

Introduction to Latex

Hao Zhong

zhonghao@cdut.edu.cn

Overview

- 1 \LaTeX basics
- 2 Document Structure
- 3 Document Formatting
- 4 Packages
- 5 Tables and Figures
- 6 Mathematics
- 7 Appendix A Customizing \LaTeX
- 8 Appendix B The $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ bundle
-

What is L^AT_EX?

- Pronounced ‘Lah Tek’ or ‘Lay Tek’
- It is a **document markup language** and a **document preparation system**

Document Markup Language

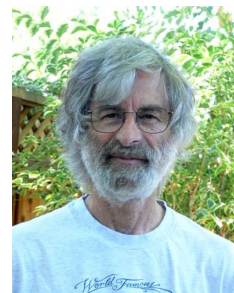
A plain text file, the **source**, contains the text along with **markup commands**, such as ‘title’, ‘section’, ‘figure’ etc.

Document Preparation System

A set of software, such as **latex** and **pdflatex**, processes the source file and produces the output, e.g. pdf

A bit of history

- In 1978, **Donald Knuth** (a well known computer scientist) got frustrated with the mistakes that his publishers made in typesetting his work
- So he decided to create his own typesetting system, one that would be easy to use, especially for typesetting mathematical formulae
- He called his system **T_EX** and made it freely available
- The problem with T_EX was that it was focusing too much on the details
- For this reason, a few years later **Leslie Lamport** created **L^AT_EX**, which focuses on document structure rather than such details



Advantages of using \LaTeX

- Separation of content and style
- Scalability – can be used for a one-page letter or a 300 page book
- High quality output – PDFs produced look elegant and professional
- Performance – \LaTeX doesn't crash like Word
- Can be used with the bibliography/reference package \bibTeX
- Files are very small – a \LaTeX file is just a text file
- It's free
- Platform independent – can be used on Windows, Mac, Linux

Disadvantages of using L^AT_EX

- It can be a bit overwhelming at first
- Learning how to use it takes time
- Complicated workflow – there are several steps to write, generate and produce a finished file
- Just as with any software, there are occasional bugs

Installing L^AT_EX

- ① The L^AT_EX engine (the ‘software’) – required
 - ▶ **MikTeX** for Windows (<http://miktex.org/>)
 - ▶ **MacTeX** for Mac (<http://www.tug.org/mactex/>)
- ② One L^AT_EX editor – optional, but highly recommended!
 - **TeXStudio** for Windows (<https://www.texstudio.org/>)
 - **Texmaker** for Windows (<http://www.xm1math.net/texmaker/>)
 - **Texnic Center** for Windows (<http://www.texniccenter.org/>)
 - **CTeX** for Windows (<http://www.ctex.org/>)

Note

Normally, you don’t fiddle with the L^AT_EX engine on your own (through the command prompt), only via the L^AT_EX editor.

The \LaTeX source file

Example

```
\documentclass{article}

\begin{document}

This is my first document! It
is very simple indeed!

\end{document}
```

The \LaTeX source file:

- is a plain text file
- by convention its file name has extension **.tex**
- contains all the content together with markup commands

\LaTeX commands

Every \LaTeX command starts with a **backslash** (\backslash) followed by the command's name. Many \LaTeX commands are further followed by one or more **arguments**, which go in **curly brackets**.

Document class

Every \LaTeX document starts by declaring what **document class** it belongs to, by using the command `\documentclass`.

There are four built-in classes provided, and many others you can download

- **report** for business, technical, legal, academic, or scientific reports
- **article** for white papers, magazine or journal articles, reviews, conference papers, or research notes
- **book** for books and theses
- **letter** for letters

The most common of these is the article.

The document environment

The actual document goes between the commands `\begin{document}` and `\end{document}`.

- **Above** the `\begin{document}` command (and below the `\documentclass`) we will be able to insert extra setup specifications; this area of the source file is known as the **preamble**
- Anything placed **underneath** the `\end{document}` command is ignored

