Introduction to LaTeX from a Practical Perspective

Y. Xiang, Y. Yin, R. Wang, B. Liu

TechJI

May 26, 2023

Wechat Group

Agenda

▶ General Introduction to LaTeX	*by* Y. Xiang
▶ Introduction to `Tikz`	*by* B. Liu
▶ Beamer	*by* Y. Yin

Table of Contents

What is LaTeX

From Wikipedia, the free encyclopedia[1]

LaTeX (lah-tekh, lah-tek or lay-tek, a shortening of Lamport TeX) is a document preparation system. When writing, the writer uses plain text in markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylize text throughout a document (such as **bold** and *italic*), and to add citations[1] and cross-references.

[1] LaTeX - https://en.wikipedia.org/wiki/LaTeX

What is LaTeX

From Wikipedia, the free encyclopedia[1]

LaTeX (lah-tekh, lah-tek or lay-tek, a shortening of Lamport TeX) is a document preparation system. When writing, the writer uses plain text in markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylize text throughout a document (such as **bold** and *italic*), and to add citations[1] and cross-references.

A TeX distribution such as TeXLive or MikTeX is used to produce an output file (such as PDF or DVI) suitable for printing or digital distribution.

[1] LaTeX - https://en.wikipedia.org/wiki/LaTeX

What is LaTeX

From Wikipedia, the free encyclopedia[1]

LaTeX (lah-tekh, lah-tek or lay-tek, a shortening of Lamport TeX) is a document preparation system. When writing, the writer uses plain text in markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylize text throughout a document (such as **bold** and *italic*), and to add citations[1] and cross-references.

A TeX distribution such as TeXLive or MikTeX is used to produce an output file (such as PDF or DVI) suitable for printing or digital distribution.

Within the typesetting system, its name is stylized as LaTeX.

[1] LaTeX - https://en.wikipedia.org/wiki/LaTeX

Comparison with other Typesetting Systems

- Advantages over MS Word

Comparison with other Typesetting Systems

- Advantages over MS Word
    - Plain text file, easy version control

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst
  - Larger community and packages

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst
  - Larger community and packages
- Drawbacks

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst
  - Larger community and packages
- Drawbacks
  - Long compile time

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst
  - Larger community and packages
- Drawbacks
  - Long compile time
  - Too many symbols and commands to remember

Comparison with other Typesetting Systems

- Advantages over MS Word
  - Plain text file, easy version control
  - Wide range of packages and templates
  - Open-source
- Advantages over typst
  - Larger community and packages
- Drawbacks
  - Long compile time
  - Too many symbols and commands to remember
  - Bad, unclear, hard-to-understand error messages

A brief History of TEX and LATEX

Donald Kunuth from Stanford University is the specialist in programming art. In year 1977, he had just received his first samples from the new typesetting system of the publisher's, and its quality was so far below that of the first edition of Volume 2 that he couldn't stand it. Kunuth decided to implement a mathematical composition system by himself (since he is a computer scientist). He figured that this would take about 6 months (Ultimately, it took nearly 10 years). The system is named as TEX, of both the meaning of Greek letters $\tau\epsilon\chi$, and "technical ".

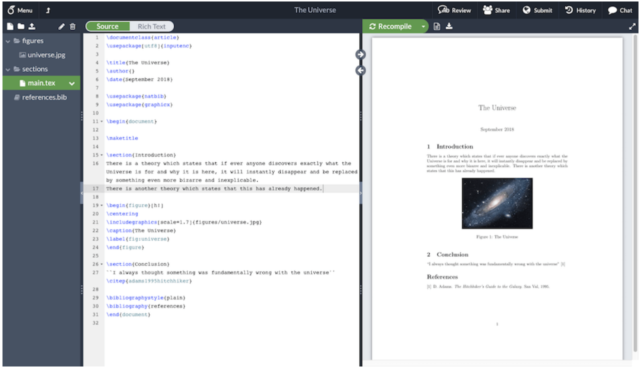
A brief History of TEX and LATEX

Donald Kunuth from Stanford University is the specialist in programming art. In year 1977, he had just received his first samples from the new typesetting system of the publisher's, and its quality was so far below that of the first edition of Volume 2 that he couldn't stand it. Kunuth decided to implement a mathematical composition system by himself (since he is a computer scientist). He figured that this would take about 6 months (Ultimately, it took nearly 10 years). The system is named as TEX, of both the meaning of Greek letters $\tau\epsilon\chi$, and "technical ".

LATEX was created in 1983 by Leslie Lamport, when he was working at SRI. He needed to write TEX macros for his own use, and thought with a little extra effort he could make a general package usable by others. Then LATEX developed rapidly and now there are thousands of packages written in TEX macros available for direct usage.

Write LaTeX on Overleaf (Online)

Another alternative choice is to write LaTeX online with the technology of Overleaf. It's free for personal usage and supports share editing which is very useful in group work. A SJTU hosted Overleaf instance can be found here.



```latex
\documentclass{article}
\usepackage[utf8]{inputenc}

\title{The Universe}
\author{}
\date{September 2018}

\usepackage{natbib}
\usepackage{graphicx}

\begin{document}

\maketitle

\section{Introduction}
There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.
There is another theory which states that this has already happened.

\begin{figure}[h!]
\centering
\includegraphics[scale=1.7]{figures/universe.jpg}
\caption{The Universe}
\label{fig:universe}
\end{figure}

\section{Conclusion}
``I always thought something was fundamentally wrong with the universe'' \citep{adams1995hitchhiker}

\bibliographystyle{plain}
\bibliography{references}
\end{document}
```

The Universe

September 2018

1 Introduction

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states that this has already happened.

Figure 1: The Universe

2 Conclusion

"I always thought something was fundamentally wrong with the universe" [1]

References

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy.* San Val, 1995.

1

Figure: Layout of the Overleaf Online LaTeX Editor.

Installation of LaTeX

Though there are some other distributions of LaTeX(like MikTeX), TeXLive is recommended in this workshop (or you can use Overleaf instead).
We recommend beginners to use VS Code as IDE to edit LaTeX files. See Hydraallen's latex-vscode for installing TeXLive and LaTeX extension for VS Code.

Documentation of LaTeX

If you've installed a full version of TeXLive (as strongly recommended), the full LaTeX documentation is already on your computer.

Documentation of LaTeX

If you've installed a full version of TeXLive (as strongly recommended), the full LaTeX documentation is already on your computer.

Open the command line and input the command

```
    texdoc <docname>
```

Documentation of LaTeX

If you've installed a full version of TeXLive (as strongly recommended), the full LaTeX documentation is already on your computer.

Open the command line and input the command

```
texdoc <docname>
```

You can also use the online version on ▸ Link

Documentation of LaTeX

If you've installed a full version of TeXLive (as strongly recommended), the full LaTeX documentation is already on your computer.

Open the command line and input the command

```
texdoc <docname>
```

You can also use the online version on ▸ Link

For example, you can use the following types for the docname

tex about TeX

article about documentclass `article`

beamer about documentclass `beamer` (used to create slides)

pgf about packages `tikz` and `pgf` (used to draw graphs)

Documentation of LaTeX

If you've installed a full version of TeXLive (as strongly recommended), the full LaTeX documentation is already on your computer.

Open the command line and input the command

```
texdoc <docname>
```

You can also use the online version on ▸ Link

For example, you can use the following types for the docname

tex about TeX

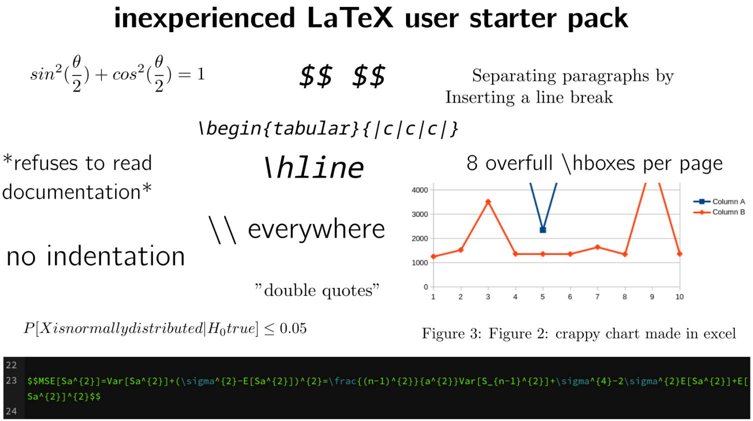
article about documentclass `article`

beamer about documentclass `beamer` (used to create slides)

pgf about packages `tikz` and `pgf` (used to draw graphs)

Try to texdoc about all new things and then you'll be an expert in LaTeX.

