

# $\text{\LaTeX}$ : An Introduction (Part 2)

## University Graduate College Training Course

Dr Martin Chorley

School of Computer Science & Informatics, Cardiff University

February 21st, 2014

Notes

---

---

---

---

---

---

---

## Introduction

- ▶ Recap of Beginners  $\text{\LaTeX}$
- ▶ Floating Environments
- ▶ Cross Referencing
- ▶  $\text{\BibTeX}$
- ▶ Defining Custom Environments
- ▶ Defining Custom Commands
- ▶ Presentations
  - ▶ Beamer class
  - ▶ Creating slides

Notes

---

---

---

---

---

---

---

## Schedule

- 09:00 - 09:15** Welcome & Introduction
- 09:15 - 10:30** Basic  $\text{\LaTeX}$ , Exercise 1, Floats, figure environment, Exercise 2
- 10:30 - 11:00** Coffee Break
- 11:00 - 12:30** Referencing with  $\text{\BibTeX}$ , Exercise 3
- 12:30 - 13:30** Lunch
- 13:30 - 15:00** Presentations in  $\text{\LaTeX}$ - beamer
- 15:00 - 15:30** Coffee Break
- 15:30 - 17:00** Preamble, Custom Environments & Commands, Exercises,  $\text{\LaTeX}$  helpdesk
- 17:00** Close

Notes

---

---

---

---

---

---

---

## LaTeX Recap

Hopefully, everyone is happy with these LaTeX concepts:

- ▶ Writing LaTeX files
- ▶ Document Classes & Structure
- ▶ Packages
- ▶ Sections & Chapters
- ▶ Text Formatting
- ▶ Tables
- ▶ Lists
- ▶ Typesetting Maths

### Notes

---

---

---

---

---

---

---

## Recap - Writing LaTeX Files

Creating documents with LaTeX is simple:

1. Write our document as plain text in a '.tex' file, using LaTeX commands to structure and format it
2. Compile our '.tex' file to produce the output

### Notes

---

---

---

---

---

---

---

## Recap - First (basic) LaTeX Example

```
\documentclass{article}
% Preamble goes here
\begin{document}
% Document content goes here
Hello World!
\end{document}
```

Hello World!

### Notes

---

---

---

---

---

---

---

## Recap - Writing $\LaTeX$ – Commands

$\LaTeX$  commands have an effect on the text in the document. Some commands have additional arguments or optional parameters. The general syntax for a  $\LaTeX$  command is:

```
\commandname[opt1, opt2, ...]{arg1}{arg2}...
```

Notes

## Recap - Writing $\LaTeX$ – Commands & Whitespace

Whitespace after  $\LaTeX$  commands will generally be ignored. If you need a space after a command, you can either add an empty parameter to the command, or use a (breaking or non-breaking) space command.

```
\LaTeX commands will ignore whitespace after them.\newline
We can force a space after a \LaTeX{} command using an empty
parameter. \\
Or we can use a space command (texttt{ } or \texttt{~}) after
our \LaTeX\ command.
This way our \LaTeX~commands and text do not flow together!
```

$\LaTeX$  commands will ignore whitespace after them.  
We can force a space after a  $\LaTeX$  command using an empty parameter.  
Or we can use a space command (texttt{ } or \texttt{~}) after our  $\LaTeX$  command.  
This way our  $\LaTeX$  commands and text do not flow together!

Notes

## Recap - Writing $\LaTeX$ – Comments

The ‘%’ character is used to create comments in  $\LaTeX$ . When  $\LaTeX$  is processing your .tex file and it comes across a ‘%’, it ignores the rest of the line.

```
%This is a comment and will not be shown.
Here is some text in our file that will be shown. %but the rest
of the line will not be.
We can even do things like br%
eak words up with comm%
ents if we want to.
```

Here is some text in our file that will be shown. We can even do things like break words up with comments if we want to.

Notes

Recap - Compiling

That's more than you need to create a basic .tex file and create your first document.

To compile your .tex file and create your document, you use a L<sup>A</sup>T<sub>E</sub>X compiler:

- ▶ latex calls the tex compiler and outputs .dvi files
- ▶ pdflatex calls the pdftex compiler and outputs .pdf files

Notes

---

---

---

---

---

---

---

Recap - Compiling

Compiling creates a lot of extra files, including the output of your document. All of these files are recoverable and can be remade by re-compiling, so can be deleted safely.

