Introduction to \LaTeX

Thanos Gentimis and Cheyne Homberger

Mathematics Department, University of Florida

March 29, 2012
<table>
<thead>
<tr>
<th>Motivation</th>
<th>Brief History</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Real Basics</td>
<td>What is \LaTeX?</td>
</tr>
<tr>
<td>Examples</td>
<td>\LaTeX\</td>
</tr>
</tbody>
</table>

\LaTeX\ is a programming language written by D.E. Knuth, the “Artist of Computer Programming.” \LaTeX\ is a system of macros written for \TeX\ by L. Lamport, so \LaTeX = \TeX\.
Who do we need to thank?

TeX is a programming language written by D.E. Knuth, the “Artist Of Computer Programming”.
Who do we need to thank?

\TeX{} is a programming language written by D.E. Knuth, the “Artist Of Computer Programming”.

What about \LaTeX{}?

\LaTeX{} is a system of macros written for \TeX{} by L. Lamport, \La + \TeX{} = \LaTeX{}.
What it’s not?

- LaTeX is NOT a MS Word/Open Office type of program.
- LaTeX is NOT a Matlab/Mathematica type of program.
- LaTeX is NOT an Excel type of program.
\LaTeX\ is NOT a MS Word/Open Office type of program.
What it’s not?

- \LaTeX{} is **NOT** a MS Word/Open Office type of program.
- \LaTeX{} is **NOT** a Matlab/Mathematica type of program.
What it’s not?

- \texttt{\LaTeX} is \textbf{NOT} a MS Word/Open Office type of program.
- \texttt{\LaTeX} is \textbf{NOT} a Matlab/Mathematica type of program.
- \texttt{\LaTeX} is \textbf{NOT} an Excel type of program.
It can be used to...

- Create documents which are heavily populated with math equations and symbols.
- Create presentations quickly out of those documents.
- Create a thesis!
It can be used to...

- Create documents which are heavily populated with math equations and symbols.
It can be used to...

- Create documents which are heavily populated with math equations and symbols.
- Create presentations quickly out of those documents.
It can be used to...

- Create documents which are heavily populated with math equations and symbols.
- Create presentations quickly out of those documents.
- Create a thesis!
In the first part of a LaTeX document, we define how our document will look like by choosing the document class (article, book, presentation etc). We list the packages we want to use, include the commands that we like (shortcuts) and tweak the format of the document.

The text and equations start after the command \begin{document} and end with \end{document}. This is the main body of our document.
Some commands

The easiest way is to insert your math equation between two dollar signs like: $ math, math, math $

You can also use the command \begin{equation} \end{equation} and there are a few more...
Some commands

- The easiest way is to insert your math equation between two dollar signs like: $ math, math, math $
Some commands

- The easiest way is to insert your math equation between two dollar signs like: $ math, math, math $.
- You can insert a mathematical equation if you include it between $[ and ]$. 
Some commands

- The easiest way is to insert your math equation between two dollar signs like: $ \text{math, math, math}$
- You can insert a mathematical equation if you include it between \[ and \]
- You can also use the command \begin{equation} \text{ and } \end{equation}
Some commands

- The easiest way is to insert your math equation between two dollar signs like: \$ \text{math, math, math} \$
- You can insert a mathematical equation if you include it between \[ and \]
- You can also use the command \begin{equation} \text{and} \end{equation}
- And there are a few more ...
Including formulas

Here is an example:

```
documentclass{article}
begin{document}
The following is the most beautiful math equation:
\[ e^{i\pi} + 1 = 0 \]
end{document}
```

Which gives the following:
The following is the most beautiful math equation:
\[ e^{i\pi} + 1 = 0 \]
Including formulas

Here is an example:

\begin{document}
\[e^{i\pi} + 1 = 0\]
\end{document}

Which gives the following:
The following is the most beautiful math equation:
\(e^{i\pi} + 1 = 0\)
Including formulas

Here is an example:
\documentclass{article}
\begin{document}
The following is the most beautiful math equation:
\[ e^{i\pi} + 1 = 0 \]
\end{document}
Including formulas

Here is an example:
\documentclass{article}
\begin{document}
The following is the most beautiful math equation:
\[ e^{i\pi} + 1 = 0 \]
\end{document}

Which gives the following:
Including formulas

Here is an example:
\documentclass{article}
\begin{document}
The following is the most beautiful math equation:
\[ e^{i\pi} + 1 = 0 \]
\end{document}

Which gives the following:

The following is the most beautiful math equation:

\[ e^{i\pi} + 1 = 0 \]
According to a lot of mathematicians the equation:

\[ e^{i\pi} + 1 = 0 \]

is the most beautiful of all!!
According to a lot of mathematicians the equation:
\[ e^{i\pi} + 1 = 0 \]
is the most beautiful of all!!
Different command?