Inexperienced LaTeX User Starter Pack



inexperienced LaTeX user starter pack

$sin^2(\frac{\theta}{2}) + cos^2(\frac{\theta}{2}) = 1$

$$ $$

Separating paragraphs by
Inserting a line break

\begin{tabular}{|c|c|c|}

refuses to read documentation

\hline

8 overfull \hboxes per page

\\ everywhere

no indentation

”double quotes”

$P[Xisnormallydistributed|H_0true] \leq 0.05$

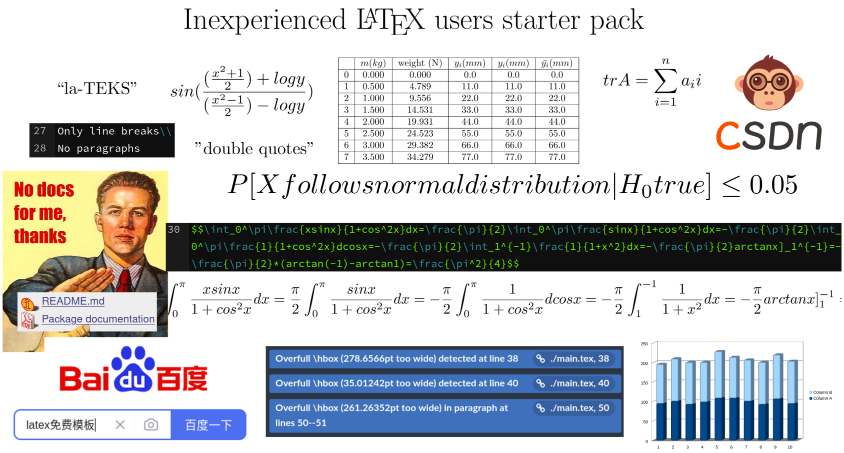
Figure 3: Figure 2: crappy chart made in excel

```

$$MSE[Sa^{2}]=Var[Sa^{2}]+(\sigma^{2}-E[Sa^{2}])^{2}=\frac{(n-1)^{2}}{a^{2}}Var[S_{n-1}^{2}]+\sigma^{4}-2\sigma^{2}E[Sa^{2}]+E[
Sa^{2}]^{2}$$

```

Inexperienced LaTeX User Starter Pack 2.0



Inexperienced LaTeX users starter pack

“la-TEKS”

$$sin(\frac{(\frac{x^2+1}{2})+logy}{(\frac{x^2-1}{2})-logy})$$

```
Only line breaks\\
No paragraphs
```

”double quotes”

	$m(kg)$	weight (N)	$y_i(mm)$	$y_i(mm)$	$\bar{y_i}(mm)$
0	0.000	0.000	0.0	0.0	0.0
1	0.500	4.789	11.0	11.0	11.0
2	1.000	9.556	22.0	22.0	22.0
3	1.500	14.531	33.0	33.0	33.0
4	2.000	19.931	44.0	44.0	44.0
5	2.500	24.523	55.0	55.0	55.0
6	3.000	29.382	66.0	66.0	66.0
7	3.500	34.279	77.0	77.0	77.0

$$trA=\sum_{i=1}^{n}a_ii$$

$$P[X followsnormaldistribution|H_0 true]\leq 0.05$$

```
$$\int_0^\pi\frac{xsinx}{1+cos^2x}dx=\frac{\pi}{2}\int_0^\pi\frac{sinx}{1+cos^2x}dx=-\frac{\pi}{2}\int_
0^\pi\frac{1}{1+cos^2x}dcosx=-\frac{\pi}{2}\int_1^{-1}\frac{1}{1+x^2}dx=-\frac{\pi}{2}arctanx]_1^{-1}=-
\frac{\pi}{2}*(arctan(-1)-arctan1)=\frac{\pi^2}{4}$$
```

$$\int_0^\pi\frac{xsinx}{1+cos^2x}dx=\frac{\pi}{2}\int_0^\pi\frac{sinx}{1+cos^2x}dx=-\frac{\pi}{2}\int_0^\pi\frac{1}{1+cos^2x}dcosx=-\frac{\pi}{2}\int_1^{-1}\frac{1}{1+x^2}dx=-\frac{\pi}{2}arctanx]_1^{-1}$$

Overfull \hbox (278.6566pt too wide) detected at line 38 ./main.tex, 38

Overfull \hbox (35.01242pt too wide) detected at line 40 ./main.tex, 40

Overfull \hbox (261.26352pt too wide) in paragraph at lines 50--51 ./main.tex, 50

Table of Contents

1. Introduction to LaTeX
   - What is LaTeX
   - Distributions and IDEs
   - Documentation
2. Basics
   - Global Structure
   - Modular Documents
3. Text
   - Special Characters
   - Fonts
   - Underline
   - Enumeration
   - Alignment
   - Spaces, lines and pages
   - Minipage and Multicolumn
4. Use Maths in LaTeX
   - Math Expressions
   - Math Environments
   - Spacing in Math Mode
   - Basic Math Commands
   - Matrices and Arrays
5. Useful Maths Packages
   - Common Packages
   - The `systeme` Package
6. Graphs
   - Include Graphs
   - Figures
   - Draw Graphs
7. Tables
   - Tabulars
   - Tables

Global Structure

While in many cases you don't write LaTeXdocuments from scratch (eg. using templates), you should still know the basic structure of a LaTeXdocument (can help you fix bugs!).
Every input file must contain the commands:

Command

```
\documentclass{...}

\begin{document}
...
\end{document}
```

The area between `\documentclass{...}` and `\begin{document}` is called the *preamble*. It normally contains commands that affect the entire document.
You would put your text where the dots are between `\begin{document}` and `\end{document}`.

Document Class

You need to set the layout standard for your document. This is done by specifying a *document class*.

Command

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

Document Class

You need to set the layout standard for your document. This is done by specifying a *document class*.

Command

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

For example, you can use `\documentclass[11pt,twoside,a4paper]{article}` to initialize an article with a base font size of 11 points, and to produce a layout suitable for double sided printing on A4 paper.

Document Class

You need to set the layout standard for your document. This is done by specifying a *document class*.

Command

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

For example, you can use `\documentclass[11pt,twoside,a4paper]{article}` to initialize an article with a base font size of 11 points, and to produce a layout suitable for double sided printing on A4 paper.
You can see a list of document classes in CTAN Class and options in Document Structure.

Packages

In the preamble area, you can load extra packages.

Command

```
\usepackage[options]{package}
% or
\usepackage{package1,package2}
```

Packages

In the preamble area, you can load extra packages.

Command

```
\usepackage[options]{package}
% or
\usepackage{package1,package2}
```

For example, if you want to have more detailed settings for the layout of your document, you can use the *geometry* package.

Packages

In the preamble area, you can load extra packages.

Command

```
\usepackage[options]{package}
% or
\usepackage{package1,package2}
```

For example, if you want to have more detailed settings for the layout of your document, you can use the *geometry* package.
A simple example would be:

Command

```
\usepackage[a4paper,left=2.5cm,right=2.5cm,top=2cm,bottom=2cm]{geometry}
```

This would set a new layout for A4 paper with 2.5cm margins on the left and right, and 2cm margins on the top and bottom of each page.

Packages

In the preamble area, you can load extra packages.

Command

```
\usepackage[options]{package}
% or
\usepackage{package1,package2}
```

For example, if you want to have more detailed settings for the layout of your document, you can use the *geometry* package.
A simple example would be:

Command

```
\usepackage[a4paper,left=2.5cm,right=2.5cm,top=2cm,bottom=2cm]{geometry}
```

This would set a new layout for A4 paper with 2.5cm margins on the left and right, and 2cm margins on the top and bottom of each page.
See Page size and margins for more details on page layout.

Top Matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc.

All of this type of information within LaTeXis collectively referred to as *top matter*.

Top Matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc.

All of this type of information within LaTeXis collectively referred to as *top matter*.

Command

```
...
\begin{document}
\title{LaTeX Document Sample}
\author{Your name}
\date{\today}
\maketitle
\tableofcontents
\end{document}
```

Top Matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc.
All of this type of information within LaTeXis collectively referred to as *top matter*.

Command

```
...
\begin{document}
\title{LaTeX Document Sample}
\author{Your name}
\date{\today}
\maketitle
\tableofcontents
\end{document}
```

You need to use `\maketitle` to generate the final title. If you omit the `\date` command, LaTeXwill use today's date based on the typographic rules.

Top Matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc.
All of this type of information within LaTeXis collectively referred to as *top matter*.

Command

```
...
\begin{document}
\title{LaTeX Document Sample}
\author{Your name}
\date{\today}
\maketitle
\tableofcontents
\end{document}
```

You need to use `\maketitle` to generate the final title. If you omit the `\date` command, LaTeXwill use today's date based on the typographic rules.
`\tableofcontents` will generate a table of contents based on the sections of your document.

Document body

In order for the table of contents to display something it is necessary to add different levels of headings.

Command

```
...
\begin{document}
...

\section{sec1}
\subsection{sec1.1}
\subsubsection{sec1.1.1}

\section{sec2}
\subsection{sec2.1}
\subsubsection{sec2.1.1}
\end{document}
```

Document body

In order for the table of contents to display something it is necessary to add different levels of headings.

Command

```
...
\begin{document}
...

\section{sec1}
\subsection{sec1.1}
\subsubsection{sec1.1.1}

\section{sec2}
\subsection{sec2.1}
\subsubsection{sec2.1.1}
\end{document}
```

To hide the number, add * like `\section*{sec3}`, but be aware that it will not be shown in the table of contents.

Multiple files

LaTeXallows you to split the content of your document into separate files. This is particularly useful if you want to create a book or a large report.

Multiple files

LATEXallows you to split the content of your document into separate files. This is particularly useful if you want to create a book or a large report.

Command

```
\documentclass{article}
\begin{document}
\input{titlepage}
\input{introduction}
\input{section1}
\input{section2}
\end{document}
```

This file will input the contents of the files *titlepage.tex*, *introduction.tex*, *section1.tex* and *section2.tex* in that order. What it does is effectively copy and paste the contents of the files into the document in the order that they are listed.

Graphics Path

When you want to include a figure from file, you normally write `\includegraphics{figures/filename}`. However, if you have a lot of figures, you may want to put them in a separate folder. In this case, you can use `\graphicspath` to specify the path to the folder.

Graphics Path

When you want to include a figure from file, you normally write `\includegraphics{figures/filename}`. However, if you have a lot of figures, you may want to put them in a separate folder. In this case, you can use `\graphicspath` to specify the path to the folder.
Write the following commands in preamble:

Command

```
\graphicspath{{figures/}}
% or many folders
\graphicspath{{subdir1/}{subdir2/}{subdir3/}...{subdirn/}}
```

After that, you can simply use `\includegraphics{filename}` to include the figure.

