Introduction to LaTeX

A Beginner's Guide

This is a simple guide that shall serve as an introduction to LaTeX (ˈlɑːtɛk or ˈleɪtɛk). Test paper.

**Please note that …*

1. What is LaTeX?

Add text here.

2. Structure of a LaTeX-Document

A LaTeX-document consists of two main parts: 1.) the preamble and 2.) the body. The preamble is where the fundamentals of the document are set up, including the type of document, format and packages (see XXX). It may also contain metadata such as the the author, date or title. The preamble is initiated by the command *\documentclass{}* and ends with the beginning of the main body. The main body is where the actual text of the document is created through a combination of normal text and markup. It always begins with the *\begin{document}* command and ends with the *\end{document}* command. Every LaTeX document requires all three of the above command lines irregardless of the contents. Thus, the general structure of a LaTeX document can be described as:

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

\begin{document}
    ...

\end{document}
```

$\left.\begin{array}{l}\\\\\end{array}\right\}$ *preamble*

$\left.\begin{array}{l}\\\\\end{array}\right\}$ *body*

Ex. 1)

```
\documentclass{minimal}
\begin{document}
Hello world!
\end{document}
```

3. LaTeX-Syntax

In LaTeX special commands are used for typesetting a text or document. These commands are usually a combination of special characters and letters, do not allow

for any spaces and are case sensitive [1] . Similarly to the valency of a verb in language, LaTeX commands can be classified by the number of arguments they take. In general, they can be categorised into two major groups:

1.) Zero-Argument-Commands
2.) Non-Zero-Argument-Commands

3.1 Zero-Argument-Commands

"Zero-Argument-Commands" are commands that do not require any additional arguments - at least on the surface level - and that can be used intransitively (i.e. on their own). These commands generally consist of three components: 1.) a backslash, 2.) a simple word or phrase indicating the function of the command and 3.) (empty) curly brackets. They can be notated as:

\somecommand{}

Ex. 1)

\LaTeX{}	–	displays the "LaTeX" symbol (LaTeX)
\noindent{}	–	suppresses paragraph indentation
\bigskip{}	–	creates a vertical empty space

Since basic commands like this do not take on any specified arguments, the curly brackets (for more detail see 3.2) may be left out. While this normally does not entail any loss of function, it can lead to minor behavioural differences (such as inserting or deleting a space), especially when the command is a direct part of the text:

\LaTeX is cool	vs.	*\LaTeX{} is cool*
(= LaTeXis cool)		(= LaTeX is cool)

3.2 Commands and Arguments

"Non-Zero-Argument-Commands" are commands that typically require an argument, i.e. an object they can be applied to. For example, the *\textit{x}* command that is used for italicising needs an object x (usually a word or text) it can refer to or else it would not execute. This type of commands follows a similar syntax to the one seen in 3.1, with the exception that a.) curly brackets cannot be empty and actively require an argument and b.) facultative arguments (arguments that modify a command but are not required for the command to work) may be passed on via square brackets. Sometimes a command may also require more than one obligatory argument, which can be marked with additional curly brackets:

[1] Case sensitive means that upper- and lowercase are treated as distinct. For example, *command1* and *Command1* would be two different commands and not interchangeable. Case sensitivity is also a common source of error, as commands must be entered as intended (e.g. *\LaTeX{}* vs. ~~*\latex{}*~~).

\somecommand[optional]{obligatory}{obligatory}

Ex. 2)

\textit{Hello world!}	–	puts specified text into italics
\documentclass[paper=a4]{scrartcl}	–	sets class of document (and specifies paper size)
\textcolor{red}{Hello world!}	–	applies specified colour to object

3.3 Spaces and Line Breaks

LaTeX mainly disregards any in-code spaces or line breaks, i.e. they have no visual impact on the actual text document. For example, a sentence can be spread across several code lines without disrupting the text output. In the same way, any number of spaces can be inserted between code elements without them being transferred to the text document. The only time where spacing matters is for commands, as they cannot be broken up. Compare:

Code

```
I'm just
a
simple  example  trying  to
make  my way
through  the universe.
```

Output

I'm just a simple example trying to make my way through the universe.

To include line breaks in a text, one may choose from various options such as:

- <empty line> (creates new line)
- *\newline* (same as <empty line>)
- *\hfill \break* (same as <empty line>)
- *\bigskip* (creates vertical space, ordinarily one line)

Code

```
You don't need to see

my identification. \newline
I'm not the example you're
looking for. \hfill \break
I can go about
my business. \bigskip{}

Move along!
```

Output

You don't need to see
my identification.
I'm not the example you're looking for.
I can go about my business.

Move along!