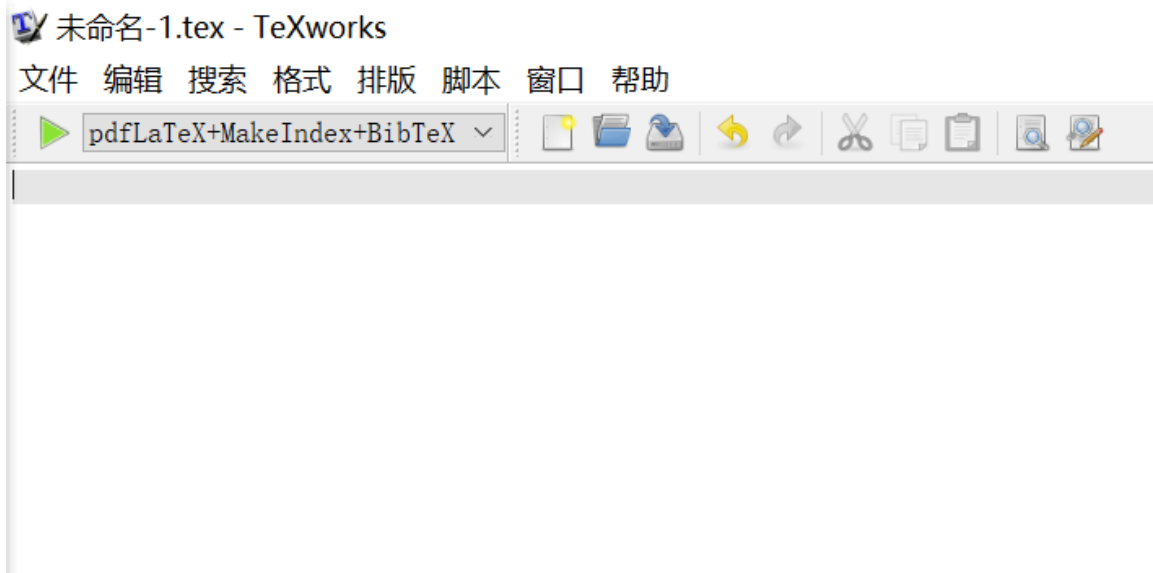
\LaTeX environment

The `\begin... \end` pair of commands is called an **environment**. Environments enclose text which is to be handled in a particular way. All environments start with `\begin{...}` and end with `\end{...}` (putting the name of the environment in the curly braces).

Producing a \LaTeX document

Creating the source file

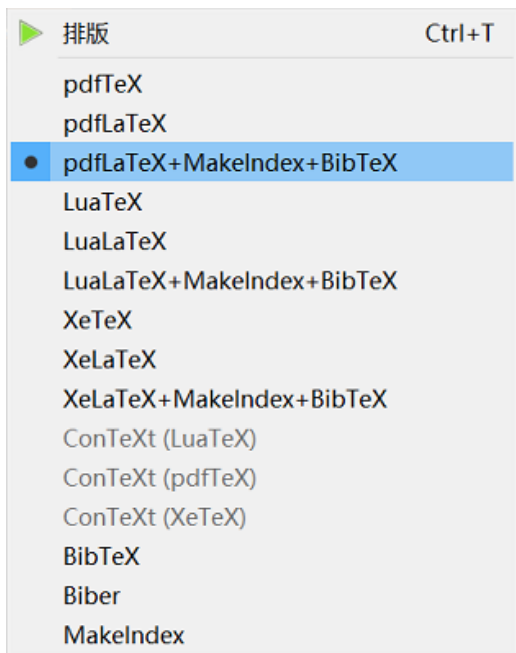
- Open the \LaTeX editor (e.g. TeXworks)
- Click on New to create a new document
- Add your latex code in the file
- Click on Save and select a folder to store your .tex file



Producing a \LaTeX document

Tools and shortcuts

In TeXworks, you can find all the available tools for producing a \LaTeX document.



Note

- -TeX:
- -LaTeX: dvi
- pdf-: pdf
- Lua-: more like a programming language
- Xe-: support a wider range of characters
- MakeIndex: create an Index
- BibTeX: create in-line references

White space in the source file

Example

This is a paragraph
that shows clearly
that two or more spaces
are treated as a single
space

Leaving one more empty
lines creates a new
paragraph. Note: it does
NOT create empty lines in
the document!

- \LaTeX protects you from accidentally adding more than one spaces between words, by treating multiple spaces as a single space.
- Multiple empty lines are interpreted as a single empty line, which itself signifies a **change of paragraph**.

This is a paragraph that shows clearly that two or more spaces are treated as a single space

Leaving one more empty lines creates a new paragraph. Note: it does NOT create empty lines in the document!

Title, Author, Date

The title, author and date provide information *about* the document and for this reason are known as **metadata**.

Example

```
\documentclass{article}

\title{Introduction to LaTeX}
\author{Alice and Bob\\
Chengdu University of
Technology}
\date{}

\begin{document}
\maketitle

Main text goes here.
\end{document}
```

- Metadata can be placed **in the preamble**
- `\maketitle` tells \LaTeX that the metadata is complete and should be typeset; if you omit it, no title will be print
- If `\date` is omitted, today's date will be print
- The command `\\` creates a new line

Abstract

To typeset an abstract in your document, you use the **abstract** environment.

Example

```
\documentclass{article}  
...  
\begin{document}  
\maketitle
```

```
\begin{abstract}
```

Latex is a professional typesetting language and document preparation system. It can be used to create all sorts of documents.

In this article we will introduce the most basic LaTeX commands and will provide practical examples.

```
\end{abstract}
```

Main text goes here.

```
\end{document}
```

Sections

There are **seven levels** of sectioning:

Depth	Division	Command	Notes
-1	Part	<code>\part</code>	Not in letters
0	Chapter	<code>\chapter</code>	Books and reports
1	Section	<code>\section</code>	Not in letters
2	Subsection	<code>\subsection</code>	Not in letters
3	Subsubsection	<code>\subsubsection</code>	Not in letters
4	Titled paragraph	<code>\paragraph</code>	Not in letters
5	Titled subparagraph	<code>\subparagraph</code>	Not in letters

There also exists the `\appendix` command which does not take an argument, but it changes the chapter numbering to letters

Section numbering

- L^AT_EX automatically calculates the correct numbering of all types of sections
- Occasionally, you may want to have a section heading that is not numbered
- You can achieve this with the **starred** versions of the section commands

```
...
\subsection{A numbered subsection
heading}
Text of the first subsection ...
\subsection*{This subsection is not
numbered}
Text of the second subsection ...
```

1.1 A numbered subsection heading

Text of the first subsection ...