Table of Contents

1. Introduction to LaTeX
   - What is LaTeX
   - Distributions and IDEs
   - Documentation
2. Basics
   - Global Structure
   - Modular Documents
3. Text
   - Special Characters
   - Fonts
   - Underline
   - Enumeration
   - Alignment
   - Spaces, lines and pages
   - Minipage and Multicolumn
4. Use Maths in LaTeX
   - Math Expressions
   - Math Environments
   - Spacing in Math Mode
   - Basic Math Commands
   - Matrices and Arrays
5. Useful Maths Packages
   - Common Packages
   - The `systeme` Package
6. Graphs
   - Include Graphs
   - Figures
   - Draw Graphs
7. Tables
   - Tabulars
   - Tables

Special Characters

Some special symbols can't be directly used since they are reserved by LATEX:

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

Special Characters

Some special symbols can't be directly used since they are reserved by LaTeX:

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

Many LaTeX starters are confused with how to correctly print quotes, hyphens and dots.

Special Characters

Some special symbols can't be directly used since they are reserved by LaTeX:

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

Many LaTeX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ' prints a right single quote.

Special Characters

Some special symbols can’t be directly used since they are reserved by LATEX:

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

Many LATEX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ’ prints a right single quote.
`` prints a left double quote, ” prints a right double quote.

Special Characters

Some special symbols can’t be directly used since they are reserved by LaTeX:

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

Many LaTeX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ’ prints a right single quote.
`` prints a left double quote, ” prints a right double quote.
one hyphen (-) print like -

Special Characters

Some special symbols can’t be directly used since they are reserved by LATEX:

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

Many LATEX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ’ prints a right single quote.
`` prints a left double quote, ” prints a right double quote.
one hyphen (-) print like -
two hyphens (--) print like –

Special Characters

Some special symbols can’t be directly used since they are reserved by LaTeX:

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

Many LaTeX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ’ prints a right single quote.
`` prints a left double quote, ” prints a right double quote.
one hyphen (-) print like -
two hyphens (--) print like –
three hyphens (---) print like —

Special Characters

Some special symbols can't be directly used since they are reserved by LATEX:

\#		#									
\#	#	\$	$	\%	%	\&	&	\˜	˜	\`	`
\{	{	\}	}	_	_	\backslash	\				

Many LATEX starters are confused with how to correctly print quotes, hyphens and dots.
` prints a left single quote, ’ prints a right single quote.
`` prints a left double quote, ” prints a right double quote.
one hyphen (-) print like -
two hyphens (--) print like –
three hyphens (---) print like —
\dots prints the dots with a correct format (. . .) instead of directly use three dots (...)

Warning

Do NOT use " and ’ to print quotes.

Deal with unfamiliar symbols

Sometimes you may want to deal with symbols you have never seen. In this case, you may refer to http://detexify.kirelabs.org/classify.html to find out how to output the character.

Basic commands about fonts

First, lets start with some commands that transform font types

Basic commands about fonts

First, lets start with some commands that transform font types

- \bf - **Sample Text**
- \it - *Sample Text*
- \rm - Sample Text
- \sc - Sample Text
- \sf - Sample Text
- \sl - *Sample Text*
- \tt - Sample Text

Basic commands about fonts

First, lets start with some commands that transform font types

- \bf - **Sample Text**
- \it - *Sample Text*
- \rm - Sample Text
- \sc - SAMPLE TEXT
- \sf - Sample Text
- \sl - *Sample Text*
- \tt - `Sample Text`

Note that the commands that transform font types influence the text in the whole scope ({...}) until another font type is specified. For example, how to use the first command \bf is shown below

{\bf Sample Text}

Sometimes we don't want to transform all the font types, instead, we can only change the font type of some specified text.

Sometimes we don't want to transform all the font types, instead, we can only change the font type of some specified text.

Example

```
\textbf{Sample text}
```

Sometimes we don't want to transform all the font types, instead, we can only change the font type of some specified text.

Example

```
\textbf{Sample text}
```

There are more options for fonts.

- \textit - *Sample Text*
- \textsc - Sample Text
- \texttt - `Sample Text`

Sometimes we don't want to transform all the font types, instead, we can only change the font type of some specified text.

Example

```
\textbf{Sample text}
```

There are more options for fonts.

- \textit - *Sample Text*
- \textsc - SAMPLE TEXT
- \texttt - `Sample Text`

However, in a math environment (will be introduced later), some other commands should be used

- \mathbf - **Sample Text**
- \mathsf - Sample Text

Sometimes we don't want to transform all the font types, instead, we can only change the font type of some specified text.

Example

```
\textbf{Sample text}
```

There are more options for fonts.

- \textit - *Sample Text*
- \textsc - Sample Text
- \texttt - `Sample Text`

However, in a math environment (will be introduced later), some other commands should be used

- \mathbf - **Sample Text**
- \mathsf - Sample Text

Note that the math environment doesn't include all of the font types on the previous page. More information about font types can be found here.

Font size can also be easily modified

- \tiny - Sample Text
- \scriptsize - Sample Text
- \footnotesize - Sample Text
- \small - Sample Text
- \normalsize - Sample Text
- \large - Sample Text
- \Large - Sample Text
- \LARGE - Sample Text
- \huge - Sample Text
- \Huge - Sample Text

Build a colorful document

Changing the color is similar to changing font types.

If you want to transform to a color (like transforming to bold with `\bf`), you can use `\color{name}`.

Similarly, you can use `\textcolor{name}` like `\textbf`.

The background color of the whole page can be set using `\pagecolor{name}`.

Build a colorful document

Changing the color is similar to changing font types.

If you want to transform to a color (like transforming to bold with `\bf`), you can use `\color{name}`.

Similarly, you can use `\textcolor{name}` like `\textbf`.

The background color of the whole page can be set using `\pagecolor{name}`.

There are some defined color name in the xcolor package.

■ black	■ gray	■ olive	■ teal	■ blue
■ green	■ orange	■ violet	■ brown	■ lightgray
■ pink	□ white	■ cyan	■ lime	■ purple
■ yellow	■ darkgray	■ magenta	■ red	

You can find more information in the documentation of xcolor (texdoc xcolor)

Ulem package

If you want to add some lines on the text, use the ulem package.

Command

```latex
\usepackage{ulem}
\uline{Sample Text}
```

Ulem package

If you want to add some lines on the text, use the ulem package.

Command

```
\usepackage{ulem}
\uline{Sample Text}
```

There are different kinds of lines supported:

- \uline - Sample Text
- \uuline - Sample Text
- \uwave - Sample Text
- \sout - ~~Sample Text~~
- \xout - ~~Sample Text~~
- \dashuline - Sample Text
- \dotuline - Sample Text

Enumerate

When you need to enumerate some items as a list, you may use the enumerate package.

Command

```
\usepackage{enumerate}
\begin{enumerate}[style]
\item % ...
\item % ...
\item % ...
\end{enumerate}
```

Enumerate

When you need to enumerate some items as a list, you may use the enumerate package.

Command

```
\usepackage{enumerate}
\begin{enumerate}[style]
\item % ...
\item % ...
\item % ...
\end{enumerate}
```

This will generate a normal list with the serial numbers in the specified style, which could be the following (as example)

- 1 - 1, 2, 3, 4, ...
- (i) - (i), (ii), (iii), (iv), ...
- [1.] - [1.], [2.], [3.], [4.], ...

Itemize

If you want to generate an unordered list, use itemize instead of enumerate.

Command

```
\begin{itemize}
\item[style] % ...
\item[style] % ...
\item[style] % ...
\end{itemize}
```

Itemize

If you want to generate an unordered list, use itemize instead of enumerate.

Command

```
\begin{itemize}
\item[style] % ...
\item[style] % ...
\item[style] % ...
\end{itemize}
```

In this case, style must be added after each item, which is different from that in enumerate, and the symbol displayed in the beginning of each item will be exactly same as the style. If style is not added, a default style will be used.

Alignment

If you want to align a paragraph of text, use these three environments for left/center/right align.

Command

```
\begin{flushleft/center/flushright}
% ...
\end{flushleft/center/flushright}
```

Alignment

If you want to align a paragraph of text, use these three environments for left/center/right align.

Command

```
\begin{flushleft/center/flushright}
% ...
\end{flushleft/center/flushright}
```

However, if only a single line needs to be aligned, use these three commands.

Command

```
\leftline{text}
\centerline{text}
\rightline{text}
```

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).
- \, - makes a 1/6 em space, commonly used before units (notice the space before em on this page)

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).
- \, - makes a 1/6 em space, commonly used before units (notice the space before em on this page)
- \; - makes a 2/7 em space

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).
- \, - makes a 1/6 em space, commonly used before units (notice the space before em on this page)
- \; - makes a 2/7 em space
- \quad - makes a 1 em space

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).
- \, - makes a 1/6 em space, commonly used before units (notice the space before em on this page)
- \; - makes a 2/7 em space
- \quad - makes a 1 em space
- \qquad - makes a 2 em space

Spaces may be confusing

There are defined command of spaces in different width and usages.

- ␣ - the basic space in LaTeXṄote that any number of spaces or tabs is equal to one space, and the space after a command is ignored. If you want to add an extra space, use \␣ which makes a 1/3 em space (1 em is approximately the width of an M in the current font)
- ˜ - If two words can't be separated on two lines, you can tell LaTeX about it using a tie (˜), such as Prof.˜Hamade (Prof. Hamade).
- \, - makes a 1/6 em space, commonly used before units (notice the space before em on this page)
- \; - makes a 2/7 em space
- \quad - makes a 1 em space
- \qquad - makes a 2 em space
- - makes actually the space of text, but text will be invisible.

