GVSU 2009 Summer REU - LaTeX Seminar

June 15, 2009

What is LaTeX ?

LaTeX is

- a typesetting system developed by Leslie Lamport based on Donland E. Knuth's TeX typesetting language.

LaTeX is not

- a what–you–see–is–what–you–get word processor (like MS Word).

A LaTeX file is created using a text editor (for example, WinEdt), then compiled into a fully formatted document by the LaTeXprogram. All formatting is inserted directly into the text in the form of commands, very similar to directly editing an HTML document and inserting the HTML code.

LaTex Features...

LaTeX contains features for:

- Typesetting journal articles, technical reports, books, and slide presentations.
- Control over large documents containing sectioning, cross-references, tables and figures.
- Typesetting of complex mathematical formulas.
- Automatic generation of references/bibliographies.
- Inclusion of artwork and graphical images.

Installation ...

- Download LaTeX from the internet. The windows version that is currently used by most people is called "MikTex" and is available for FREE from: `http://www.miktex.org`.
- Next we need a nice editor. Currently the most popular is "WinEdt", which is a shareware, i.e almost free.(the student version is $30 and the academic rate is $40.) It is available from: `http://www.winedt.com`.

Both MikTeX and WinEdt are installed in your Desktop computers.

Learning LaTeX– learn by doing, use templates and example files.

Creating Basic LaTeX Document ...

All we need to do is type the following using WinEdt into a file.

```
\documentclass{article}

\begin{document}
your text goes here...
\end{document}
```

What does each line mean?

- `\documentclass{article}` – tells LaTeX to use the standard *article* formatting. Other classes: *book*, *letter*, *report* ...
- `\begin{document}` – tells LaTeX the beginning of the document, and that most of the document formatting commands have been stated. Other declarations such as such as margins and other packages to be included are entered between this line and `\documentclass` command.
- `\end{document}`– tells LaTeX that this is the end of the document.

Compiling Basic LaTeX Document ...

- To see what the document look like, we need to compile it. (It is always a good practice to compile twice)
- To compile click on the "LaTeX" symbol in WinEdt.
- If the compilation is successful, LaTeXoutputs `.dvi` file.
- To see the `.dvi` file click on the "dvi" symbol in WinEdt.
- To create a `pdf` file click on the "dvi→pdf" symbol.

Comments in LaTeX...

To insert a comment into the file that will not be compiled into the main document, insert a % at the beginning of the line. For example:

```
%Remember to insert the figure in here.
```

Special Characters in LaTeX...

To get your quotation marks to appear pretty, use ' ' and ' '
instead of " and ". There are some characters that
LaTeX considers commands, so if you want them in your
document, you need to type them in a special way. If you want any
of these characters to occur in you printout:

    & $ #   %   _   {   }

You will need to type:

    \&   \$ \#   \%   _   \{   \}

Title, Author, date ... in LaTeX

Title, Author, date, etc., are specified as in the following example:

```
\documentclass{article}

\author{Jon Doe}
\title{A Brief Introduction to \LaTeX\ }
\date{\today}

\begin{document}

\maketitle

......

\end{document}
```

Title, Author, date...

- The \maketitle instruction which causes the title, author, etc to be printed, goes after \begin{document}
- The command \today instructs LATEXto use today's date. You can, of course, specify a date explicitly by saying something like \date{June 14, 2009}.

Sectioning in LaTeX

Titles of sections, subsections, etc., are specified by using commands like the following in the body of the document (i.e., between \begin{document} and \end{document}).

\section{Section 1 Title goes here}

\subsection{Subsection 1 title goes here}

\subsection{Subsection 2 title goes here}

....

\section{Section n title goes here}

\section{Conclusion}

Sectioning in LaTeX

Note: LaTeX automatically numbers the sections. The numbering can be prevented by using "asterisk" versions of the sectioning commands: \section*{...}, \subsection*{...}, etc.

Typesetting mathematical expressions ..

The $ before and after an expression tell LaTeX to go into "math mode", but to stay on the same line. Use the $ method to insert an equation or variable into a line of text.

- An exponent can be displayed using the ^ and maybe some {}: c^2 or e^{-3i} can be typeset with the code c^2 or e^{-3i}.
- A subscript can be entered using the _ and maybe some {}: the code x_1 or x_{i+j} yields x_1 or x_{i+j}.
- To display an equation centered on a separate line, enclose it by a pair of brackets (\[and \]) For example:
  ```
  \[
  c^2 = a^2 + b^2
  \]
  ```

Typesetting mathematical expressions ...