This subsection is not numbered

Text of the second subsection ...

Note

The starred versions of section commands are automatically **excluded** from the table of contents (if one is created)

Table of contents

- To add a table of contents add the command `\tableofcontents` in the place where you want the table printed (normally after the abstract)
- You need to run `pdflatex` (or `latex`) **twice** in order to get the correct table of contents; the first time all section headings are recorded and the second time they are printed
- Similarly, there exist commands `\listoffigures` and `\listoftables` which automatically list all your tables and figures

Document class optional arguments

- The built-in document classes provided by L^AT_EX can be customized at will, by using **optional arguments**
- All optional arguments are placed in square brackets ([]) and separated (if we have more than one) by commas; in `\documentclass` (and in most other commands) the optional arguments are placed **before** the compulsory arguments

Paper size

- `a4paper`
- `letterpaper`

Example

```
\documentclass[a4paper]{report}
```

Font size (default: 10pt)

- `11pt`
- `12pt`

Example

```
\documentclass[a4paper,11pt]{report}
```

Document class optional arguments (cont'd)

One- or two-side printing

- `oneside` (this is the default for articles and letters)
- `twoside` (this is the default for books and reports)

Separate titlepage

- `notitlepage` (this is the default for articles and letters)
- `titlepage` (this is the default for books and reports)

Columns

- `onecolumn` (this is the default)
- `twocolumn`

Example

```
\documentclass[a4paper,twoside,titlepage]{article}
```

Page numbering and headings

The command `\pagestyle` controls page numbering and headings; it takes one argument and should be placed in the **preamble**

Page style argument options

- **plain**, prints page numbers and no headings (this is the default)
- **empty**, prints neither page numbers nor headings
- **headings**, prints page numbers and headings from the sections
- **myheadings**, prints page numbers and custom headings

Example

```
\documentclass[11pt]{article}  
\pagestyle{headings}  
\begin{document}...
```

Note

Similarly, the command `\thispagestyle` applies one of these options to a single page; it is placed inside the document

Special characters

The following symbols are reserved characters that have a special meaning, so if you would add them in your source file they will not print:

\$ % ^ & - { } ~ \

If you want to print one of these symbols you have to type

\# \\$ \% \^{} \& \- \{ \} \~{} \textbackslash

Note

- Most of the symbols are simply preceded by a \
- \^ and \~ need also to be followed by curly brackets
- The backslash cannot be preceded by another backslash (\\) as this generates a new line; instead we type \textbackslash

Comments

The special character `%` indicates that whatever follows this symbol up until the end of the line is a **comment** and thus will be ignored by \LaTeX

Example

```
This shows % this is a comment  
that comments don't show  
% this whole line is also a comment  
in the final output
```

This shows that comments don't show in the final output

Emphasizing text

- `\textbf{text}` typesets text in **boldface**
- `\textit{text}` typesets text in *italics*
- `\underline{text}` underlines the text

Note

It is common practise to use `\emph{text}`, instead of `\textit{text}`, whose operation depends on the context

- typesets text in *italic* if its context is normal, and
- typesets text in normal font if its context is italic

Example

```
\emph{Emphasizing when the context is emphasized  
prints in \emph{normal} font}
```

Emphasizing when the context is emphasized prints in normal font

Font size

- In general, it is not recommended to mix font sizes within your text
- If however, you need to do so you may use the following commands

Command	Output
<code>\tiny</code>	<small>sample text</small>
<code>\scriptsize</code>	<small>sample text</small>
<code>\footnotesize</code>	<small>sample text</small>
<code>\small</code>	<small>sample text</small>
<code>\normalsize</code>	sample text
<code>\large</code>	sample text
<code>\Large</code>	sample text
<code>\LARGE</code>	sample text
<code>\huge</code>	sample text
<code>\Huge</code>	sample text

- All of these commands take no arguments; if you want the effect to be applied only to a specific context enclose it in curly brackets, e.g.
Normal text, `{\tiny then some tiny text}`, then normal again

Footnotes

Footnotes are created with the `\footnote` command

Example

This is a paragraph\footnote{This is a footnote.} of normal text.

This is a paragraph¹ of normal text.

¹This is a footnote.

Quotation marks

- You should NOT use the " (double quote symbol) for quotation marks as you would on a typewriter
- In publishing there are special opening and closing quotation marks. In \LaTeX , use two ` (grave accent) for opening quotation marks and two ' (vertical quote) for closing quotation marks
- For single quotes you use just one of each

Example

```
`double quote' and `single  
quote'
```

“double quote” and ‘single quote’

Cross references

- The command `\label{marker}` creates a non-printable label
- The command `\ref{marker}` inserts the number of the section, subsection, figure, table, or theorem after which the corresponding `\label` command was issued
- The command `\pageref{marker}` prints the page number of the page where the `\label` command occurred

Example

...

```
\subsection{A random subsection}
```

This is the text we would like to refer to later, so let's add a label.

```
\label{mytext}
```

...

```
\subsection{Another subsection}
```

As I mentioned in section `\ref{mytext}` on page `\pageref{mytext}` ...

