository.
  - `sudo apt-get install texlive-full`
- **OS X:** MacTex
  - `http://www.tug.org/mactex/`
  - `brew install pdflatex`
- **Windows:** MiKTeX
  - `http://miktex.org/`

Compiling [command line]

- `pdflatex -file-line-error -interaction=nonstopmode yourfile.tex`

Compiling [GUI]

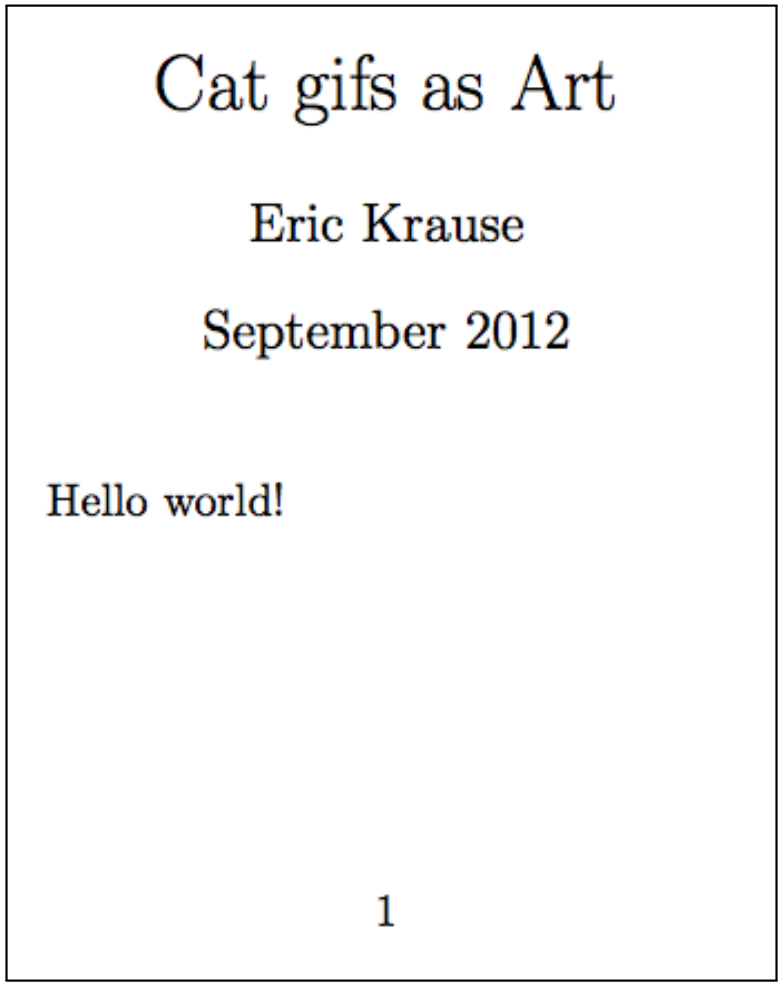
- Click buttons and/or mash keyboard.
- If that doesn't work, try touching the screen or using voice commands.

Hello LaTeX!

