LaTeXnotes

alepoptosis[*], too much tea

April 2019

1 Font style and size

Here is a series of font styles:

Use `\textbf{}` **for bold!**
Use `\textit{}` *for italics!*
Use `\underline{}` to underline!
Also ***combine them!***
Use `\textsl{}` *for slanted!*
Use `\textsc{}` for small caps!
Use `\emph{}` *emphasize!*

Here are the default font families:

`\textrm{}` makes text Serif (roman)!
`\textsf{}` makes it Sans Serif!
`\texttt{}` `makes it Typewriter (monospace)!`

Here are some font sizes (relative to starting size):

Here, size names are the commands (case sensitive)! e.g. `\tiny{}, \large{}, \Large{} and \huge{}`

This is tiny! This is small! This is large! I said Large! NO, LARGE!!! Now huge! Aaand back to normal size.

[*] find me on twitter!

2 Formatting

Abstract

`\begin{abstract}` formats an abstract environment.

This is a first paragraph.
`\\` creates a line break.
`\newline` also creates a line break.

`\begin{center}` centres the text! You can also align it left (`\flushleft`) or right (`\flushright`), or do nothing to leave it justified.

A blank line creates a new paragraph!

The `\par` command also creates a new paragraph!

You can insert horizontal space in text!
You can set a specific width with `\hspace{width}` (all units allowed.) or you can have a break with `\hfill` that automatically fills all the space available.
You can make this pretty by adding a line with `\hrulefill`
or dots with `\dotfill` .... Useful for signatures and indices!

You can also insert vertical space! Again, you can either use `\vspace{height}` to

specify a distance or use `\vfill` to automatically

fill the space available (will take into account other elements in page).
`\smallskip, \medskip and \bigskip` are also somewhat dynamic ways to insert vertical space.

Columns

Square brackets mark the header text on top of the columns. Put whatever you want here but figures/tables.

This text should end up in column 1.

This text should end up in column 2.

This text should end up in column 3.

And would you look at that - it did. Yay.

Sections hierarchy (commands):

-1. part: only available in report doc class

0. chapter: only available in book doc class

1. section

2. subsection

3. subsubsection

4. paragraph

5. subparagraph

Any of these can be made unnumbered by adding an * before the opening curly brace.

Units accepted by LaTeX:

Abbreviation	Value
pt	∼0.0138 inch or 0.3515 mm
mm	a millimeter
cm	a centimeter
in	an inch
ex	∼height of an 'x' in the current font
em	∼width of an 'M' in the current font
mu	math unit equal to 1/18 em

Table 1: LaTeXunits summary

Paragraph formatting

By default, the first paragraph of a section or a chapter is not indented.

The second is, and the indent size is defined in the preamble.

All subsequent ones are too.

The `\noindent` command changes this.

This is a paragraph that spans multiple lines that I am using to test the line spacing options. Really there is not that much to see here. Carry on.

3 Figures and referencing

Figure 1: me

`\cref{}`, from the `cleveref` package, is used to reference the figure label, Figure 1, while `\cpageref{}` will refer to the page it is in, Page 4. You can also use `\cref{}` to refer to a section, like Section 3.

Change settings for the package in the preamble. You can use the built-in reference system with `\ref{} and \pageref{}`, but it will only produce the number, and not the label.

4 Lists and maths

Environments are sections of the document that present themselves in a different way to the rest. They start with `\begin{}` and end in... well, `\end{}`. You already encountered plenty in the previous sections. Here are some of them in more details.

Lists

- `\itemize` environment
- This is an unordered list
- It uses bullet points
- Text can be of any length

1. `\enumerate` environment
2. This is an ordered list
3. It uses numbers
4. The list number increases with each item

Description environment: adds a description for every item!

This is an item, and this the description.

Numbering: none.

Math

These are ways to write mathematical expressions, inline mode: $E = mc^2$, $E = mc^2$, $E = mc^2$

These are ways to write mathematical expressions, display mode:

$$E = mc^2 \tag{1}$$

$$E = mc^2$$

$$E = mc^2$$

Like everything else, display mode mathematical equations can be referenced with `\cref`, e.g. Equation (1). Unnumbered equations can also be referenced by number but cannot be easily recognised.

Tables

Col1	Col2	Col2
A	23	10000
B	2	66
C	5	58

Table 2: Fancy table

The table can then be referred to as Table 2 on Page 6.

5 Colours

The `xcolour` package allows to colour things. The basic colours it supports are , (white), black, red, green, blue, cyan, magenta, and yellow.

Adding `dvipsnames` to the package in the preamble allows you to name a few more, like Teal Blue or Wild Strawberry (careful with the caps!). You can also change the background colour for the text.

The color command (instead of textcolor) can be used to switch the colour of an entire block of text until it ends - or until the end of the environment. Remember to switch back in that case!

Here is a list of custom colours (defined in the preamble) instead.