Lists

There are three *environments* for creating lists

- `enumerate` for numbered lists
- `itemize` for bulleted lists
- `description` for description lists

Within a list, each item starts with the `\item` command.

Example

```
\begin{enumerate}  
\item First item on my list  
\item Second item on my list
```

Note that one item can span multiple paragraphs

```
\end{enumerate}
```

1. First item on my list
2. Second item on my list
Note that one item can span multiple paragraphs

Customizing a list's items

The command `\item` takes an optional argument which replaces the item's default bullet or number with another symbol or number

Example

```
\begin{itemize}
\item First item
\item Second item
\item[-] Third item, with dash not
bullet
\end{itemize}
```

- First item
- Second item
- Third item, with dash not bullet

Example

```
\begin{description}
\item[Stupid] things will not become
smart because they are in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
```

Stupid things will not become smart because they are in a list.

Smart things, though, can be presented beautifully in a list.

Paragraph alignment

There are three environments for aligning paragraphs

- **flushleft** for left-aligned paragraphs
- **flushright** for right-aligned paragraphs
- **center** for centered paragraphs

By default, \LaTeX aligns paragraphs both on the left and on the right, by expanding/shrinking spaces between words universally within each line

This is a normal paragraph, not enclosed in any paragraph-alignment environment. It can be clearly seen that the lines are aligned both on the left and on the right, as expected.

On the other hand this paragraph is centered. \LaTeX decides what fits in each line, which it then centers.

Finally, this paragraph is right-aligned, by being enclosed in a **flushright** environment, which we just learned.

Quotes

You can quote text from another source (book, article, etc) by

- The `quote` environment, for single-paragraph quotes
- The `quotation` environment, for multiple-paragraph quotes

Example

A typographical rule of thumb for the line length is:

```
\begin{quote}
```

On average, no line should be longer than 66 characters.

```
\end{quote}
```

This is why `\LaTeX` pages have such large borders by default and also why multicolumn print is used in newspapers.

A typographical rule of thumb for the line length is:

On average, no line should be longer than 66 characters.

This is why `LATEX` pages have such large borders by default and also why multicolumn print is used in newspapers.

Accented characters

- In \LaTeX accented characters are produced by using commands like $\backslash\{o\}$
 - ▶ It starts with a backslash (\backslash)
 - ▶ Followed by a symbol that indicates the accent to be placed on our character (e.g. $\grave{}$)
 - ▶ Followed by the letter on which to place the accent, normally enclosed withing curly bruckets

Command	Output	Command	Output
$\backslash\{'\{o\}$	ò	$\backslash\{'\{o\}$	ó
$\backslash\{"\{o\}$	ö	$\backslash\text{H}\{o\}$	ö
$\backslash\^{\{o\}$	ô	$\backslash\sim\{o\}$	õ
$\backslash\text{v}\{o\}$	ǎ	$\backslash=\{o\}$	ō
$\backslash\text{b}\{o\}$	Ɑ	$\backslash.\{o\}$	ȯ
$\backslash\text{d}\{o\}$	ȯ	$\backslash\text{c}\{o\}$	ȯ
$\backslash\text{r}\{o\}$	ř	$\backslash\text{t}\{oo\}$	ř
$\backslash\text{i}$	ı		

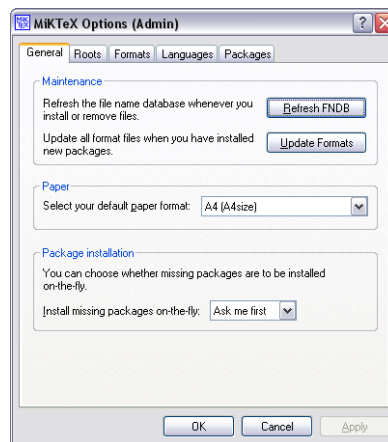
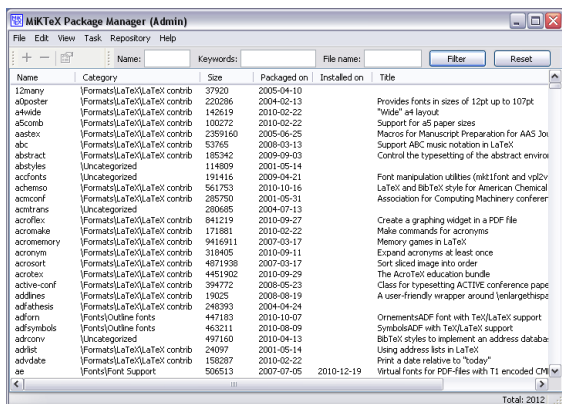
Extending \LaTeX using packages

- \LaTeX provides commands and environment to achieve the most common things
- However, some problems cannot be solved by basic \LaTeX
- In such cases, we can extend the functionality by using **packages**
- A package is a file containing extra \LaTeX commands and programming which add new styling features or modify those already existing
- Package files all end with **.sty**
- You can add a package to a document by calling the following command in the **preamble**: `\usepackage[options]{package}`
 - ▶ **package** is a the name of the package we want to use
 - ▶ **options** is a list of optional parameters to the package