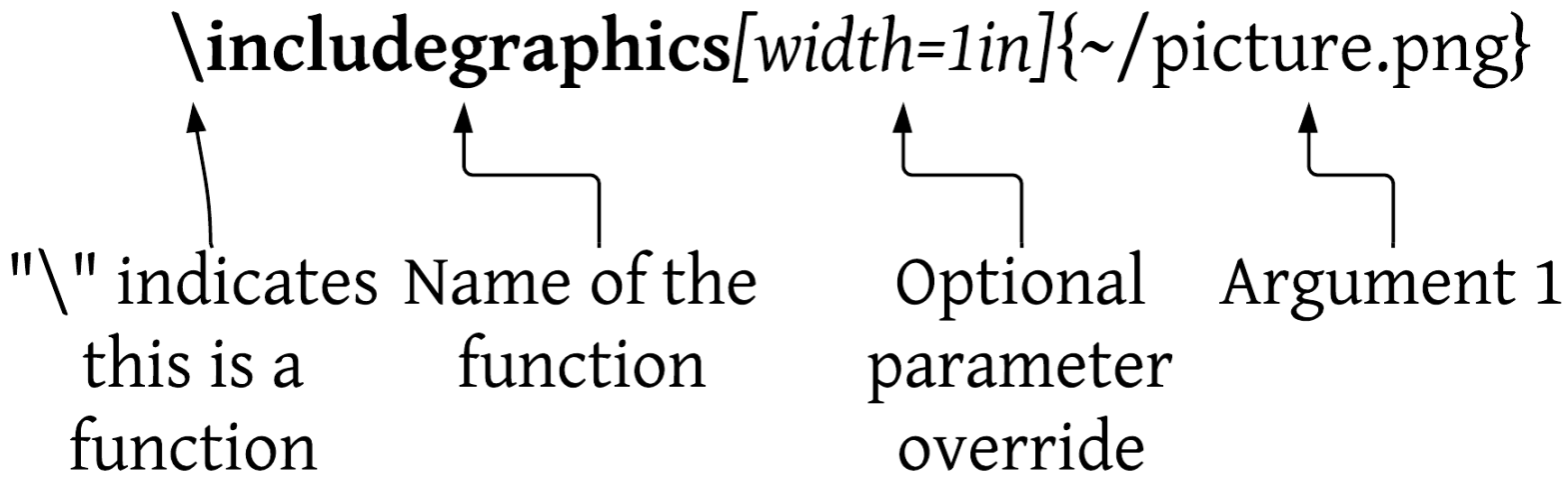
Source

```
1  \documentclass{article}
2
3  % this is a comment
4  \title{Cat gifs as Art}
5  \author{Eric Krause}
6  \date{September 2012}
7
8  \begin{document}
9
10 % generate title block
11 \maketitle
12
13 % let's add some content
14 Hello world!
15
16 \end{document}
```

Output



Function Syntax



Spaces and Escaped Characters

- Additional spaces between words are ignored.
- Manually add spaces by escaping a space '\ '
- Line break: (no indent) two backslashes '\\'
- Paragraph break (indent) two newlines ('Enter' twice)

	Unescaped Function	**To Print, Type:**
\	escape character, command identifier	\textbackslash
{ }	group and separate commands	\{ and \}
%	begin a line comment	\%
$	enter/leave math mode	\$
_	for subscripts (math mode)	_
^	for superscripts (math mode)	\textasciicircum
&	designate columns in tables	\&
#	reference arguments in functions	\#
~	insert unbreakable space	\textasciitilde

Common Text Formatting

Source

```
Formatting:\\
\textbf{Bold Text}\\
\textit{Italic Text}\\
\emph{Emphasized Text}\\
(\textit{emph is \emph{
    smarter} than textit})\\
\underline{Underlined Text}\\
\texttt{Monospace Text}\\
```

Output

Formatting:
Bold Text
Italic Text
Emphasized Text
(*emph is* smarter *than textit*)
Underlined Text
`Monospace Text`

Sections and Subsections

Source

```
1 \section{Vegetables}
2 Words.
3 \section{Fruits}
4 Section \thesection\ text.
5 \subsection{Apples}
6 More text.
7 \subsubsection{Braeburn}
8 Moar txt.
9 \section{Gourds}
10 wat.
11 \section*{Appendix}
12 Bees?
```

Output

1 Vegetables

Words.

2 Fruits

Section 2 text.

2.1 Apples

More text.

2.1.1 Braeburn

Moar txt.

3 Gourds

wat.

Appendix

Bees?

Lists

Source

```
The three types of lists:\\\\
Itemize:
\begin{itemize}
  \item these are bulleted
  \item as such
\end{itemize}
Enumerate:
\begin{enumerate}
  \item these are numbered
  \item as you might expect
\end{enumerate}
Description:
\begin{description}
  \item[thing A] text
  \item[thing B] words
\end{description}
```

Output

The three types of lists:

Itemize:

- these are bulleted
- as such

Enumerate:

1. these are numbered
2. as you might expect

Description:

item A text

item B words

Formatting Miscellanea

- Quotes:
  Backtick (`) for open quote, single quote (´) for close quote. ‘single’ or “double”
- Centering

```
\begin{center}
    %centered text
\end{center}
```

- Verbatim

```
\begin{verbatim}
    %won't be parsed by LaTeX
    %great for console output
\end{verbatim}
```

Math Mode

- `http://en.wikibooks.org/wiki/LaTeX/Mathematics`
- Begin and end inline equations:

```
$ math goes here $
```

- Begin and end display equations:

```
\begin{equation}
    math goes here
\end{equation}
```

- The only symbols accessed directly from the keyboard:
  - $+ - = ! / ( ) [ ] < > | ' :$

  The rest are all commands!
- Spaces are ignored entirely in equations.
- How to remember them all? Don't! Use **Detexify**!
  - `http://detexify.kirelabs.org/classify.html`

Math Mode Examples

Output:

Inline math mode:

I think $x(t) = a_0 + \sum_{n=1}^{\infty} [a_n cos(n\omega_0 t) + b_n sin(n\omega_0 t)] = \sum_{n=-\infty}^{+\infty} c_n e^{jn\omega_0 t}$ is a Fourier series or something.

