LaTeX basics

Brian Buccola & Alanah McKillen
February 2, 2012
McGill University

1 Overview

LaTeX is a markup language (and program) for typesetting documents. It consists of many commands and environments that specify (to the compiler) how the output document should look. In general, commands produce or modify a small bit of text, whereas environments modify a larger part of text.

1.1 Document structure

- Every document consists of a "preamble" and a "body".
- The preamble includes declarations of document classes, packages, definitions, and other document-wide specifications.
- The body includes the main text (title, sections, ...).

Example

```
% My first LaTeX document!

% PREAMBLE
\documentclass{article}
\usepackage{ulem}

% BODY
\begin{document}
Hello beautiful world! \sout{Goodbye cruel world!}
\end{document}
```

Example output

Hello beautiful world! ~~Goodbye cruel world!~~

1.2 Commands and environments

- Every command starts with a backslash, is case-sensitive, and may take options (in square brackets) and arguments (in curly brackets).

Syntax

```
\command[option1,option2,...]{arg1}{arg2}...
```

Example

```
\textbf{This sentence will be bold.}
```

Example output

This sentence will be bold.

- Some commands do not require options or arguments; they simply produce something as is.

  Example

  ```
  \ldots
  ```

  Example output

  ...

- Every environment starts with a `\begin` commad and ends with an `\end` command, whose arguments are the environment name.

  Syntax

  ```
  \begin{environmentname}
  Text to be modified.
  \end{environmentname}
  ```

  Example

  ```
  \begin{enumerate}
  \item First item in a numbered list.
  \item Second item.
  \end{enumerate}
  ```

  Example output

  1. First item in a numbered list.
  2. Second item.

1.3 Special characters and accents

- LaTeX reserves a number of characters for special purposes, such as curly brackets, as we've seen. The following is a more complete list.

  ```
  # $ % ^ & _ { } ~ \
  ```

- There are usually two ways to produce these characters: (1) use a backslash to "escape out", e.g. `\#` produces #, and (2) use the relevant command for that symbol, e.g. `\textbackslash` produces \.

- Accents and other diacritics are produced similarly.

  Syntax

  `\symbol{character}`

  Example

  `\^{a} \"{a} \~{a}`

  Example output

  â ä ã

- Beginning and ending single quotation marks are produced using ` (grave accent) and ' (vertical quote), respectively; double quotes are produced by doubling these characters, *not* by using the double quotation character ".

  Example

  ```
  Bill said, ``Hello''.
  ```

  Example output

  Bill said, "Hello".

1.4 Whitespace and comments

- Consecutive "whitespace" characters (spaces, tabs) are treated as one "space".

  Example

  ```
  The extra whitespace right here        will not show up
  in the output.
  ```

  Example output

  The extra whitespace right here will not show up in the output.

- Similarly, single line breaks are treated as one "space".

  Example

  ```
  This
  will
  all be
  on one line.
  ```

  Example output

  This will all be on one line.

- A double linebreak indicates the start of a new paragraph; more than that is treated as a double linebreak.

  Example

  ```
  This is one paragraph that stretches stretches
  stretches stretches stretches onto multiple lines.

  This begins a new paragraph.
  ```

  Example output

  This is one paragraph that stretches stretches stretches stretches stretches onto multiple lines.
  This begins a new paragraph.

- You can insert your own comments by appending each comment line with `%`; the compiler ignores anything following a `%` on a given line.

  Example

  ```
  Some \textit{italic} text. % My comment about fun
                             % with italics
  ```

  Example output

  Some *italic* text.

2 Typesetting: some specifics

The beauty of LaTeX is that, with just a handful of basic commands, you can write a nicely structured and good-looking document without the distractions of tweaking and fiddling that come with WYSIWYG word processors.

2.1 Document structuring commands

- To automatically create a title page, first in the preamble, use the `\title`, `\author`, and `\date` commands to define the title, author, and date; then use the `\maketitle` command in the body to create a title page.