Installing new packages

Many packages come pre-installed with MikTeX. If you require a package not already installed, you may either:

- install the package through MikTeX's **Package Manager**
 - ▶ MikTeX will automatically install it for you if you have this option enabled
- or download the package from the CTAN search page to the directory where your source file is located



Example Packages

Line spacing

- The package `setspace` provides commands for setting the spacing between lines
- It provides three commands:
`\singlespacing` `\onehalfspacing` `\doublespacing`
- The package can also take options (in `\usepackage`) `singlespacing`, `onehalfspacing`, and `doublespacing` to set the spacing for the whole document

Example

```
\documentclass{article}
```

```
\usepackage{setspace}
```

```
\begin{document}
```

This paragraph will be typeset as normal, in single space.

```
\doublespacing
```

All paragraphs from now on will be in doublespace. I can switch back to single or `\end{document}`

Example Packages

Changing the page layout

- the package **geometry** allows you to change the size of the page, the margins, the orientation, etc.
- the package **fancyhdr** provides commands to customize the headers and footers of the document
- the package **multicol** allows you to typeset text in multiple columns (up to 10) and enables you to switch between any (permitted) number of columns at will
- the package **longtable** can be used to create tables than span more than one page
- ...

Note

An enormous number of packages can be found on the **Comprehensive TeX Archive Network** (CTAN)

<http://www.ctan.org>

Creating tables

The environment `tabular` is used to create tables. It is formatted as

`\begin{tabular}{table spec}`

table spec is a list of options that define the format of the table:

- `l` specifies a left-aligned column
- `r` specifies a right-aligned column
- `c` specifies a centered column
- `p{width}` specifies a column of given width, where the text will wrap around if it doesn't fit
- `|` specifies a vertical line

Within a `tabular` environment

- `&` jumps to the next column
- `\\` starts a new line
- `\hline` inserts a horizontal line
- `\cline{i-j}` inserts a horizontal line that spans columns i through j
- `\vline` draws a vertical line with the height of the row

Table example

Example

```
\begin{tabular}{lp{3cm}c|r}  
Item & Description & Quantity & Price \\  
\hline  
Belkin F5D7632 & Belkin ADSL Wireless G Modem Router F5D7632 for BT Line  
& 1 & 9.59\\  
&&&\\  
\hline  
Total & & 9.59  
\end{tabular}
```

Item	Description	Quantity	Price
Belkin F5D7632	Belkin ADSL Wireless G Modem Router F5D7632 for BT Line	1	9.59
Total			9.59

Text spanning multiple columns

The command `\multicolumn{cols}{pos}{text}` can be used inside a table to insert text that spans multiple columns

- `cols` is the number of columns the text will span
- `pos` specifies the alignment within the current multi-column
 - ▶ it can take values: `r`, `l`, `c`, `p{width}`
- `text` is the text to be inserted in the multi-column

Example

```
\begin{tabular}{l|c|r}  
Item & Quantity & Price \\  
\hline  
F5D7632 & 1 & 9.59\\  
\hline  
\multicolumn{2}{c}{Total} & 9.59  
\end{tabular}
```

Item	Quantity	Price
F5D7632	1	9.59
Total		9.59

Table positioning

The tabular environment can also take an optional argument `\begin{tabular}[pos]{table spec}` which specifies the vertical position of the table relative to the baseline of the surrounding text

- **t** aligns the text with the top of the table
- **b** aligns the text with the bottom of the table
- **c** aligns the text with the center of the table

Example

The bill:

```
\begin{tabular}[c]{|l|c|r|}  
\hline  
Item & Quantity & Price \\  
\hline  
Belkin F5D7632 & 1 & 9.59\\  
NetGear WGR614 & 1 & 17.95\\  
\hline  
Total & & 27.54\\  
\hline  
\end{tabular}
```

The bill:

Item	Quantity	Price
Belkin F5D7632	1	9.59
NetGear WGR614	1	17.95
Total		27.54

Create LaTeX tables online

- Tables Generator
- <https://www.tablesgenerator.com/>

The screenshot shows the 'Tables Generator' web application. At the top, there's a header with the logo and 'LaTeX' tab. Below it is a menu bar with 'File', 'Edit', 'Table', 'Column', 'Row', 'Cell', and 'Help'. A toolbar contains icons for table structure, text formatting (bold, italic, underline), and alignment. The main area features a table grid with 4 rows and 5 columns. The first cell is highlighted in yellow. Below the grid is a 'Generate' button. The 'Result' section shows the generated LaTeX code:

```
1 \begin{table}[]
2 \begin{tabular}{lllll}
3 & & & & \\
4 & & & & \\
5 & & & & \\
6 & & & & \\
7 \end{tabular}
8 \end{table}
```