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

- \newline - begin a new line

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

- \newline - begin a new line
- \\ - begin a new line (not recommended[1])

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

- \newline - begin a new line
- \\ - begin a new line (not recommended[1])
- \par - begin a new paragraph (a new line with indent)

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

- \newline - begin a new line
- \\ - begin a new line (not recommended[1])
- \par - begin a new paragraph (a new line with indent)
- \newpage - begin a new page

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Separate contents into lines and pages

Here are some basic commands about lines and pages in LaTeX, you will use them everywhere.

- \newline - begin a new line
- \\ - begin a new line (not recommended[1])
- \par - begin a new paragraph (a new line with indent)
- \newpage - begin a new page
- % - begin a line comment

[1] According to Manuel Charlemagne, \\ should only be used for a force break (where \newline doesn't work).

Warning

Never use \\ to create a new line in the text, it will lead to unexpected side effects.

Spacing

When trying to separate two paragraphs by a certain space, many new learners of LaTeX may use multiple empty lines and linebreaks, which is a very dirty fix and is not so accurate. Actually, LaTeX provides a precise spacing mechanism.

Command

```
\vspace{space}
```

```
\vspace*{space}
```

Spacing

When trying to separate two paragraphs by a certain space, many new learners of LaTeX may use multiple empty lines and linebreaks, which is a very dirty fix and is not so accurate. Actually, LaTeX provides a precise spacing mechanism.

Command

```
\vspace{space}
\vspace*{space}
```

When trying to show the next paragraph or sentence precisely at the bottom of the current page, we can use

Command

```
\vfill
```

between the contents of two paragraphs to separate them.

Predefined skipping

More often[1], we don't need to think about the skipping space, we can use the predefined skipping commands to achieve a small, medium or big skip. They are actually particular cases of `\vspace`

Command

```
\smallskip

\medskip

\bigskip
```

[1] According to Manuel Charlemagne, you should always use these skipping commands if possible instead of using \\ (as in many online tutorials).

Predefined skipping

More often[1], we don't need to think about the skipping space, we can use the predefined skipping commands to achieve a small, medium or big skip. They are actually particular cases of `\vspace`

Command

```
\smallskip

\medskip

\bigskip
```

You may note that the effects are these skipping commands have been already shown above.

[1] According to Manuel Charlemagne, you should always use these skipping commands if possible instead of using \\ (as in many online tutorials).

Spacing units

The space can be anything representing a size, such as 1cm, 2em and 10pt. In LaTeX, spacing units can be

- cm
- mm
- in - inch, 1 inch = 2.54 cm
- pt - 72 pt = 1 inch, the smallest unit in LaTeX
- em - 1em equals to the width of letter M
- ex - 1ex equals to the width of letter x
- `\linewidth` - the width of current line in the container
- `\pagewidth` - the width of the page
- `\pageheight` - the height of the page
- `\textwidth` - the normal width of text on the page
- `\textheight` - the normal height of text on the page

Minipage

minipage is a very useful environment for dividing pages into a grid.

Example

```
\begin{minipage}{0.32\linewidth}
  Part A
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  Part B
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  Part C
\end{minipage}
\vfill % Fill vertical space
```

```
\begin{minipage}{0.32\linewidth}
  Part D
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  Part E
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  Part F
\end{minipage}
```

The code above generate six minipages in a grid of 3 columns × 2 rows. Don't try to add up the width of minipages in a line for more than about `0.98\linewidth` (since a minipage have a small margin on each side), or the last minipage may be on a new line.

For each minipage, it can be seem as an independent LaTeX document, where text, formulas, graphics, tables and etc. can be inserted, and most importantly, they won't affect each other. What's more, you can even use minipages in a minipage to form a multi-level nesting.

Example

Part A

Part B

Part C

Part D

Part E

Part F

The multicol package

When typesetting contents with small line width and many lines (for example, source code), the multicol package is recommended.

Command

```
\usepackage{multicol}
\begin{multicols}{cols}
    % contents on column one
    \breakcolumn % break the current column here
    % contents on column two
\end{multicols}
```

Here cols is the number of columns, it must be specified. If `\breakcolumn` is not used, the multicol package will automatically balance the length of each column.

Table of Contents

Introduction

Basic equations in LaTeX can be easily “programmed”, for example:

Example

The well known Pythagorean theorem $x^2 + y^2 = z^2$ was proved to be invalid for other exponents. Meaning the next equation has no integer solutions:

$$x^n + y^n = z^n$$

Subscripts and Superscripts

The use of superscripts and subscripts is very common in mathematical expressions involving exponents, indexes, and in some special operators. [1]

Example

```
\[ a_1^2 + a_2^2 = a_3^2 \]
```

$$a_1^2 + a_2^2 = a_3^2$$

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

Subscripts and Superscripts

The use of superscripts and subscripts is very common in mathematical expressions involving exponents, indexes, and in some special operators. [1]

Example

```
\[ a_1^2 + a_2^2 = a_3^2 \]
```

$$a_1^2 + a_2^2 = a_3^2$$

Note that here we use `\[` and `\]` to typeset a mathematical expression. You may see many people using a pair of `$$` instead. It is a plain-TEX command, and is nowadays heavily deprecated. See this discussion ▸ Link on Stack Exchange for more information.

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

If the expression contains long superscripts or subscripts, these need to be collected in braces, as LaTeXnormally applies the mathematical commands `^` and `_` only to the following character:

If the expression contains long superscripts or subscripts, these need to be collected in braces, as LaTeXnormally applies the mathematical commands ^ and _ only to the following character:

Example

```
\[ x^{2 \alpha} - 1 = y_{ij} +
↪  y_{ij}  \]
\[ (a^n)^{r+s} = a^{nr+ns}  \]
\[ x^abc, \quad x_abc, \quad
↪  x^abc_abc \]
\[ x^{abc}, \quad x_{abc}, \quad
↪  x^{abc}_{abc} \]
```

$$x^{2\alpha} - 1 = y_{ij} + y_{ij}$$

$$(a^n)^{r+s} = a^{nr+ns}$$

$$x^abc, \quad x_abc, \quad x^abc_abc$$

$$x^{abc}, \quad x_{abc}, \quad x^{abc}_{abc}$$

Brackets and Parentheses

Parentheses and brackets are very common in mathematical formulas. You can easily control the size and style of brackets in LaTeX. [1]

Here's how to type some common math braces and parentheses in LaTeX:

Type	LaTeX	Code
Parentheses; round brackets	$(x+y)$	`(x+y)`
Brackets; square brackets	$[x+y]$	`[x+y]`
Braces; curly brackets	$\{x+y\}$	`\{x+y\}`
Angle brackets	$\langle x+y\rangle$	`\langle x+y \rangle`
Pipes; vertical bars	$\lvert x+y\rvert$	x+y——
Double pipes	$\lVert x+y\rVert$	`\\|x+y\\|`
Floor brackets	$\lfloor x+y\rfloor$	`\lfloor x+y \rfloor`
Ceil brackets	$\lceil x+y\rceil$	`\lceil x+y \rceil`

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

The size of brackets and parentheses can be manually set, or they can be resized dynamically in your document, as shown in the next example:

Example

```latex
\[ F = G \left( \frac{m_1 m_2}{r^2}
↪  \right) \]
```

$$F = G \left(\frac{m_1 m_2}{r^2} \right)$$

The size of brackets and parentheses can be manually set, or they can be resized dynamically in your document, as shown in the next example:

Example

```
\[ F = G \left( \frac{m_1 m_2}{r^2}
↪  \right) \]
```

$$F = G \left(\frac{m_1 m_2}{r^2} \right)$$

Notice that to insert the parentheses or brackets, the `\left` and `\right` commands are used. Even if you are using only one bracket, both commands are mandatory, you can use invisible brackets `\left.` or `\right.` for this.

Example

```
\[ \int_a^b x^2 {\sf d} x = \left.
↪  \frac{1}{3}x^3 \right|_a^b \]
```

$$\int_a^b x^2 {\sf d} x = \left. \frac{1}{3}x^3 \right|_a^b$$

Sometimes you may want to control the sizes of the brackets yourselves, which is called manually sized brackets. The commands listed are designed for thus purpose.

Size	LaTeX	Code
big	$\big(\big)$	`\big ( \big )`
Big	$\Big[\Big]$	`\Big [ \Big ]`
bigg	$\bigg\{\bigg\}$	`\bigg \{ \bigg \}`
Bigg	$\Bigg\|$	`\Bigg` $\Bigg\vert$—

Mathematical Modes

LaTeX allows two writing modes for mathematical expressions: the inline mode and the display mode. The first one is used to write formulas that are part of a text. The second one is used to write expressions that are not part of a text or paragraph, and are therefore put on separate lines.

Mathematical Modes

LaTeX allows two writing modes for mathematical expressions: the inline mode and the display mode. The first one is used to write formulas that are part of a text. The second one is used to write expressions that are not part of a text or paragraph, and are therefore put on separate lines.

To put your equations in inline mode use `\(` and `\)`, `$` and `$` or `\begin{math}` and `\end{math}`. They all work and the choice is a matter of taste.

Mathematical Modes

LaTeX allows two writing modes for mathematical expressions: the inline mode and the display mode. The first one is used to write formulas that are part of a text. The second one is used to write expressions that are not part of a text or paragraph, and are therefore put on separate lines.

To put your equations in inline mode use `\(` and `\)`, `$` and `$` or `\begin{math}` and `\end{math}`. They all work and the choice is a matter of taste.

Example

In physics, the mass-energy equivalence is stated by the equation $E = mc^2$, discovered in 1905 by Albert Einstein.

Mathematical Modes

LaTeX allows two writing modes for mathematical expressions: the inline mode and the display mode. The first one is used to write formulas that are part of a text. The second one is used to write expressions that are not part of a text or paragraph, and are therefore put on separate lines.