Compare it to the next set of commands:

\documentclass{article}
\begin{document}
According to a lot of mathematicians the equation:
\[ e^{i\pi} + 1 = 0 \]
is the most beautiful of all!!
\end{document}
Different command?

Compare it to the next set of commands:

\documentclass\{article\}
\begin\{document\}
According to a lot of mathematicians the equation:
$e^{i\pi} + 1 = 0$
is the most beautiful of all!!
\end\{document\}

Which gives the following:

According to a lot of mathematicians the equation $e^{i\pi} + 1 = 0$ is
the most beautiful of all!!
Different enumeration commands

\begin{itemize}
  \item Example 1
  \item Example 2
\end{itemize}

\begin{enumerate}
  \item Item A
  \item Item B
\end{enumerate}
Different enumeration commands

- You can use the command \begin{itemize}, \end{itemize} and the different arguments are denoted by \item
Different enumeration commands

- You can use the command \begin{itemize}, \end{itemize} and the different arguments are denoted by \item
- You can use the command \begin{enumerate}[a)], \end{enumerate} and the different arguments are denoted by \item
Different enumeration commands

- You can use the command `\begin{itemize}`, `\end{itemize}` and the different arguments are denoted by `\item`.

- You can use the command `\begin{enumerate}[a)]`, `\end{enumerate}` and the different arguments are denoted by `\item`.

- And a few more ...
Example with a list

\begin{document}

$LaTeX$ is a great program because:

\begin{itemize}
  \item \textbf{\textalpha)} It is free!
  \item \textbf{\textbeta)} It is very easy once you work on it a bit.
  \item \textbf{\textgamma)} The math output looks great!
\end{itemize}

\end{document}
Example with a list

Here is an example:

\begin{itemize}
\item $\alpha$ It is free!
\item $\beta$ It is very easy once you work on it a bit.
\item $\gamma$ The math output looks great!
\end{itemize}
Example with a list

Here is an example:

\documentclass{article}
\begin{document}
$\LaTeX$ is a great program because:
\begin{itemize}
\item [\alpha)] It is free!
\item [\beta)] It is very easy once you work on it a bit.
\item [\gamma)] The math output looks great!
\end{itemize}
\end{document}
And the result

Motivation
The Real Basics
Examples
end

Format
Math formulas
Lists, bullets
Custom Format

LATEX

T. Gentimis

LaTeX is a great program because:

Alpha) It is free!

Beta) It is very easy once you work on it a bit.

Gamma) The math output looks great!
And the result

After you compile it:

\begin{itemize}
\item Alpha) It is free!
\item Beta) It is very easy once you work on it a bit.
\item Gamma) The math output looks great!
\end{itemize}
And the result

After you compile it:

\LaTeX is a great program because:

Alpha) It is free!
Beta) It is very easy once you work on it a bit.
Gamma) The math output looks great!
The role of documentclass

Notice that when we choose the class of our document we predefine the overall format settings, like the margins, the width and height of the text, the distance between letters and words. Try a few different ones (article, amsart, book) to see the results.

Still... we can customize it even further and even change the format of a single page!
The role of documentclass

Notice that ...

When we choose the class of our document we predefine the overall format settings, like the margins, the width and height of the text, the distance between letters and words. Try a few different ones (article, amsart, book) to see the results.
The role of documentclass

Notice that ... 

When we choose the class of our document we predefine the overall format settings, like the margins, the width and height of the text, the distance between letters and words. Try a few different ones (article, amsart, book) to see the results.

Still ... 

... we can customize it even further and even change the format of a single page!
Changing the overall look

If you want...

... You can configure various settings at the beginning of your document to make it look exactly like you want. For a complete list of all the different setting check our webpage. Here are a few:

- \texttt{hoffset=-15pt} Sets the horizontal offset of the page.
- \texttt{voffset=-40pt} Sets the vertical offset of the page.
- \texttt{topmargin=17pt} Sets the margin on the top of the page.
- \texttt{textwidth=17cm} Sets the width of the text.
- \texttt{textheight=10in} Sets the height of the text.
- \texttt{parindent=0cm} Sets leading space for paragraphs.
- \texttt{parskip=0cm} sets spacing between paragraphs.