The only files you always need to keep and should not delete are .tex, .cls, .sty, .bib and .bst.

Notes

---

---

---

---

---

---

---

Recap - Document Structure

Every L<sup>A</sup>T<sub>E</sub>X document must have a certain structure:

```
\documentclass{...}
% Preamble here
\usepackage{...}
\begin{document}
  % Document contents here
  ...
\end{document}
```

The area before \begin{document} is called the *preamble*. It contains commands concerning the setup of the document.

The text of your document is enclosed between the \begin{document} and \end{document}, within the 'document' *environment*.

Notes

---

---

---

---

---

---

---

Recap - Environments

Environments enclose text and cause it to be treated a certain way, similar to commands. They usually have a larger scope than a command though. They begin with `\begin{...}` and end with `\end{...}`

```
\begin{document}
  Here is some text
  \begin{center}
    Here is some centred text
  \end{center}
\end{document}
```

Here is some text

Here is some centred text

Notes

---

---

---

---

---

---

---

Recap - Document Class

The `\documentclass{...}` command tells L<sup>A</sup>T<sub>E</sub>X which type of document we are creating, and how it should be set up and formatted. This command usually comes at the very beginning of the file.

As with many commands it has optional parameters, which will change aspects of the structure, formatting or layout.

```
\documentclass[opt1,opt2,...]{class}
```

Notes

---

---

---

---

---

---

---

Recap - Document Class

L<sup>A</sup>T<sub>E</sub>X comes with many types of document class built in. Some of the most commonly used are:

article	for scientific articles, short reports, papers etc.
IEEEtran	for articles in the IEEE Transactions format.
report	for longer reports containing chapters, small books, theses.
books	for real books
beamer	for writing presentations

Notes

---

---

---

---

---

---

---

## Recap - Document Class Example

So, to make a two-sided article in 12pt font on A4 paper, you can use the command:

```
\documentclass[12pt,a4paper,twoside]{article}
```

Notes

---

---

---

---

---

---

---

## Recap - Top Matter

After we've specified the document class and included any packages we want to use, we can define information about the document in the top matter.

```
\documentclass{article}

\title{Document Title}
\author{Me}
\date{February 2013}

\begin{document}
  \maketitle
\end{document}
```

Notes

---

---

---

---

---

---

---

## Recap - Abstract

Usually, scientific papers and reports will have an abstract, so L<sup>A</sup>T<sub>E</sub>X includes an environment for specifying which part of your document is the abstract. article and report document classes can use the abstract environment.

```
\documentclass{article}

\begin{document}
  \begin{abstract}
    ...
    Abstract goes here
    ...
  \end{abstract}
  \ldots
\end{document}
```

Notes

---

---

---

---

---

---

---

Recap - Sections & Chapters

We often want to break documents into different parts, chapters or sections.

Command	Level
<code>\part{part_title}</code>	-1
<code>\chapter{chapter_title}</code>	0
<code>\section{section_title}</code>	1
<code>\subsection{subsection_title}</code>	2
<code>\subsubsection{subsubsection_title}</code>	3
<code>\paragraph{paragraph_title}</code>	4
<code>\subparagraph{subparagraph_title}</code>	5

Which section commands you can use depends on which document class you are using.

Notes

---

---

---

---

---

---

---

Recap - Packages

Often, the default set of commands available to L<sup>A</sup>T<sub>E</sub>X cannot solve all of our problems alone. To include graphics, use coloured text or other complicated functionality you will need to include extra packages.

These packages will often have extra optional parameters:

```
\usepackage[opt1, opt2, ...]{packagename}
```

So, for example, to use the package allowing us to use coloured text:

```
\usepackage{color}
```

Notes

---

---

---

---

---

---

---

Recap - Packages

We can include multiple packages in the \usepackage command:

```
\usepackage{color,graphicx,geometry}
```

Any packages where we want to set optional parameters need to use their own \usepackage command:

```
\usepackage{color,graphicx}
\usepackage[margin=2cm]{geometry}
```

Notes

---

---

---

---

---

---

---

Basic LaTeX Example - Exercise 1

So, we can put all this together, and create a simple  $\text{\LaTeX}$  document.