To put your equations in inline mode use `\(` and `\)`, `$` and `$` or `\begin{math}` and `\end{math}`. They all work and the choice is a matter of taste.

Example

In physics, the mass-energy equivalence is stated by the equation $E = mc^2$, discovered in 1905 by Albert Einstein.

The display mode is usually used with mathematical environments together, which will be discussed in the next subsection.

Numbering of Equations

The display mode has two versions: numbered and unnumbered.

Example

The mass-energy equivalence is described by the famous equation

$$E = mc^2$$

discovered in 1905 by Albert Einstein. In natural units ($c = 1$), the formula expresses the identity

$$E = m \tag{1}$$

The equation Environment

An equation environment contains a set of maths equations

Command

```
\begin{equation*}
% ...
\end{equation*}
```

Example

$$x = y + z$$

The equation Environment

An equation environment contains a set of maths equations

Command

```
\begin{equation*}
% ...
\end{equation*}
```

Example

$$x = y + z$$

If a star(*) is added, the sequence number of the equation won't be displayed (this feature is from the amsmath package, and should behave very similar as directly using \[and \]). Note that the environment name in the \begin and \end statements must be the same (both or neither have a * here).

In math environments, unlike in plain text, normal spaces will not lead to visible spaces in output. Only `\␣` or `\quad`,`\qquad` etc. will create spaces between words.

`\partial` prints the symbol ∂, `\frac{...}{...}` makes a fraction.

`\left(` and `\right(` make braces that fit the equation's height.

The split Environment (inline)

In order to deal with extremely long equations or equation with multiple lines, we can use the split environment. It is an inline environment being used in other maths environments.

Example

```
\begin{equation}
  \begin{split}
    F &= 1+2+3+4+5 \\
      &= 15
  \end{split}
\end{equation}
```

$$\begin{split} F &= 1+2+3+4+5 \\ &= 15 \end{split} \tag{2}$$

The split Environment (inline)

In order to deal with extremely long equations or equation with multiple lines, we can use the split environment. It is an inline environment being used in other maths environments.

Example

```
\begin{equation}
  \begin{split}
    F &= 1+2+3+4+5 \\
      &= 15
  \end{split}
\end{equation}
```

$$\begin{split} F &= 1+2+3+4+5 \\ &= 15 \end{split} \tag{2}$$

& is used to align the equal marks, and \\ is used to split the equation into two lines. Only one equation number will be generated in an equation environment.

The split Environment (inline)

In order to deal with extremely long equations or equation with multiple lines, we can use the split environment. It is an inline environment being used in other maths environments.

Example

```
\begin{equation}
  \begin{split}
    F &= 1+2+3+4+5 \\
      &= 15
  \end{split}
\end{equation}
```

$$\begin{split} F &= 1+2+3+4+5 \\ &= 15 \end{split} \tag{2}$$

& is used to align the equal marks, and \\ is used to split the equation into two lines. Only one equation number will be generated in an equation environment.

The split environment is designed to serve as the entire body of an equation, or an entire line of an align or gather environment. There cannot be any printed material before or after it within the same enclosing structure.

The aligned Environment (inline)

For linear equation systems, the aligned environment can be used, which is similar to the split environment above. It is also an inline environment, which can be used in inline mode such as $$! Here split doesn't work because \left and \right is an enclosing structure. See this discussion ▸ Link for more information.

Example

```
Equations:
$
  \left\lbrace\begin{aligned}
    x+y &= 1 \\ x-y &= 1
  \end{aligned}\right.
  \Longrightarrow
  \left\lbrace\begin{aligned}
    x &= 1 \\ y &= 0
  \end{aligned}\right.
$
```

Equations: $\left\lbrace\begin{aligned} x+y &= 1 \\ x-y &= 1 \end{aligned}\right. \Longrightarrow \left\lbrace\begin{aligned} x &= 1 \\ y &= 0 \end{aligned}\right.$

Actually things can also be easier with packages like systeme, which will be demonstrated later.

The `align` Environment

An `align` environment can be used to simply the `split` or `aligned` in the `equation` environment. But it numbers the equation on each line.

Example

```
\begin{align}
  F &= 1+2+3+4+5 \\
    &= 15
\end{align}
```

$$\begin{align} F &= 1+2+3+4+5 \tag{3} \\ &= 15 \tag{4} \end{align}$$

Use `align*` so that there will be no number(s).

Example

```
\begin{align*}
  a+b & \Leftrightarrow b+a \\
  (a+b)+c & \Leftrightarrow a+(b+c)
\end{align*}
```

$$\begin{align*} a+b & \Leftrightarrow b+a \\ (a+b)+c & \Leftrightarrow a+(b+c) \end{align*}$$

The ampersand character `&` determines where the equations align. The odd columns are right-aligned, and the even ones are left-aligned, so you can use `&&` if you want to make two neighbor column aligned to the same direction.

Example

$$\begin{aligned}
\text{(right)} &\text{(left)} && \text{(left)} & \text{(right)} && \text{(right)} &\text{(left)} \\
x &= y && w & = z && a &= b + c \\
2x &= -y && 3w & = z/2 && a &= b \\
-4 + 5x &= 2 + y && w + 2 & = -1 + w && ab &= cb
\end{aligned}$$

The cases Environment (inline)

The linear system of equations can also be typeset simply with the cases environment. It is less flexible than an aligned environment, eg., there can only be one & on each row. Another minor difference is that the horizontal space before & is larger than other similar environments.

Example

```
\begin{equation}
  \left\lbrace\begin{aligned}
    x+y &= 1 \\
    x-y &= 1
  \end{aligned}\right.
\end{equation}
```

$$\left\{\begin{aligned} x+y &= 1 \\ x-y &= 1 \end{aligned}\right. \tag{5}$$

```
\begin{equation}
  \begin{cases}
    x+y &= 1 \\
    x-y &= 1
  \end{cases}
\end{equation}
```

$$\begin{cases} x+y &= 1 \\ x-y &= 1 \end{cases} \tag{6}$$

The gather Environment

If you just need to display a set of consecutive equations, centered and with no alignment whatsoever, use the gather environment. The asterisk trick to set/unset the numbering of equations also works here.

Example

```
\begin{gather}
  2x - 5y =  8 \\
  3x^2 + 9y =  3a + c
\end{gather}
```

$$2x - 5y = 8 \tag{7}$$

$$3x^2 + 9y = 3a + c \tag{8}$$

The `gathered` Environment (inline)

There is also an inline version of `gather`, called `gathered`. The relationship of them is similar to `align` and `aligned`.

Example

```
\begin{equation}
  \begin{gathered}
    2x - 5y =  8 \\
    3x^2 + 9y =  3a + c
  \end{gathered}
\end{equation}
```

$$\begin{gathered} 2x - 5y = 8 \\ 3x^2 + 9y = 3a + c \end{gathered} \tag{9}$$

The `multline` Environment

For equations longer than a line use the `multline` environment. Insert a double backslash to set a point for the equation to be broken. The first part will be aligned to the left and the second part will be displayed in the next line and aligned to the right.

Example

$$\begin{multline} p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 \\ - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \tag{10} \end{multline}$$

The equation number will be in the last line, use `multline*` for no numbering.

For equations equal or longer then three lines,

Example

$$a + b + c = 1$$

$$b + c = 2$$

$$c + d = 1$$

$$d = 3$$

Here, the first column is left-aligned, the last column is right-aligned and the others ones are center-aligned.

Horizontal Spacing

Horizontal spacing in maths mode is useful in several situations, let's see an example: [1]

Example

Assume we have the next sets

$$S = \{z \in \mathbb{C} \,|\, |z| < 1\} \quad \text{and} \quad S_2 = \partial S$$

As you see in this example, a mathematical text can be explicitly spaced by means of some special commands.

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

The spacing depends on the command you insert, the example below contains a complete list of spaces and how they look like.

Example

```
\begin{align*}
f(x) =& x^2\! +3x\! +2 \\
f(x) =& x^2+3x+2 \\
f(x) =& x^2\, +3x\, +2 \\
f(x) =& x^2\: +3x\: +2 \\
f(x) =& x^2\; +3x\; +2 \\
f(x) =& x^2\ +3x\ +2 \\
f(x) =& x^2\quad +3x\quad +2 \\
f(x) =& x^2\qquad +3x\qquad +2
\end{align*}
```

$$\begin{aligned}
f(x) =& x^2\! +3x\! +2 \\
f(x) =& x^2+3x+2 \\
f(x) =& x^2\, +3x\, +2 \\
f(x) =& x^2\: +3x\: +2 \\
f(x) =& x^2\; +3x\; +2 \\
f(x) =& x^2\ +3x\ +2 \\
f(x) =& x^2\quad +3x\quad +2 \\
f(x) =& x^2\qquad +3x\qquad +2
\end{aligned}$$

Vertical Spacing

When the space between display maths and the main body paragraph is considered larger than expectation, is there any way to modify the line spacing?

In default style of display mode is like

Example

your body paragraph is supposed to be typed here

$$a \times b = c \tag{11}$$

your body paragraph is supposed to be typed here

You can use `\setlength` to set the displayskip.

Command

```
\setlength\abovedisplayskip{<length>}
\setlength\belowdisplayskip{<length>}
```

Example

your body paragraph is supposed to be typed here

$$a \times b = c \tag{12}$$

your body paragraph is supposed to be typed here

Fractions and Binomials

Fractions and binomial coefficients are common mathematical elements with similar characteristics - one number goes on top of another. [1]

Command

```
\frac{top}{bottom}  % fraction
\binom{top}{bottom} % binomial coefficients
```

Using fractions and binomial coefficients in an expression is straightforward.