Display math mode:

$$\frac{Dies}{Wafer} = \frac{\pi \cdot (Wafer\ Diameter/2)^2}{DieArea} - \frac{\pi \cdot Wafer\ Diameter}{\sqrt{2 \cdot Die\ Area}} \tag{1}$$

Use display math mode for centered, numbered equations like equation 1 that you can refer to later.

Source:

```
1  \noindent \textbf{Inline math mode:}\\
2  I think $x(t)=a_0 + \sum\limits_{n=1}^{\infty}[a_n cos(n\omega_0 t
       ) + b_n sin(n\omega_0 t)] = \sum\limits_{n=-\infty}^{+\infty}
       c_n e^{jn\omega_0t}$ is a Fourier series or something.\\
3
4  \noindent \textbf{Display math mode:}\\
5  \begin{equation}
6  \label{eq:dies}
7  \dfrac{Dies}{Wafer} = \dfrac {\pi \cdot (Wafer\ Diameter/2)^2}{Die
       Area} - \dfrac {\pi \cdot Wafer\ Diameter} {\sqrt{2 \cdot
       Die\ Area}}
8  \end{equation}
9  Use display math mode for centered, numbered equations like
       equation \ref{eq:dies} that you can refer to later.
10 \end{document}
```

Using the Tabular Environment

1. **Begin tabular mode** specifying the number of columns, alignment, and vertical lines.
2. **Input table rows** indicating separations between cells, specifying when to begin a new row, and where to include horizontal lines.
3. **End tabular mode**

Beginning A Tabular Environment

Started using the following command:

```
\begin{tabular}[]{column specification}
```

- The environment we are starting is `tabular`.
- The type, location, and alignment of columns and vertical lines is given using the `column specification`
  - l — left-aligned column
  - c — center-aligned column
  - r — right-aligned column
  - p{width} — paragraph column, must specify width.
  - | — vertical line (|| = double, ||| = triple ...)

Adding Contents

- Once in a tabular environment, table contents, separations between cells, and newlines are entered.
  - `&` — column separator
  - `\\` — start new row
  - `\hline` — horizontal line
  - `\newline` — start new line in cell (paragraph cells only)

Sample Table:

```
\begin{tabular}{|l||l|c|r|p{1in}||}\hline
            & -Tubes & -Sphere & -Net &  -Web \\\hline
  Blogo-    &   x    &         &      &  x    \\
  E-        &        &    x    &  x   &       \\
  Inter-    &        &    x    &      &       \\
  Web       &   x    &    x    &      &  x    \\\hline
\end{tabular}
```

Tabular Example

Output:

	-Tubes	-Sphere	-Net	-Web
Blogo-	x			x
E-		x	x	
Inter-		x		
Web	x	x		x

Importing Graphics

- `\usepackage{graphicx}`
- Once `graphicx` is included, images are imported using:

  `\includegraphics[options]{image name}`
- Useful optional parameters:
  - width=xx — manual width
  - height=xx — manual height
  - angle=xx — used to rotate image
  - scale=xx — manual scaling
- `[width=\textwidth] % full-page width`
- `[width=.5\textwidth] % half-page width`

Includegraphics Example

Source

```
1 \documentclass{article}
2 \usepackage{graphicx}
3 \begin{document}
4 \includegraphics[width=2in]
      {../Resources/cat.jpg}
5
6 \end{document}
```

Output

Floats

- A container that cannot be broken across multiple pages
- LaTeX defines `figure` and `table` floats
- Floats (should) have captions and references.
- Floats are automatically arranged by LaTeX , however you can manually specify placement

Float Placement

Format:

```
\begin{figure} [placement specifier]
    \label {fig:cat}
    \caption { this is a photo of a cat }
    %... figure contents ...
\end{figure}
```

- To get number of the float:
  `\ref{name}`
- Placement Specifiers:
  - `h` --- Place the float (approximately) here
  - `t` --- Position at the top of the page.
  - `b` --- Position at the bottom of the page.
  - `p` --- Put on a special page for floats only.
  - `!` --- Modifier. Override internal parameters LaTeX uses for determining "good" float positions.