- Pink with rgb
- Pink with RGB
- Pink with cmyk
- Gray with gray
- Rubine red at 70% intensity
- A mix of 10% green and 90% orange
- Defined with HTML code

These can be used for any element that takes a colour as parameter, like a line.

6 Table of contents

Contents

Include me please!

Table of contents don't include unnumbered sections. if one needs to be added, then the command `\addcontentsline{toc}{section}{Unnumbered Section}` can be used above that unnumbered section. For example, this unnumbered section is in the ToC.

I want to be left out!

This unnumbered section is not in the ToC.

7 Referencing with BibLaTeX

Cite things like this: Dirac, 1981, Einstein, 1905, D. E. Knuth, 1973, D. Knuth, n.d.! You can change the referencing style in the preamble. Make sure your editor of choice is using Biber (or your backend of choice) as its default bibliography tool!

References

Einstein, A. (1905). Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, *322*(10), 891–921. doi:http://dx.doi.org/10.1002/andp.19053221004

Knuth, D. E. (1973). Fundamental algorithms. (Chap. 1.2). Addison-Wesley.

Dirac, P. A. M. (1981). *The principles of quantum mechanics*. International series of monographs on physics. Clarendon Press.

Knuth, D. (n.d.). Knuth: Computers and typesetting. Retrieved from http://www-cs-faculty.stanford.edu/~uno/abcde.html. ((accessed: 01.09.2016))

8 Input and include

These commands allow to import content from other .tex files into the main documents, which makes things a lot easier when working on a bigger project.

8.1 Input

This subsection comes from a different file, added to the main document with the input command. This is the more basic command of the two:

- It inputs the contents of the file where the command is inserted (no page breaks);
- Ideal for adding long sections of text or separate figures;
- Can be nested;
- Unlikely to have side effects;
- Can be used everywhere, in the document body, preamble, in packages etc.
- Section numbering remains consistent with main document.

8.2 Include

This subsection also comes from a different file, added to the main document with the include command. This is the more complex command of the two:

- It forces a page break (clearpage) before inserting the new content;
- Ideal for for bigger contents, like entire book chapters;
- Cannot be nested;
- It creates a dedicated .aux file, meaning the file could potentially compiled on its own (this one cannot though!) - the `subfile` and `standalone` packages offer a better alternative if you want to do this;
- Can only be used in the document body, not in preamble, packages or restricted/math mode;
- Also here, numbering remains consistent.

The command `\includeonly{filename1,filename2,...}` can be used in the preamble of the main document to only load selected sections. This allows you to only work on one section at a time while also keeping consistent numbering and referencing.

9 Macros

9.1 Custom commands

Macros allow you to create customised commands for a variety of uses. A new command can be created with `\newcommand{\name}[num][default]{definition}`, where:

- **name** is the name of the new command (digits not allowed);
- **num** is the number of argument the command takes (optional);
- **default** is the default parameter to the first argument (optional). If absent, all arguments become required;
- **definition** is the custom command.

You cannot overwrite an existing command with a new one, unless you use `\renewcommand` instead of `\newcommand`, which use the same syntax. If `\providecommand` is used instead, LaTeXwill silently ignore the new command if one with the same name already exists.

If you find yourself doing something often (e.g. write a complex name, a symbol, an equation etc.) making it a custom command is a good idea for two reasons:

1. It will save you time when writing the document;
2. If you change your mind about how you want those words formatted, you can just change the original command instead of updating them manually one by one. This ensures consistency and saves you more time!

Example: say I write ”but Character name had **no idea** what was lurking around the corner.” a lot. Like, a lot. I can define the following command in the preamble: `\newcommand{\omg}[1][John]{{#1} was \textbf{shocked} beyond measure!}`.
Calling it without any parameters would yield:
John was **shocked** beyond measure!
While calling it with a different parameter would yield:
Agatha was **shocked** beyond measure!

9.2 Custom environments

Macros can also be used to create customised environments, like special types of lists or formatted tables. A new environment can be created with `\newenvironment{name}[num][default]{before}{after}`, where:

- **name** is the name of the new environment;
- **num** is the number of argument the environment takes (optional);
- **default** is the default parameter to the first argument (optional). If absent, all arguments become required;
- **before** contains the material that gets processed before the text in the environment (when `\begin{name)` is encountered);
- **after** contains the material that gets processed after the text in the environment (when `\end{name)` is encountered);

You cannot overwrite an existing environment with a new one, unless you use `\renewenvironment` instead, which uses the same syntax.

Like with commands, using a custom environment instead of repeating the same formatting commands multiple times saves you time and ensures consistency.

Example: I want to put my text in a box with a title. To do that, I would have to put this entire block of commands:

```
\begin{center}
I am a title!\\[1ex]
\begin{tabular}{|p{0.9\textwidth}|}
\hline\\
I am text in the box!
\\\\\hline
\end{tabular}
\end{center}
}
```

To obtain this:

I am a title!

I am text in the box!

Instead, if I create a command for it which takes the title of the box as an argument, I only need to use `\begin{boxed}[Title]` to have the same result.

I am also a title!

And I am also text in the box!