Example

The binomial coefficient is defined by the next expression:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

In inline and display mode, the appearance of the fractions and binomials may differ. You can use `\displaystyle` or `\textstyle` to adjust the size of the fractions and binomials, or use `\dfrac` if not all fractions in an equation need to be resized.

In inline and display mode, the appearance of the fractions and binomials may differ. You can use `\displaystyle` or `\textstyle` to adjust the size of the fractions and binomials, or use `\dfrac` if not all fractions in an equation need to be resized.

Example

When displaying fractions in-line, for example $\frac{3x}{2}$ you can set a different display style: $\displaystyle\frac{3x}{2}$. Or you can use $\dfrac{3x}{2}$. This is also true the other way around

$$f(x) = \binom{n}{x} = \frac{n!}{x!(n-x)!} \quad \text{and} \quad \textstyle f(x) = \binom{n}{x} = \frac{n!}{x!(n-x)!}$$

In inline and display mode, the appearance of the fractions and binomials may differ. You can use `\displaystyle` or `\textstyle` to adjust the size of the fractions and binomials, or use `\dfrac` if not all fractions in an equation need to be resized.

Example

When displaying fractions in-line, for example $\frac{3x}{2}$ you can set a different display style: $\displaystyle \frac{3x}{2}$. Or you can use $\dfrac{3x}{2}$. This is also true the other way around

$$f(x) = \binom{n}{x} = \frac{n!}{x!(n-x)!} \quad \text{and} \quad \textstyle f(x) = \binom{n}{x} = \frac{n!}{x!(n-x)!}$$

The command `\displaystyle` will format the fractions and binomials as if they were in mathematical display mode. On the other side, `\textstyle` will change the style of them as if they were part of the text.

The usage of fractions is quite flexible, they can be nested to obtain more complex expressions. And `\cfrac` can be used to make continued fractions.

Example

The fractions can be nested

$$\frac{1+\frac{a}{b}}{1+\frac{1}{1+\frac{1}{a}}}$$

Now a wild example

$$a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_3+\cdots}}}$$

Operators

Characters in mathematical mode are usually shown in italics, but sometimes especial function names require different formatting (font and skip), this is accomplished by using operators defined in LaTeX. [1]

Trigonometrical functions, logarithms, and some others can be written in a document by means of some special commands.

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

Operators

Characters in mathematical mode are usually shown in italics, but sometimes especial function names require different formatting (font and skip), this is accomplished by using operators defined in LaTeX. [1]

Trigonometrical functions, logarithms, and some others can be written in a document by means of some special commands.

Example

```
\[ \sin(a + b ) = \sin(a)\cos(b) +
↪  \cos(a)\sin(b) \]
\[ \log_a b = \frac{\log_c
↪  b}{\log_c a} = \frac{\ln b}{\ln
↪  a} \]
\[ \tan a,\quad \arccos a,\quad
↪  \arcsin a,\quad \arctan a \]
```

$$\sin(a + b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

$$\log_a b = \frac{\log_c b}{\log_c a} = \frac{\ln b}{\ln a}$$

$$\tan a,\quad \arccos a,\quad \arcsin a,\quad \arctan a$$

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

Integrals

Integral expression can be added using the command

Command

```
\int_{lower}^{upper}
```

Note, that integral expression may seems a little different in inline and display math mode - in inline mode the integral symbol and the limits are compressed.

Example

```
Integral $\int_{a}^{b} x^2 dx$
↪  inside text
\[ \int_{a}^{b} x^2 dx \]
```

Integral $\int_{a}^{b} x^2 dx$ inside text

$$\int_{a}^{b} x^2 dx$$

There is always an argue about whether *italic* or roman style of “d” should be used in integrals and derivatives. There’s no right or wrong. If you prefer to use the sans-serif style, try `\mathsf{d}`.

Multiple Integrals

To obtain double/triple/multiple integrals you must use amsmath package.

Example

```
\begin{gather*}
\iint_V \mu(u,v) \,du\,dv \\
\iiint_V \mu(u,v,w) \,du\,dv\,dw \\
\iiiint_V \mu(t,u,v,w)
↪  \,dt\,du\,dv\,dw \\
\idotsint_V \mu(u_1,\dots,u_k)
↪  \,du_1 \dots du_k \\
\end{gather*}
```

$$\begin{gather*}
\iint_V \mu(u,v) \,du\,dv \\
\iiint_V \mu(u,v,w) \,du\,dv\,dw \\
\iiiint_V \mu(t,u,v,w) \,dt\,du\,dv\,dw \\
\idotsint_V \mu(u_1,\dots,u_k) \,du_1 \dots du_k
\end{gather*}$$

Cyclic Integrals

To obtain cyclic integrals you must use `esint` package.

Example

```
\begin{gather*}
\oint_V f(s) \,ds \\
\oiint_V f(s,t) \,ds\,dt \\
\end{gather*}
```

$$\begin{gather*}
\oint_V f(s) \,ds \\
\oiint_V f(s,t) \,ds\,dt
\end{gather*}$$

Limits, Sums and Products

Like integrals, limits, sums and products expression are compressed in inline mode.

Command

```
\limits_{lower}
\sum_{lower}^{upper}
\prod_{lower}^{upper}
```

Example

Limit $\lim_{x\to\infty} f(x)$ inside text

$$\lim_{x\to\infty} f(x)$$

Example

Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

$$\sum_{n=1}^{\infty} 2^{-n} = 1$$

Example

Product $\prod_{i=a}^{b} f(i)$ inside text

$$\prod_{i=a}^{b} f(i)$$

Improvement of Integrals, Limits, Sums and Products

In inline math mode the integral/sum/product lower and upper limits are placed right of integral symbol. Similar is for limit expressions. If you want the limits of an integral/sum/product to be specified above and below the symbol in inline math mode (or in display mode), use the `\limits` command before limits specification.

Example

Integral $\int_a^b x^2 dx$ inside text

Improved integral $\int\limits_a^b x^2 dx$ inside text

Use limits in display mode

$$\int\limits_a^b x^2 dx$$

Moreover, adding `\displaystyle` beforehand will make the symbol in inline mode large and easier to read, as in display mode.

Example

Limit $\lim_{x\to\infty} f(x)$ inside text
Display style limit $\displaystyle\lim_{x\to\infty} f(x)$ inside text

On the other hand, `\mathlarger` command (provided by `relsize` package) is used to get bigger integral symbol in display.

Example

$\int \frac{1}{2} dx - \displaystyle\int \frac{1}{2} dx$

Other Math Symbols

Some examples of other common used math symbols are shown.

Name	LaTeX	Code
Square Root	$\sqrt{a}\ \sqrt[b]{a}$	`\sqrt {a}\ \sqrt [b]{a}`
Over/Under Line	$\overline{a+b}\ \underline{a+b}$	`\overline {a+b}\ \underline {a+b}`
Over Brace	$\overbrace{1+2+\cdots+n}^n$	`\overbrace {1+2+\cdots +n}^n`
Under Brace	$\underbrace{1+2+\cdots+n}_n$	`\underbrace {1+2+\cdots +n}_n`
Over Arrow	$\overrightarrow{a+b}\ \overleftarrow{a+b}$	`\overrightarrow {a+b}\ \overleftarrow {a+b}`
Under Arrow	$\underrightarrow{a+b}\ \underleftarrow{a+b}$	`\underrightarrow {a+b}\ \underleftarrow {a+b}`
Dots	$\dots \quad \cdot \quad \cdots \quad \vdots \quad \ddots$	`\dots \ \cdot \ \cdots \ \vdots \ \ddots`
Arrows	$\rightarrow \ \leftarrow \ \leftrightarrow$	`\rightarrow \ \leftarrow \ \leftrightarrow`
	$\Rightarrow \ \Leftarrow \ \Leftrightarrow$	`\Rightarrow \ \Leftarrow \ \Leftrightarrow`
	$\longleftarrow \ \Longrightarrow$	`\longleftarrow \ \Longrightarrow`

Mathematical Fonts

In mathematical mode as well as in text mode, you can change the typeface as needed. For instance, it's customary to represent real numbers with a blackboard bold font, or topological spaces with calligraphic font. [1]

For some elements is convenient to have the possibility of changing the font typeface.

Example

Let $\mathcal{T}$ be a topological space, a basis is defined as

$$\mathcal{B} = \{B_\alpha \in \mathcal{T} \mid U = \bigcup B_\alpha \forall U \in \mathcal{T}\}$$

[1] Some of this part is ported from the tutorial of Overleaf: ▸ Link

Mathematical Fonts for Capital Letters

There are some font typefaces that support only a limited number of characters; these fonts usually denote some special sets.

Example

```
\begin{gather*}
  RQSZ \\
  \mathcal{RQSZ} \\
  \mathfrak{RQSZ} \\
  \mathbb{RQSZ}
\end{gather*}
```

$$\begin{gather*} RQSZ \\ \mathcal{RQSZ} \\ \mathfrak{RQSZ} \\ \mathbb{RQSZ} \end{gather*}$$

This example shows Calligraphic, Fraktur and Blackboard bold typefaces. For instance, to display the R in blackboard bold typeface `$\mathbb{R}$` will do the trick.

Other Mathematical Fonts

It's possible to set a different font family for a complete mathematical expression.

Example

```
\begin{gather*}
  3x^2 \in R \subset Q \\
  \mathnormal{3x^2 \in R \subset Q}
  ↪  \\
  \mathrm{3x^2 \in R \subset Q} \\
  \mathit{3x^2 \in R \subset Q} \\
  \mathbf{3x^2 \in R \subset Q} \\
  \mathsf{3x^2 \in R \subset Q} \\
  \mathtt{3x^2 \in R \subset Q}
\end{gather*}
```

$$\begin{gather*}
3x^2 \in R \subset Q \\
\mathnormal{3x^2 \in R \subset Q} \\
\mathrm{3x^2 \in R \subset Q} \\
\mathit{3x^2 \in R \subset Q} \\
\mathbf{3x^2 \in R \subset Q} \\
\mathsf{3x^2 \in R \subset Q} \\
\mathtt{3x^2 \in R \subset Q}
\end{gather*}$$

In this case, not only letters but all characters change its appearance, for example `$\mathit{3x^2}$` italicises the entire expression.