Floating bodies

- Tables and figures normally cannot be broken across pages
- Starting a new page every time a table or figure does not fit in the current page, leaves partially empty pages which is ugly! (see MS Word!)
- \LaTeX floats any table or figure that does not fit on the current page to a later page, while filling the current page with body text
- There are two environments for floating bodies
`\begin{table}[...]` and `\begin{figure}[...]`
- The optional parameter takes one or more (in order of preference) of the following options:
 - ▶ `t` to place the item at the top of the page
 - ▶ `b` to place the item at the bottom of the page
 - ▶ `h` to place the item right here
 - ▶ `p` to place the item on a special “floats” page
-

Floating bodies – Example

Example

```
\begin{table}[htbp]
\begin{center}
\begin{tabular}{||c|l||}
\hline
option & meaning \\
\hline
h & here on the page \\
b & bottom of a page \\
t & top of a page \\
p & floating page \\
\hline
\end{tabular}
\end{center}
\caption{Table-Figure Destinations}
\label{tb:loc}
\end{table}
```

- The `\caption` command defines a caption for the float
- \LaTeX automatically adds the word “**Table**” or “**Figure**” and a number in front of the caption
- Using the commands `\listoffigures` and `\listoftables` \LaTeX automatically generates a list of figures and tables respectively

Including graphics

- Load the **graphicx** package in the preamble:
`\usepackage{graphicx}`
- To include a picture in your document use
`\includegraphics[key=value,...]{filename}`
 - ▶ **filename** is the filename (including the path) of the picture
 - ▶ The optional parameter accepts a comma separated list of keys and associated values. Available keys include:
width, height, angle, scale
- The picture may (optionally) be enclosed in a **figure** environment

Example

```
\begin{figure}  
\centering  
\includegraphics[angle=90, width=0.5\textwidth]{mypicture}  
\caption{This is my figure.}  
\end{figure}
```

Typesetting mathematics

Mathematics can be typeset in two styles:

- in **text style**, where maths is printed inline within a paragraph
- in **display style**, where maths is printed as a separate paragraph

The inline mode is achieved by enclosing the maths in **\$... \$**

Example

Within the current paragraph I will typeset some maths in `\emph{text style}`, as follows:
\$a + b = 5\$; notice that the math occurs inline; notice also that when typesetting maths (in either style) all spaces are being ignored!

Within the current paragraph I will typeset some maths in *text style*, as follows: $a + b = 5$; notice that the math occurs inline; notice also that when typesetting maths (in either style) all spaces are being ignored!

Mathematics in display style

There are two ways to typeset maths in *display style*

- using the commands `\[...\]` (`$$... $$`) where equations are by default **non-numbered**
- using the environment `equation` where equations are by default **numbered**

Example

Typesetting maths
in display style

```
\[ a*b=6 \]
```

is similar to

```
\begin{equation}
```

```
a*b=6 \label{eq:atimesb}
```

```
\end{equation}
```

As shown, in Equation

(`\ref{eq:atimesb}`) above, bla
bla bla

Typesetting maths in display style

$$a * b = 6$$

is similar to

$$a * b = 6 \tag{1}$$

As shown, in Equation (1) above,
bla bla bla

Math mode vs text mode

In math mode:

- More than one spaces and single line breaks are ignored
 - ▶ if you want to add custom spaces use `\quad` for a single space and `\qquad` for a double space
- Empty lines are not allowed
- Letters are considered as variable names and are typeset in italics
 - ▶ if you want to typeset text within math mode use the `\text{rm}{...}` command

Example

```
$x^2 \geq 0 \qquad \text{for all } x \in \mathbf{R}$
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}$$

Greek letters in math mode

- Lowercase greek letters are entered as `\alpha`, `\beta`, `\gamma`, ...
- Uppercase greek letters are entered as `\Alpha`, `\Beta`, `\Gamma`, ...

Example

```
$ \chi + \Psi - \Omega = \xi/2 $
```

$$\chi + \Psi - \Omega = \xi/2$$

Exponents, Superscripts and Subscripts

- Exponents and superscripts are achieved with the `^` symbol
- Subscripts are achieved with the `_` symbol
- `^` and `_` act only on the next character, so if you want a command to affect several characters, you have to group them together using curly braces: `{...}`

Example

```
\[ x^{\alpha+5}_i = \sum_{j=1}^m y_j \]
```

$$x_i^{\alpha+5} = \sum_{j=1}^m y_j$$

More mathematical symbols

- `\cdot` produces a single centered dot (multiplication)
- `\cdots` produces three centered dots
- `\ldots` produces three low dots
- `\sum` produces the sum
- `\prod` produces the product
- `\lim` produces the limit

Example

```
\[ x_1 + x_2 + \ldots +  
x_n = \sum_{i=1}^n x_i \]  
\[ x_1 \cdot x_2 \cdots x_n =  
\prod_{i=1}^n x_i \]
```

$$x_1 + x_2 + \dots + x_n = \sum_{i=1}^n x_i$$

$$x_1 \cdot x_2 \cdots x_n = \prod_{i=1}^n x_i$$

More mathematical symbols

- `\binom` takes two arguments and typesets the binomial
- `\int` produces the integral
- `\sqrt` produces the square root of its argument
 - takes also an optional argument $[n]$ to define the n -th root
- `\frac` takes two arguments and produces their fraction

Example