Honestly...

You can change EVERYTHING about the format.

T. Gentimis
If you want ...

... You can configure various settings at the beginning of your document to make it look exactly like you want. For a complete list of all the different setting check our webpage. Here are a few:

- \textwidth=17cm
- \textheight=10in
- \topmargin=17pt
- \hoffset=-15pt
- \voffset=-40pt
- \parindent=0cm
- \parskip=0cm
Changing the overall look

If you want ...

... You can configure various settings at the beginning of your document to make it look exactly like you want. For a complete list of all the different setting check our webpage. Here are a few:

\hoffset=-15pt     Sets the horizontal offset of the page.
\voffset=-40pt      Sets the vertical offset of the page.
\topmargin=17pt     Sets the margin on the top of the page.
\textwidth=17cm     Sets the width of the text.
\textheight=10in    Sets the height of the text.
\parindent=0cm      Sets leading space for paragraphs
\parskip=0cm        sets spacing between paragraphs
Changing the overall look

If you want ...

... You can configure various settings at the beginning of your document to make it look exactly like you want. For a complete list of all the different setting check our webpage. Here are a few:

\hoffset=-15pt \hspace{1cm} Sets the horizontal offset of the page.
\voffset=-40pt \hspace{1cm} Sets the vertical offset of the page.
\topmargin=17pt \hspace{1cm} Sets the margin on the top of the page.
\textwidth=17cm \hspace{1cm} Sets the width of the text.
\textheight=10in \hspace{1cm} Sets the height of the text.
\parindent=0cm \hspace{1cm} Sets leading space for paragraphs
\parskip=0cm \hspace{1cm} sets spacing between paragraphs

Honestly ...

You can change EVERYTHING about the format.
Creating Your own commands

Including pictures

Find your favorite command (there are lists for all the commands online)

Use the `\newcommand` before the `\begin{document}`.

On the first slot you put your new way of calling the command always starting with `\`. On the second slot you put the name of the command you want to use.

There is a way to add a variable to your new command.
Find your favorite command (there are lists for all the commands online)
Find your favorite command (there are lists for all the commands online)

Use the `\newcommand {} {}` before the begin document.
- Find your favorite command (there are lists for all the commands online)
- Use the `\newcommand {} {}` before the begin document.
- On the first slot you put your new way of calling the command always starting with `\`. 
Find your favorite command (there are lists for all the commands online)

Use the \newcommand \{\} \{\} before the begin document.

On the first slot you put your new way of calling the command always starting with \\

On the second slot you put the name of the command you want to use.
- Find your favorite command (there are lists for all the commands online)
- Use the `\newcommand \{\}\{\}` before the `\begin{document}`.
- On the first slot you put your new way of calling the command always starting with `\`.
- On the second slot you put the name of the command you want to use.
- There is a way to add a variable to your new command.
Basic Steps

Get your favorite picture. Change it into eps format (not needed but looks much better)

A great site that converts pictures is http://image.online-convert.com/convert-to-eps

Make sure you put the picture in the same folder.

Use the package graphicx

Use the command includegraphics.
Basic Steps

- Get your favorite picture. Change it into eps format (not needed but looks much better)
Basic Steps

- Get your favorite picture. Change it into eps format (not needed but looks much better)
- A great site that converts pictures is http://image.online-convert.com/convert-to-eps
Basic Steps

- Get your favorite picture. Change it into eps format (not needed but looks much better)
- A great site that converts pictures is http://image.online-convert.com/convert-to-eps
- Make sure you put the picture in the same folder.
Basic Steps

- Get your favorite picture. Change it into eps format (not needed but looks much better)
- A great site that converts pictures is http://image.online-convert.com/convert-to-eps
- Make sure you put the picture in the same folder.
- Use the package graphicx
Basic Steps

- Get your favorite picture. Change it into eps format (not needed but looks much better)
- A great site that converts pictures is http://image.online-convert.com/convert-to-eps
- Make sure you put the picture in the same folder.
- Use the package graphicx
- Use the command includegraphics.
Links

http://www.math.ufl.edu/gma/textalk
http://tobi.oetiker.ch/lshort/lshort.pdf
The End ...
The End ...

... or maybe the beginning of \LaTeX\ for you?