Define Own Symbols

If you need to add a personalized operator to be displayed in Roman font instead of italics use `\DeclareMathOperator`, provided by the the package `amsmath`.

Example

```
\DeclareMathOperator{\Mr}{M_{\mathbb{R}}}
User-defined operator for matrices with Real entries $ x \in \Mr $
```

User-defined operator for matrices with Real entries $x \in \mathrm{M}_{\mathbb{R}}$

Define Own Symbols

If you need to add a personalized operator to be displayed in Roman font instead of italics use `\DeclareMathOperator`, provided by the the package `amsmath`.

Example

```
\DeclareMathOperator{\Mr}{M_{\mathbb{R}}}
User-defined operator for matrices with Real entries $ x \in \Mr $
```

User-defined operator for matrices with Real entries $x \in \mathrm{M}_{\mathbb{R}}$

The command can be slightly modified if you need that your defined operator uses subscripts, as the `\lim` operator, in such case use `\DeclareMathOperator*`.

Define Own Symbols

If you need to add a personalized operator to be displayed in Roman font instead of italics use `\DeclareMathOperator`, provided by the the package amsmath.

Example

```
\DeclareMathOperator{\Mr}{M_{\mathbb{R}}}
User-defined operator for matrices with Real entries $ x \in \Mr $
```

User-defined operator for matrices with Real entries $x \in \mathrm{M}_{\mathbb{R}}$

The command can be slightly modified if you need that your defined operator uses subscripts, as the `\lim` operator, in such case use `\DeclareMathOperator*`.

You can also use `\mathop` to define a italics math operator supporting subscripts, and change it to Roman font by hand.

Example

```
\[
↪   \mathop{\mathrm{limsup}}_{n\to\infty}\mathop{rot}
↪   F_n \]
```

$$\mathop{\mathrm{limsup}}_{n\to\infty}\mathop{rot} F_n$$

The matrix Environment (inline)

There are various kinds of matrix environments defined in amsmath package, they are matrix, pmatrix, bmatrix, Bmatrix, vmatrix, Vmatrix.

Command

```
\begin{[p/b/B/v/V]matrix}
  a_{11} & a_{12} & ... & a_{1n} \\
  a_{21} & a_{22} & ... & a_{2n} \\
  ...    & ...    & ... & ...    \\
  a_{n1} & a_{n2} & ... & a_{nn} \\
\end{[p/b/B/v/V]matrix}
```

Example

```
\begin{equation}
  \begin{pmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    a_{31} & a_{32} & a_{33} \\
  \end{pmatrix}
\end{equation}
```

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{13}$$

Here is some examples of the style of these matrix.

Example

`matrix`

$$\begin{matrix} a & b \\ c & d \end{matrix}$$

`bmatrix`

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

`vmatrix`

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

`pmatrix`

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

`Bmatrix`

$$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$$

`Vmatrix`

$$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

Some packages may also help simplify the typesetting of matrix, for example, there is some macros defined in the `physics` package to make identity matrix, or generate the examples above more simply.

If you need to create matrices with different delimiters, you can add them manually to a plain matrix. For example:

Example

```
\begin{equation}
  \left\lceil
  \begin{matrix}
    1 & 2 & 3 \\
    a & b & c
    \end{matrix}
  \right\rceil
\end{equation}

\begin{equation}
  \left\langle
  \begin{matrix}
    1 & 2 & 3 \\
    a & b & c
  \end{matrix}
  \right\rangle
\end{equation}
```

$$\left\lceil\begin{matrix}1 & 2 & 3 \\ a & b & c\end{matrix}\right\rceil \tag{14}$$

$$\left\langle\begin{matrix}1 & 2 & 3 \\ a & b & c\end{matrix}\right\rangle \tag{15}$$

The `smallmatrix` Environment

When typesetting inline math, the usual `matrix` environments above may look too big. It may be better to use `smallmatrix` in such situations, although you will need to provide your own delimiters.

Example

Trying to typeset an inline matrix here $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ but it looks too big, so let's try $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ instead.

The array Environment

An array environment is actually a math mode tabular environment, and the usage of them are almost the same. You can refer to the lecture about tables for this part.

A simple example is given here:

Example

```
\begin{equation}
  \chi(\lambda) =
  \left| \begin{array}{ccc}
    \lambda - a & -b & -c \\
    -d & \lambda - e & -f \\
    -g & -h & \lambda - i
  \end{array} \right|
\end{equation}
```

$$\chi(\lambda) = \left| \begin{array}{ccc} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{array} \right| \tag{16}$$

Table of Contents

1. Introduction to LaTeX
   - What is LaTeX
   - Distributions and IDEs
   - Documentation
2. Basics
   - Global Structure
   - Modular Documents
3. Text
   - Special Characters
   - Fonts
   - Underline
   - Enumeration
   - Alignment
   - Spaces, lines and pages
   - Minipage and Multicolumn
4. Use Maths in LaTeX
   - Math Expressions
   - Math Environments
   - Spacing in Math Mode
   - Basic Math Commands
   - Matrices and Arrays
5. Useful Maths Packages
   - Common Packages
   - The `systeme` Package
6. Graphs
   - Include Graphs
   - Figures
   - Draw Graphs
7. Tables
   - Tabulars
   - Tables

The AMS-LaTeX Packages

AMS-LaTeX is a collection of LaTeX document classes and packages developed for the American Mathematical Society (AMS).

It is an extension of plain-LaTeX maths, with many new maths environments (most of them were introduced in the previous section), maths symbols and maths fonts.

Usually you can insert all of the commands in the preamble of your document.

Command

```
\usepackage{amsmath}  % loads maths environments
\usepackage{amssymb}  % loads maths symbols
\usepackage{amsfonts} % loads maths fonts
```

Some Other Packages

Recall that we also use some other packages in this lecture:

Command

```
\usepackage{esint}    % for cyclic integrals
\usepackage{relsize}  % for \mathlarger
```

For a better `array` environment, though it's not mandatory (you can use it without the package), you're recommended to add the `array` package.

Command

```
\usepackage{array}
```

The systeme Package

To use the systeme package, simply insert the command in the preamble of your document.

Command

```
\usepackage{systeme}
```

This package can make it really easy when typesetting linear systems by the command `\systeme`.

Example

```
\begin{equation}
  \systeme{
    2a-3b+4c=2,
    a+8b+5c=8,
    -a+2b+c=-5
  }
\end{equation}
```

$$\left\{\begin{array}{rcrcrcr} 2a & - & 3b & + & 4c & = & 2 \\ a & + & 8b & + & 5c & = & 8 \\ -a & + & 2b & + & c & = & -5 \end{array}\right. \tag{17}$$

It also works for subscripts.

Example

```
\begin{equation}
  \systeme{
    4x_1-x_2=3,
    -x_1+5x_2=-1
  }
\end{equation}
```

$$\left\{\begin{array}{rcrcl} 4x_1 & - & x_2 & = & 3 \\ -x_1 & + & 5x_2 & = & -1 \end{array}\right. \tag{18}$$

It can also reorder the variables and numbers in the equations.

Example

```
\begin{equation}
  \systeme{
    3y+2x=0,
    x-z+9=0,
    2+3x+5-y-7+z=0
  }
\end{equation}
```

$$\left\{\begin{array}{rcrcrcrcrcrcl} 2x & + & 3y & & & & & & & & & = & 0 \\ x & & & - & z & + & & & & & 9 & = & 0 \\ 3x & - & y & + & z & + & 2 & + & 5 & - & 7 & = & 0 \end{array}\right. \tag{19}$$

Complicated coefficients can be handle correctly. Note that + and - should be replaced with \+ and \- in the coefficients.

Example

```
\begin{equation}
  \systeme{
    (2\+\sqrt{2})x-
    (1\-\sqrt{2})y=1,
    x+(1\+\sqrt{2})y=-1
  }
\end{equation}
```

$$\left\{\begin{array}{r}(2+\sqrt{2})x-(1-\sqrt{2})y=1\\ x+(1+\sqrt{2})y=-1\end{array}\right. \tag{20}$$

The documentation of the systeme package can be found in http://mirrors.ctan.org/macros/generic/systeme/systeme_fr.pdf, however it's in French, and the author is Manuel de l'utilisateur.

Table of Contents

Include Graphs

Before all, you need the `graphics` or `graphicx` package, where `graphicx` is an extended and enhanced one. So you are recommended to insert the command in the preamble of your document.

Command

```
\usepackage{graphicx}
```

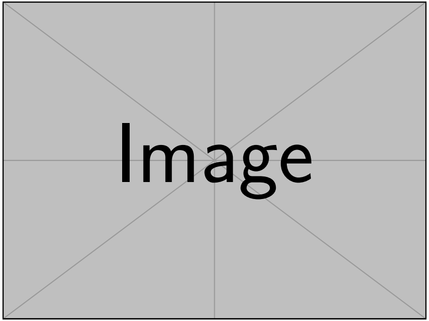
Then you can use the command `\includegraphics` to insert images of many formats, including `jpg`, `png` images and even other `pdf` files. `eps` images should be supported by most modern LaTeX distributions as well.

Command

```
\includegraphics[options]{filename}
```

There are some example images defined, you can insert them if the figure is not yet ready when writing LaTeX code. They are `example-image`, `example-image-golden`, `example-image-a`, `example-image-b` and etc.