```
\[ \int_0^{\frac{\pi}{2}} x \mathrm{d}x \\ \qquad \qquad \binom{n}{m} \\ \qquad \qquad \sqrt[3]{125} \]
```

$$\int_0^{\frac{\pi}{2}} x \mathrm{d}x \qquad \binom{n}{m} \qquad \sqrt[3]{125}$$

A list of available mathematical symbols can be found at
[LATEX Mathematical Symbols.pdf](#)

Left and right delimiters

- If you put `\left` in front of an opening delimiter and `\right` in front of a closing delimiter, \LaTeX will automatically determine the correct size of the delimiter
- You must close every `\left` with a corresponding `\right`
 - ▶ If you don't want anything on the right, use the invisible `"\right."`

Example

```
$1 + \left(\right.  
\frac{1}{1-x^2}\right)^3  
\qquad \left\{  
\frac{1}{1-x^2}\right.$
```

$$1 + \left(\frac{1}{1-x^2}\right)^3 \quad \left\{\frac{1}{1-x^2}\right.$$

Arrays and matrices

- To typeset arrays, use the `array` environment. It works similar to the `tabular` environment

Example

```
\[ \mathbf{X} = \left(
\begin{array}{ccc}
x_1 & x_2 & \ldots \\
x_3 & x_4 & \ldots \\
\vdots & \vdots & \ddots
\end{array}
\right) \]
```

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Example

```
\[ |x| = \left\{
\begin{array}{rl}
-x & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
x & \text{if } x > 0.
\end{array}
\right. \]
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

Exercise

Task

- Generate the mathematical formula you see below

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

- Visit

Find out how to typeset the following

$$\mu \left(\bigcap_{j=1}^n O_j \right) \subset \bigcap_{j=1}^n \mu(O_j)$$

*****Also see MathType**

Defining theorem environments

- With the `\newtheorem` command in the `preamble` you define a new environment that looks like a theorem (theorem, lemma, axiom, etc.)

The syntax is: `\newtheorem{name}[counter]{text}[section]`

- ▶ `name` is the name of the environment used in the \LaTeX source file to identify the “theorem” (e.g. `lem` for typesetting Lemmas)
- ▶ `text` is the text that will be printed in front of the “theorem” in the output file (e.g. Lemma)
- ▶ `counter` (optional) is the name of a previously declared “theorem” and makes the new “theorem” to be numbered in the same sequence
- ▶ `section` (optional) specifies the sectional unit within which the “theorem” should get its numbers

Using theorems

- After defining a new “theorem”, we use it like this

```
\begin{name}[title]  
This is my interesting theorem  
\end{name}
```

- ▶ **name** is the name of the environment we created using the `\newtheorem` command
- ▶ **title** (optional) gives a title to the particular theorem

Example

```
\documentclass{article}  
\newtheorem{lem}{Lemma}[section]  
...  
\begin{document}  
...  
\begin{lem}[Manolis' new lemma]  
Typesetting theorems in \LaTeX is easy!  
\end{lem}
```

Lemma 2.1 (Manolis' new lemma)
Typesetting theorems in \LaTeX is easy!

Where to go next

- Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl. *The (Not So) Short Guide to L^AT_EX2e: L^AT_EX2e in 157 Minutes*. 2011, Technical report.
<http://tobi.oetiker.ch/lshort/lshort.pdf>
- Peter Flynn *A beginner's introduction to typesetting with L^AT_EX (Fifth Edition)*. 2011
- Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L^AT_EX Companion, (2nd Edition)*. Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- The Comprehensive L^AT_EX Symbol List
<http://ctan.org/tex-archive/info/symbols/comprehensive>

Appendix A

Customizing L^AT_EX

Paragraph indents and spacing

- Paragraph indentation depends on the document class in use
- Occasionally you may want a paragraph to have no indentation; in such cases you can precede the paragraph with the command `\noindent`
- If you want to change the size of the indent of *all* paragraphs in your document you can change the parameter `\parindent`
- The `\setlength` command can be used in the preamble to specify a custom indentation length
`\setlength{\parindent}{length}`
- Similarly, the space between paragraphs is held in `\parskip`, which could be altered in a similar fashion as above

Customising enumerated lists

- For all objects that \LaTeX automatically numbers (sections, theorems, etc.), there is a **counter** associated with it
- Each counter has a default format that dictates how it is displayed

Command	Example
<code>\arabic</code>	1, 2, 3 ...
<code>\alph</code>	a, b, c ...
<code>\Alph</code>	A, B, C ...
<code>\roman</code>	i, ii, iii ...
<code>\Roman</code>	I, II, III ...