- If you'd like to number an equation, use the equation mode:

```
\begin{equation} \label{eq:pytha}
c^2 = a^2 + b^2
\end{equation}
```

To reference this equation later, use the \ref command:

```
From equation (\ref{eq:pytha}) we see that . . .
```

Note: To see the equation reference number displayed correctly, you will need to compile the document TWICE.

Typesetting mathematical expressions ...

- Fractions and Binomial Coefficients are typeset as follows:
  - $\frac{x}{y}$ is typeset with `$\frac{x}{y}$`
  - $\binom{x}{y}$ is typeset with `$\binom{x}{y}$` or
    `${{x} \choose {y}}$`
- Sums and Integrals are typeset as follows:
  - The symbol for sum is `\sum`. For example, to display the
    identity: $\displaystyle\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$, one can use:
    ```
    \[
    \sum_{k = 1}^{n} k = \frac{n(n + 1)}{2}
    \]
    ```
  - The symbol for single integral is `\int`. To get $\int_0^1 f(x)\,dx$, one
    can use `$\int_{0}^{1}f(x)\,dx$`

Typesetting mathematical expressions ...

Limits Example:

- $\displaystyle \lim_{x \to 0} f(x) = f(0)$ is typeset with
  `$\displaystyle \lim_{x \to 0} f(x) = f(0)$`

Definitions and Theorems

LaTeX has environments for displaying definitions and theorems.
For example, the LaTeXcodes:

```
\begin{defn}\label{def:ab-int}
An integer $n$ is said to be \emph{Aboufadelian}
if $n^2 + 3n - 10 = 0$.
\end{defn}
```

and

```
\begin{thm}\label{thm:ab-int}
 There are exactly two Aboufadelian integers.
\end{thm}
```

generate the following.

Definition
An integer n is said to be *Aboufadelian* if $n^2 + 3n - 10 = 0$.

Theorem
There are exactly two Aboufadelian integers.

Definitions and Theorems

- Note that the environments defn and thm are user defined in the preamble using the \newtheorem command.
- The commands \label{def:ab-int} and \label{thm:ab-int} assign labels to the definition and the theorem, respectively. These labels are used to refer the definition and the theorem in the document.
- To refer labeled definitions and theorems we use the \ref command. For more detail explanation read the sections "Definitions and Theorems", and "Referencing" in the handout.

Importing graphics

- First we need to include the command
  \usepackage{graphicx} in the preamble of our LATEXsource
  document.
- We can import graphics having the following formats:
  encapsulated postscript (.eps), bitmap (.bmp), or JPEG
  (.jpg.
- DVI documents like bitmaps and .eps files, while PDF
  documents like .eps and JPEG files.
- We use \includegraphics command to include graphics.
  For example, the code:

  \begin{center}
  \includegraphics[width = 4 in, height = 2 in]{c:/images
  \end{center}

  produces the following figure:

Importing graphics
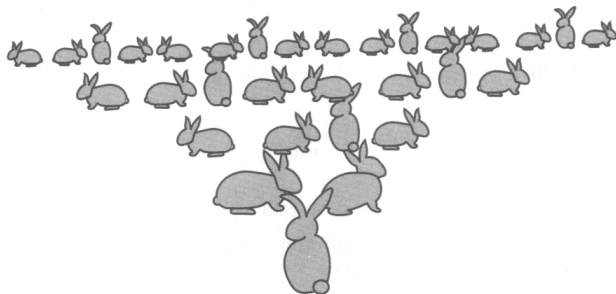
- Since eps images work best both in DVI and PDF files, it is recommended to convert bitmap and JPEG images to eps format
- you can do the conversion for free at
  http://www.online-utility.org/image_converter.jsp

Overhead Slides using LATEXBeamer

- Beamer is a LATEX class that can be used to make nice overhead slides with all of the features you know and love from your normal LATEX documents.
- A beamer document consists of a sequence of overhead slides, known as *frames*.
- To process a beamer source file you need to click the symbol PDFLATEX in WinEdt.
- As with learning LATEX, the best way to learn Beamer is to simply play with some examples. The following examples help you learn and master the basics of beamer. (The files are listed in the order of complexity)
    1. `slidesTemplate.tex` – A template that shows simple *beamer* slides structure.
    2. `simpleslides.tex` – A sample file that illustrates how to make some simple overhead slides using the *beamer* class.
    3. `latextalk.tex` – a LATEX source file of this LATEX tutorial presentation.