  Example

  ```
  \documentclass{article}
  \title{Minimalist Program II: Maximize Minimalism}
  \author{Noam Chomsky}
  \date{\today}
  \begin{document}
  \maketitle
  ```

```
In this paper, I revise my earlier proposal \ldots
\end{document}
```

- *Tip:* Use the `\today` command to automatically insert the day on which the document is compiled.
- To add an abstract, use the `abstract` environment.

Example

```
\begin{abstract}
This paper defends the geocentric model of the
universe.
\end{abstract}
```

- To create section and subsection headings, use the `\section` and `\subsection` commands. They take the (sub)section name as their argument are numbered automatically.

Syntax

```
\section{Section Name}
```

- *Tip:* To exclude a (sub)section number, use `\section*` and `\subsection*` instead.
- To add a footnote, use the `\footnote` command. It takes the entire footnote text as its argument.

Syntax

```
Here's some text.\footnote{Here's a footnote!}
```

2.2 Modifying text

- To make text bold, italics, smallcaps, or typewriter, use the `\textbf`, `\textit`, `\textsc`, and `\texttt` commands.
- Alternatively, enclose the text in curly braces, preceded by the commands `\bfseries`, `itshape`, `\scshape`, or `\ttfamily`, commands.

Example

```
This is some \textbf{bold}, \textit{italic},
\textsc{smallcaps}, and \texttt{typewriter} text.

This is some {\bfseries bold}, {\itshape italic},
{\scshape smallcaps}, and {\ttfamily typewriter} text.
```

Example output

This is some **bold**, *italic*, SMALLCAPS, and `typewriter` text.
This is some **bold**, *italic*, SMALLCAPS, and `typewriter` text.

- *Tip:* Some of the fullform commands can be nested: `\textbf{\textit{hi!}}` produces ***hi!***.
- To change font size, use the commands `\tiny`, `\small`, `\large`, `\Large`, `\LARGE`, etc. (See documentation for full list.)

Example

```
Some {\tiny tiny}, {\small small}, normal, {\large
large}, {\Large larger}, {\LARGE even larger}
text.
```

Example output

Some tiny, small, normal, large, larger, even larger text.

- These font sizes are relative to the font size option declared in the document class, e.g. `\large` in a `12pt` document is roughly the same size as `\Large` in a `10pt` document.

2.3 Math mode

- To produce math symbols, enclose your math text in dollar symbols; this is called math mode.

Example

```
$\forall x \in \mathcal{P}(A): \exists y \in
\phi(B \times C): y = x^5 - \alpha_3$
```

Example output

$\forall x \in \mathcal{P}(A) : \exists y \in \phi(B \times C) : y = x^5 - \alpha_3$

- Math symbols can be thrown right into normal text.

Example

```
If $\alpha$ is of type $a \rightarrow b$ \ldots
```

Example output

If α is of type $a \rightarrow b \ldots$

2.4 Some useful commands

`\ldots` create ellipsis dots

`\noindent` disable indent for this paragraph

`\hfill` fill horizontal space (moves the following text all the way rightward)

2.5 Some useful environments

- Here's a list of some useful environments, all of which follow the syntax described earlier.

`itemize`	create bulleted lists
`enumerate`	create numbered lists
`center`	center a block of text
`flushleft`	left-align a block of text
`flushright`	right-align a block of text[1]
`tabular`	create a table
`figure`	create a figure
`quote`	create a block quote

3 Next time?

Advanced topics

- bibliography (BibTEX)
- labels and references
- defining new commands and environments; redefining current ones
- hyperlinks, PDF metadata (`hyperref`)
- presentation slides (`beamer`)
- special headers and footers (`fancyhdr`)
- custom sections (`sectsty`)

Linguistics-specific topics

- numbered examples (`linguex`)
- syntax trees (`qtree`)
- IPA (`tipa`)

[1] Text is left- *and* right-aligned (justified) by default.