Notes

---

---

---

---

---

---

---

Floats

When using a WYSIWYG editor (such as Word), it is common to control *exactly* where pictures or tables are placed in the text. However, many scientific publications allow pictures or tables to go on separate dedicated pages, or at other points in the document in order to not disrupt the flow of the text.  $\text{\LaTeX}$  handles this using *floating environments*.

It can be disconcerting to 'let go' of the control of where items are placed in your document at first, but in general it results in better looking and easier to read documents.

Notes

---

---

---

---

---

---

---

Floating Tables

In order to make a table 'floating' we wrap the tabular environment in a table environment. This makes the table float so that  $\text{\LaTeX}$  can place it in the most appropriate location within the document. It also allows us to add a caption and label to our table.

```
\begin{table}[ position specifier ]
\centering
\begin{tabular}{|l|}
... your table here ...
\end{tabular}
\caption{This is my table}
\label{tab:mytable}
\end{table}
```

... your table here ...
Table 1 : This is my table

Notes

---

---

---

---

---

---

---



Position Specifier

The optional position specifier on a floating environment gives a ‘hint’ to L<sup>A</sup>T<sub>E</sub>X as to where you want to place the table. L<sup>A</sup>T<sub>E</sub>X will try and honour this position, but it is not guaranteed. The options for location specifier are:

Position Specifier	Location
h	here - where the table is declared
t	at the top of the page
b	at the bottom of the page
p	on a special page of floats

Note that h is automatically replaced by ht, as it can cause problems when used alone. You can try and force L<sup>A</sup>T<sub>E</sub>X to use a specific position by adding ! to the specifier.

Notes

---

---

---

---

---

---

---

Cross Referencing

If our tables (and later images) are ‘floating’ around the document, they may end up being in a different location to the text describing them. L<sup>A</sup>T<sub>E</sub>X provides methods for cross-referencing within documents.

\label allows us to label floats and sections:

```
\label{label_name}
```

\ref allows us to refer back to the labelled float or section:

```
\ref{label_name}
```

\page ref allows us to refer to the page the labelled float or section is on:

```
\pageref{label_name}
```

When using any form of referencing we are required to compile our document twice, so that L<sup>A</sup>T<sub>E</sub>X is able to work out where our references should point to within the document.

Notes

---

---

---

---

---

---

---

Cross Referencing

```
\begin{table}[htb]
  \centering
  \begin{tabular}{|l|}
    ... your table here ...
  \end{tabular}
  \caption{This is my table}
  \label{tab:mytable}
\end{table}
Now in my text I can refer to the Table~\ref{tab:mytable}.
```

Notes

---

---

---

---

---

---

---

## Figure environment

The figure environment allows us to 'float' our images, much like the table environment allows us to 'float' our tabular environments.

As a floating environment, it is then possible to label and caption our images.

```
\begin{figure}[placement option]
... figure contents ....
\end{figure}
```

## Notes

## Figure environment

```
\begin{figure}[ht!]
\centering
\includegraphics[width=0.4\textwidth]{img/background}
\caption{I have no idea what this is}
\end{figure}
```

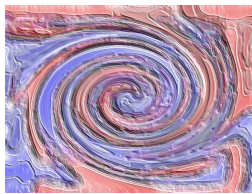


Figure 1 : I have no idea what this is

## Notes

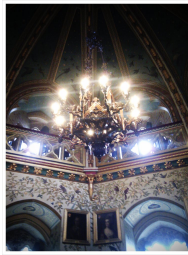
## Subfigures

It is often desired to combine multiple images or figures within a single floating environment. For this we can use the subcaption package.

```
\usepackage{graphicx}
\usepackage{subcaption}
\begin{figure}[htbp]
\centering
\begin{subfigure}{0.3\textwidth}
\includegraphics[width=\textwidth]{img/lights}
\caption{Some lights}
\end{subfigure}
\begin{subfigure}{0.3\textwidth}
\includegraphics[width=\textwidth]{img/bench}
\caption{A bench}
\end{subfigure}
\caption{Some lights and a bench}
\end{figure}
```