Floats Example

Source

```
\begin{figure}[h!]
  \centering
  \label{fig:astrocat}
  \caption{An awesome cat}
  \includegraphics[width=1in]
      {cat.jpg}
\end{figure}

\begin{table}[h!]
  \begin{center}
    \begin{tabular}{| lcr |}
    \hline
    1 & 2 & 3 \\
    4 & 5 & 6 \\
    7 & 8 & 9 \\
    \hline
    \end{tabular}
  \end{center}
  \caption{A simple table}
```

Output

Figure 1: An awesome cat

1	2	3
4	5	6
7	8	9

Table 1: A simple table

Listings

- `\usepackage{listings}`
- Made specifically for listing source code.
- Syntax highlighting for all common languages.
- (Bad) Write/paste code into LaTeX document:

```
\begin{lstlisting}[options]
    Paste your code here
\end{lstlisting}
```

- (Good) Reference original source file:

```
\lstinputlisting[options]{filepath}
```

- `http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings#Settings`

Preferred Listing Settings

Source

```
\usepackage{listings}
\usepackage[usenames,dvipsnames]{color}
\begin{document}

\lstinputlisting[
    language=           Python,
    basicstyle=         \footnotesize,
    breaklines=         true,
    commentstyle=       \color{ForestGreen},
    keywordstyle=       \bf\color{RoyalBlue},
    stringstyle=        \it\color{Plum},
    numbers=            left,
    showstringspaces=   false,
    numberstyle=        \tiny\color{Gray},
    frame=              single,
    morekeywords=       {shuffle},
    caption=            Python Bogosort
    ]{../Resources/bogo.py}
```

Listings Example

Output

Listing 1: Python Bogosort

```
1  def bogo(x):
2      count = 0
3
4      while not inorder(x):
5          # if sorted, randomly shuffle
6          shuffle(x)
7          count = count + 1
8
9      # once sorted, return!
10     print "Sorted! Attempts: " +
           count
11     return x
```

Bibliographies with BibTeX

1. Create a bibliography (.bib) file

```
@article{Meyer2000,
  author = "Bernd Meyer",
  title = "A constraint-based framework for
      diagrammatic reasoning",
  journal = "Applied Artificial Intelligence",
  volume = "14",
  issue = "4",
  pages = "327--344",
  year = 2000
}
```

2. Cite source:

```
~\cite{Meyer2000}
```

3. Include at end of document:

```
\bibliography{bibfilename}
\bibliographystyle{plain}
```

4. Compile (with BibTeX)

Compiling with BibTeX

- Recommended method, according to `www.bibtex.org`
  1. `pdflatex mydocument`
  2. `bibtex mybib`
  3. `pdflatex mydocument`
  4. `pdflatex mydocument`
- Don't like that?

  `http://users.phys.psu.edu/~collins/latexmk/`

BibTeXDemo

BibTeXcitations are widely used in academics and available for free from ACM digital library, IEEE Xplore, and other libraries.

- `ACM demo`
- `IEEE Xplore Demo`

Example bibliography and cited document:

- `bibliography`
- `cited document`
- `final output`

Additional Resources

- All example code listed in this presentation (anything with line numbers) located in Examples/
- Many additional examples (omitted from presentation) located in Appendix/
  - Custom sizing
  - Algorithms
  - Defining new functions
  - Custom header files
- The source code from this presentation
- First places to go for help:
  - `http://detexify.kirelabs.org/`
  - `http://en.wikibooks.org/wiki/LaTeX`
  - `http://tex.stackexchange.com/`
  - `http://lmgtfy.com/?q=listings+latex`

Questions?

Presented By:

Eric Krause
PSU ECE Alumni
eric@sauerkrause.org
http://www.sauerkrause.org

Cody Gabriel
Eta Kappa Nu (HKN)
Iota Theta (PSU Chapter)
cwg2@pdx.edu

This presentation and source code available at:
https://github.com/ekrause/LaTeX-Presentation