A brief introduction to LaTeX

Chris Bowers

October 16, 2007

What is LaTeX?

- TeX developed late 70's as typesetting language.
- LaTeX is a set of macro extensions to TeX
- It's a document preparation system.
- It was designed for large technical documents.
- Now used in a much wider context.

Advantages of LaTeX

- Seperates content from formatting.
- Good for technical documents.
- Can handle complex mathematical expressions.
- Very adaptive yet consistent.
- It's free on all platforms.

Disadvantages of LaTeX

- Not word processor like MS Word.
- Needs to be compiled.
- It's not WYSIWYG.
- Easy to use, but difficult to master.

Whitespace

```
consecutive white spaces    are treated
     as           one,
as    are    single    line      breaks.
```

consecutive white spaces are treated as one, as are single line breaks.

Whitespace

```
An empty line

signifies a new paragraph.
```

An empty line

signifies a new paragraph.

LaTeX commands

All LaTeX commands begin with a backslash, \.

- `\textit{text}` → *text*
- `\textbf{text}` → **text**
- `Today is \today` → Today is October 15, 2007.

Command arguments {} - parameters that are required.

[] - parameters that are optional.

A simple example

```
\documentclass[12pt]{article}  - defines document type
\usepackage{times}             - use times font package
\begin{document}               - start of document contents
   \textbf{Hello World!}       - the document contents
\end{document}                 - end of document contents
```

Hello World!

Document Classes

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

Example:

```
\documentclass[10pt, twocolumn]{article}
```

The document is an *article* with a main font of 10pt and the text flowing in two columns.

Packages

Packages are how the basic features of latex are enhanced.

```
\usepackage[options]{package}
```

Example:

```
\usepackage[compact]{titlesec}
```

The *titlesec* package provides new features for controlling the appearance of titles.
The *compact* option reduces spacing around titles.

Reserved Characters

LaTeX has several characters with special meanings.

If they are required in the text then the following representations must be used:

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

Getting yourself started

- LaTeX documents need to be compiled.
- In some cases several times to build auxillary files for indexes and references.
- Compilation can result in:

Errors	which are critical and may prevent the production of a legible typeset document.
Warnings	which are are not critical but may result in unexpected behaviour.
Bad Boxes	indicate items which don't fit as well as they could - not really serious.

Compiling LaTeX

At the command line: `latex document.tex`

Should result in something like:

```
LaTeX-Result: 0 Error(s), 3 Warning(s), 2 Bad Box(es), 12 Pages(s)
```

May need to repeat this process if using BibTeX or other indexes (more later).

Where to get LaTeX

Windows MikTex(miktex.org) is probably the best and free under GNU license.

Linux probably already installed or available as rpm or similar.

Mac MacTeX (tug.org/mactex)

LaTeX development environments

Several purpose built LaTeX IDE's now exists.

Windows	TeXnicCenter	www.toolscenter.org
	LEd	latexeditor.org
Linux	Kile	kile.sourceforge.net
Mac	TeXShop	uoregon.edu/ koch/texshop

The Various TEX macros

tex	compiles TEX into a .dvi format.
latex	compiles TEX with LATEX2e extensions to .dvi format.
pdfTex	as above but compiles straight to PDF.
bibtex	compiles Bibliography sources for use in LATEX documents (more on this later).

Be prepared

Have a look at some of the compiler and IDE options available. Try compiling your own simple LaTeX document. Find an IDE that suits you.

Building a structured document 1

The pre-amble should define the document type and what packages you wish to use.

Generally anything that applies to the entire document.

Common examples include:

- `\title{title of document}`
- `\author{authors' name}`
- `\date{some date}`

Building a structured document 2

The main document contents are defined within the document environment:

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

Building a structured document 3

The document can be split into sections:

```
\section{new section heading}
```

or subsections:

```
\subsection{new subsection heading}
```

Building a structured document 4

For example:

```
\documentclass[12pt,a4paper]{article}
\usepackage[compact]{titlesec}

\title{Yet Another Dull Story}
\author{John Smith}

\begin{document}

\section{Introduction}
This is the introduction followed by a subsection.

\subsection{First subsection}
This is the first subsection

\section{New Section}
Here's a totally new section.

\end{document}
```

LaTeX tables 1

- Tables are created within the *tabular* environment.
- In the simplest case LaTeX automatically controls table spacing and widths.

To use the tabular environment:

```
\begin{tabular}[pos]{table_spec}
```

where the *table_spec* instructs LaTeX on how to structure the columns of the table and *pos* defines where to place the document.

LaTeX tables 2

```
\begin{tabular}{lcr}
```

- would align the first column to the left, second would be centred whilst the third would be to the right.

```
\begin{tabular}{|c|c|c|}
```

- would centre the contents of each column and place a verticle seperation line.

```
\begin{tabular}{|c||p{4.5cm}||c|}
```

- same as above except middle column is justified at 4.5cm wide and surrounded by double verticle seperation lines.

LaTeX tables 3

Content is added in rows

```
row1 & row2 & row3 \\
```

- `&` - mark the column seperation
- `\\` - marks the end of a row.
- `\hline` - used to draw horizontal lines across the table.

LaTeX tables 4

An example table:

```
\begin{tabular}{|c|c|}
\hline
Constant & Value \\
\hline
$\pi$ & 3.142 \\
c & 299792458 m/s \\
\hline
\end{tabular}
```

Constant	Value
π	3.142
c	299792458 m/s

Maths mode

LaTeX has a special mode for writing mathematical text - `$`.

Example:

`$ E = mc^2 $` → $E = mc^2$

There are lots of mathematical symbols which can be represented in math mode:

`$\infty\hbar\sum\int\partial\beta\Psi$` → $\infty\hbar\sum\int\partial\beta\Psi$

There are also mathematical strcutures:

`$\frac{top}{bottom}$` → $\frac{top}{bottom}$

`$\sqrt{val}$` → $\sqrt{val}$

Including figures 1

- A figure is most commonly in the form of an imbedded image.
- All LaTeX documents are ultimately used to generate either a postscript (.ps) file or pdf.
- For a postscript file all embedded images must be in the form of encapsulated postscript (.eps).
- Modern LaTeX graphics packages can handle on-the-fly conversion from a range of image formats.

Including figures 2

```
\includegraphics[width=5cm]{misty.jpg}
```

You can include all kinds of files as long as there is a suitable conversion tool into postscript - *dvips*.

You may need to include a suitable graphics package and converter- `\usepackage[dvips]{graphicx}`

pdfTex or other PDF orientated compilers tend to be easier but no .eps support.

Floating bodies 1

- Allow LaTeX to position a section of content.
- Useful for figures, tables or equations in large documents.
- LaTeX is very good at adjusting layout to ensure consistant spacing and reduce gaps.
- The most common floating environments you will see are `\begin{table}[pos]` and `\begin{figure}[pos]`.
- *pos* gives some control over where LaTeX positions the float.

  h right here
  t top of a page
  b bottom of a page
  p at then end of this chapter/section/document
  ! ignore LaTeXand just do as I say!

Floating bodies 2

Of course you may also need a caption for a figure or table.
Example:

```
\begin{figure}[htp]
  \includegraphics{file.eps}
  \caption{A simple figure}
\end{figure}
```