## Notes

## Subfigures



(a) Some lights



(b) A bench

Figure 2 : Some lights and a bench

## Notes

---

---

---

---

---

---

---

---

## Subfigures - alignment

We can supply position options to the subfigure environment to align the images within a subfigure

```
\usepackage{graphicx}
\usepackage{subcaption}
\begin{figure}[htbp]
  \centering
  \begin{subfigure}[b]{0.3\textwidth}
    \includegraphics[width=\textwidth]{img/lights}
    \caption{Some lights}
  \end{subfigure}
  \begin{subfigure}[b]{0.3\textwidth}
    \includegraphics[width=\textwidth]{img/bench}
    \caption{A bench}
  \end{subfigure}
  \caption{Some lights and a bench}
\end{figure}
```

## Notes

---

---

---

---

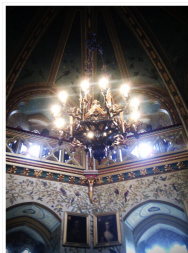
---

---

---

---

## Subfigures



(a) Some lights



(b) A bench

Figure 3 : Some lights and a bench

## Notes

---

---

---

---

---

---

---

---

## Caption Style

The caption package has many options for customising the appearance of captions

```
\usepackage[font=small, labelfont=bf]{caption}
```

### Notes

---

---

---

---

---

---

---

## Double Column Floats

When creating a two-column document, it may sometimes be desirable to have your float placed across both columns.

This can be done using the figure\* and table\* environments, which will place tables or images across both columns of a two-column document.

Note however, this will force the floats to be either at the top of the page, or on a page of their own.

### Notes

---

---

---

---

---

---

---

## Cross Referencing

As with tables, our images ‘float’ around the document and so may end up being in a different location to the text describing them. L<sup>A</sup>T<sub>E</sub>X provides methods for cross-referencing within documents.

\label allows us to label floats:

```
\label{label_name}
```

\ref allows us to refer back to the labelled float or section:

```
\ref{label_name}
```

\pageref allows us to refer to the page the labelled float or section is on:

```
\pageref{label_name}
```

When using any form of referencing we are required to compile our document twice, so that L<sup>A</sup>T<sub>E</sub>X is able to work out where our references should point to within the document.

### Notes

---

---

---

---

---

---

---

## Cross Referencing

Labels must be added *after* the caption, but still within the figure, subfigure or table environment.

```
Figure~\ref{fig:subfigex} has two subfigures: Figure~\ref{lights} is the image with lights, and Figure~\ref{fig:bench} is the image of a bench.
\begin{figure}[htbp]
  \centering
  \begin{subfigure}{0.3\textwidth}
    \includegraphics[width=\textwidth]{img/lights}
    \caption{Some lights} \label{fig:lights}
  \end{subfigure}
  \begin{subfigure}{0.3\textwidth}
    \includegraphics[width=\textwidth]{img/bench}
    \caption{A bench} \label{fig:bench}
  \end{subfigure}
  \caption{Some lights and a bench} \label{fig:subfigex}
\end{figure}
```

Notes

---

---

---

---

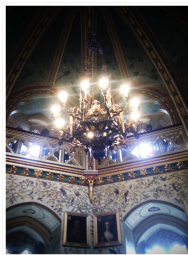
---

---

---

## Cross Referencing

Figure 4 has two subfigures: Figure 4a is the image with lights, and Figure 4b is the image of a bench.



(a) Some lights



(b) A bench

Figure 4 : Some lights and a bench

Notes

---

---

---

---

---

---

---

## Exercise 2

Experiment with adding images into your documents.

Add captions and labels, and refer to them within your text.

Experiment with layout and positioning.

Notes

---

---

---

---

---

---

---

## Help?

There are *many, many* places to get more help with  $\text{\LaTeX}$ .

If you have a problem, use Google! Often that will lead you straight to the documentation for the package or command you have a problem with.

Otherwise, StackExchange has a thriving  $\text{\TeX}$  community where you can ask for help and advice:

<http://tex.stackexchange.com>

## Notes

---

---

---

---

---

---

---

## Help?

All the  $\text{\LaTeX}$  code for the slides and exercises today is available online:

<https://github.com/martinjc/LaTeX-an-Introduction-Part-2->

or

<http://martinjc.com>

## Notes

---

---

---

---

---

---

---

## Notes

---

---

---

---

---

---

---