- There are four individual counters that are associated with enumerated lists, representing the four possible levels of nesting: **`enumi`**, **`enumii`**, **`enumiii`**, **`enumiv`**

Customising enumerated lists (cont'd)

- There are two commands associated with a counter, e.g. `\enumi`
 - ▶ `\theenumi` which defines how the counter is used in cross-references, and
 - ▶ `\labelenumi` which defines how the counter is printed in the list
- You can modify the number of a list by issuing:

```
\renewcommand{\theenumi}{\Roman{enumi}}  
\renewcommand{\labelenumi}{\theenumi}
```

Customising itemized lists

- Itemised lists are not as complex as they do not need to count
- Therefore, to customise, you simply change the labels
- The itemize labels for the four respective levels are accessed via:
`\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`

- Example:

```
\renewcommand{\labelitemi}{\textgreater}
```

Customising footnote numbers

- By default, footnotes are numbered in Arabic
- You can change footnote numbering by using the following command
`\renewcommand{\thefootnote}{\alpha{footnote}}`

Appendix B

The $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX bundle

The $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX bundle

- If you want to typeset (advanced) mathematics, you should use $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX ; it is a collection of packages and classes for mathematical typesetting
- $\mathcal{A}\mathcal{M}\mathcal{S}$ stands for the American Mathematical Society
- The most commonly used package of the bundle is `amsmath`, which contains commands for typesetting maths
- Other useful packages of the $\mathcal{A}\mathcal{M}\mathcal{S}$ include:
 - ▶ `amsthm` for theorems
 - ▶ `amssymb` for mathematical symbols
 - ▶ `amsfont` for mathematical fonts

The amsmath package

A sample of commands and environments provided by amsmath:

- The **multline** environment allows for line breaks in equations

```
\begin{multline}
a + b + c + d + e + f
+ g + h + i
\\
= j + k + l + m + n
\end{multline}
```

$$\begin{aligned} a + b + c + d + e + f + g + h + i \\ = j + k + l + m + n \end{aligned} \quad (3.4)$$

- The **align** environment for aligning equations

```
\begin{align}
a &= b + c \\
&= d + e
\end{align}
```

$$\begin{aligned} a &= b + c & (3.7) \\ &= d + e & (3.8) \end{aligned}$$

- The **matrix** environment and its variants for matrices

```
\begin{equation*}
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix}
\quad \quad
\begin{bmatrix}
p_{11} & p_{12} & \dots & p_{1n} \\
p_{21} & p_{22} & \dots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{m1} & p_{m2} & \dots & p_{mn}
\end{bmatrix}
\end{equation*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix}$$

The amsthm package

- As noted earlier, the command `\newtheorem` allows you to define your own theorem environments
- The `amsthm` package provides the `\theoremstyle{style}` command which lets you define what the theorem is all about by picking from three predefined styles:
 - ▶ `definition` fat title, roman body
 - ▶ `plain` fat title, italic body
 - ▶ `remark` italic title, roman body
- Simply, put the right `\theoremstyle` command before defining the new theorem environments with `\newtheorem`

```
\theoremstyle{definition} \newtheorem{law}{Law}
\theoremstyle{plain}      \newtheorem{jury}[law]{Jury}
\theoremstyle{remark}     \newtheorem*{marg}{Margaret}
```

```
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}.\end{jury}
\begin{marg}No, No, No\end{marg}
```

Law 1. Don't hide in the witness box

Jury 2 (The Twelve). *It could be you! So beware and see law 1.*

Margaret. No, No, No

Proofs

- The `amsthm` package also provides the **proof** environment
- It will automatically print the end-of-proof symbol at the end of the proof

```
\begin{proof}  
  Trivial, use  
  \begin{equation*}  
    E=mc^2.  
  \end{equation*}  
\end{proof}
```

Proof. Trivial, use

$$E = mc^2.$$

□

The amssymb package

- The **amssymb** package provides a large number of mathematical symbols that were not available in plain \LaTeX
- For example:

$\dot{+}$	<code>\dotplus</code>	\cdot	<code>\centerdot</code>	
\ltimes	<code>\ltimes</code>	\rtimes	<code>\rtimes</code>	\div <code>\divideontimes</code>
\mathcal{W}	<code>\doublecup</code>	\mathcal{M}	<code>\doublecap</code>	\smallsetminus <code>\smallsetminus</code>
\veebar	<code>\veebar</code>	$\bar{\wedge}$	<code>\barwedge</code>	$\overline{\wedge}$ <code>\doublebarwedge</code>
\boxplus	<code>\boxplus</code>	\boxminus	<code>\boxminus</code>	\ominus <code>\circleddash</code>
\boxtimes	<code>\boxtimes</code>	\boxdot	<code>\boxdot</code>	\odot <code>\circledcirc</code>
\intercal	<code>\intercal</code>	\circledast	<code>\circledast</code>	\times <code>\rightthreetimes</code>
\curlyvee	<code>\curlyvee</code>	\curlywedge	<code>\curlywedge</code>	λ <code>\leftthreetimes</code>

Mathematical fonts

- The packages **amssymb** and **amsfont** provide mathematical fonts and font-formatting commands
- For example:

Example	Command	Required package
$ABCDEabcde1234$	<code>\mathrm{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathit{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathnormal{ABCDE abcde 1234}</code>	
$ABCDE$	<code>\mathcal{ABCDE abcde 1234}</code>	
\mathscr{ABCDE}	<code>\mathscr{ABCDE abcde 1234}</code>	<code>mathrsfs</code>
$\frac{ABCDEabcde}{1234}$	<code>\mathfrak{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>
$ABCDE\mathbb{K}\mathbb{H}\mathbb{Z}$	<code>\mathbb{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>