Example



We usually use the `width` option to adjust the size of the image, according to a ratio of `\textwidth`, which means the maximum width of text here.

Options of Include Graphs

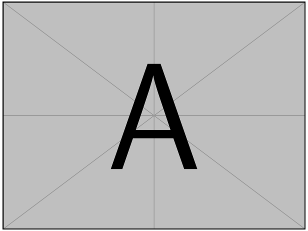
Here some useful `options` are listed:

- `height` - use any LaTeX measuring unit.
- `width` - use any LaTeX measuring unit.
- `scale` - scale the graph to this proportion
- `angle` - rotate the graph in anti-clockwise by this angle

LaTeX measuring unit can be `\textwidth`, `\linewidth`, `\textheight`, `\lineheight`, cm, pt, em, and etc..

Example

```
\includegraphics[width=4cm]%
{example-image-a}
```



The figure Environment

The figure environment provides a wrapper of image inserted by \includegraphics, which add caption and label (reference) to an image. They are especially useful in report and paper writing, here is a template of how to use the environment.

Command

```
\begin{figure}[position]
  \centering
  \includegraphics[options]{filename}
  \caption{caption}
  \label{fig:label}
\end{figure}
```

- filename - the filename or relative path of the graph you want to insert, usually placed in the same or child directory as the tex file
- position - we usually use !htbp or !H here, which will be introduced later in this chapter
- caption - the caption displayed above/under the graph
- label - used for references in a document (will be introduced later)

Labels and References

You can use \ref to have a reference of a figure by its label. The figures will be automatically numbered (like equations), and the reference is also a hyperlink.

Example

```
\begin{figure}[!htbp]
  \centering
  \includegraphics[
    width=0.7\textwidth,
    angle=90
  ]{example-image-b}
  \caption{Example Image B rotated
  ↪  by 90 degree.}
  \label{fig:img-b}
\end{figure}
B was shown in Figure
\ref{fig:img-b}.
```

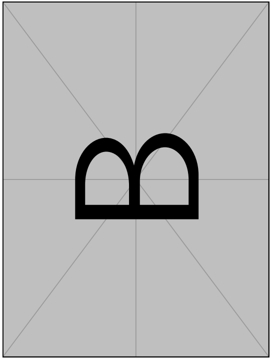


Figure: Example Image B rotated by 90 degree.

Floats and Positions

Floats are containers for things in a document that cannot be broken over a page. LaTeX by default recognizes `figure` and `table` (will be introduced later) floats.

If you don't provide the `position` option, LaTeX will try to help you find a place to set the figure. However, the position is often not ideal, so you need to add some specifiers yourselves.

- `h` - Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
- `t` - Position at the top of the page.
- `b` - Position at the bottom of the page.
- `p` - Put on a special page for floats only.
- `!` - Override internal parameters LaTeX uses for determining "good" float positions.
- `H` - Places the float at precisely the location in the LaTeX code. Requires the float package, i.e., `\usepackage{float}`.

Include Multiple Graphs

A useful extension is the `subcaption` package, which provides a `subfigure` environment to add multiple subfigures in a figure.

Note that there is also a package called `subfigure`, but is has been deprecated (not maintained), please do not use it. Another package called `subfig` provides the same commands as that of `subfigure` package. However, they can't be used together.

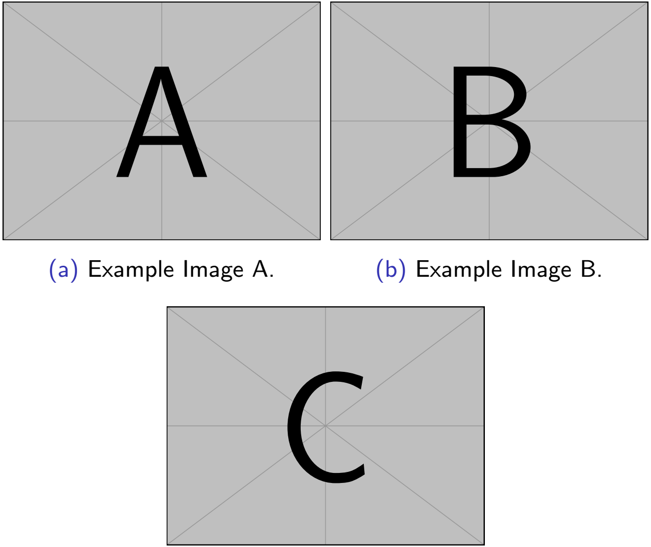
In simplicity, if there is some compatibility problem with your template after you tried the `subcaption` package, choose the `subfig` package.

Here is an example with the `subcaption` package.

Example

```
\begin{figure}
    \centering
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[width=\textwidth]{example-image-a}
        \caption{Example Image A.}
        \label{fig:subcaption-a}
    \end{subfigure}
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[width=\textwidth]{example-image-b}
        \caption{Example Image B.}
        \label{fig:subcaption-b}
    \end{subfigure}

    \begin{subfigure}{0.3\textwidth}
        \includegraphics[width=\textwidth]{example-image-c}
        \caption{Example Image C.}
        \label{fig:subcaption-c}
    \end{subfigure}
    \caption{Example Images}\label{fig:subcaption}
\end{figure}
```



(a) Example Image A.

(b) Example Image B.

(c) Example Image C.

Figure: Example Images

As shown in Figure 3, the figures can be arranged in columns and rows.

Between Figure 3a and Figure 3b, a ~ was added. You can add desired spacing between images, e. g. ~, `\quad`, `\qquad`, `\hfill` (fill all rest horizontal spaces) and etc..

Between Figure 3b and Figure 3c, a newline was added. It will force the subfigure onto a new line.

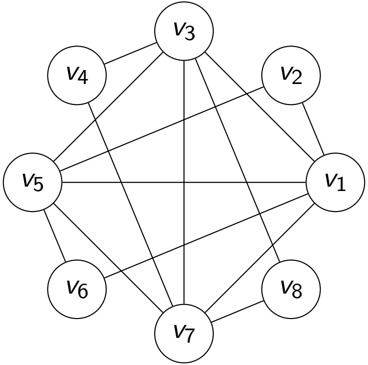
The references of subfigures can be used by their `\label` as well. For example, above references are generated by these commands:

Example

```
\ref{fig:subcaption}
\ref{fig:subcaption-a}
\ref{fig:subcaption-b}
\ref{fig:subcaption-c}
```

The tikz and pgf packages

The tikz and pgf packages can help you draw graphs in LaTeX for example:



Later we will have an introduction to tikz and pgf. If you are interested in it, please refer to the pgf manual by `texdoc tikz` or `texdoc pgf`.

Another example

Binary tree:

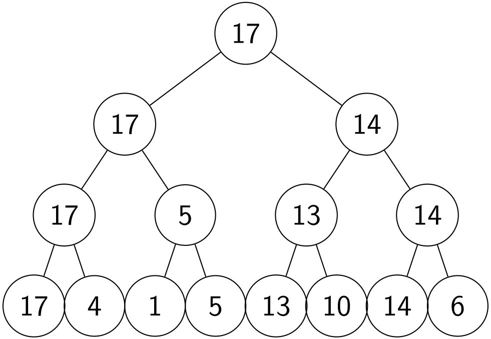


Table of Contents

1. Introduction to LaTeX
   - What is LaTeX
   - Distributions and IDEs
   - Documentation
2. Basics
   - Global Structure
   - Modular Documents
3. Text
   - Special Characters
   - Fonts
   - Underline
   - Enumeration
   - Alignment
   - Spaces, lines and pages
   - Minipage and Multicolumn
4. Use Maths in LaTeX
   - Math Expressions
   - Math Environments
   - Spacing in Math Mode
   - Basic Math Commands
   - Matrices and Arrays
5. Useful Maths Packages
   - Common Packages
   - The `systeme` Package
6. Graphs
   - Include Graphs
   - Figures
   - Draw Graphs
7. Tables
   - Tabulars
   - Tables

The `tabular` Environment

Table is another common element in LaTeX, usually you will need the `booktabs, multirow` packages for enhanced functions of tables. You can insert the command in the preamble of your document.

Command

```
\usepackage{booktabs, multirow}
```

Example

```
\begin{tabular}{lcr}
  \toprule
  Title 1 & Title 2 & Title 3 \\
  \midrule
  1 & 2 & 3 \\
  \bottomrule
\end{tabular}
```

Title 1	Title 2	Title 3
1	2	3

The syntax is similar to the `align` environment in maths. `&` is used to split the columns are `\\` is used to split the rows.

Warning

Never, ever use vertical lines in tables.
Never use double rules in tables.

They are not professional!
Please read ▸ Link for instructions on designing professional tables.

Column Format

Command

```
\begin{tabular}{format}
...
\end{tabular}
```

`format` can be set as follow:

- `|` - vertical line (not recommended)
- `l` - align left in this column
- `c` - align center in this column
- `r` - align right in this column

Example

`lll`

Title 1	Title 2	Title 3
1	2	3

`c|cc`

Title 1	Title 2	Title 3
1	2	3

Combining Rows and Columns

There are two commands being used to combine rows and columns

Command

```
\multicolumn{ncols}{format}{text}
```

- `ncols` - the number of columns to be merged
- `format` - the format of the merged column, excluding the left `|`
- `text` - the text in the merged column

```
\multirow{nrows}{width}[fixup]{text}
```

- `nrows` - the number of rows to be merged
- `width` - the width of the merged rows (use `*` for auto)
- `fixup` - the vertical position of the text (optional, default in the center)
- `text` - the text in the merged row

To use the `\multirow` command, you need to insert the package `multirow` in the preamble of your document.

Horizontal lines

There are four commands to create a horizontal line in a table.

- `\toprule` - Top line
- `\midrule` - Middle line
- `\cmidrule{2-3}` - Middle line that you can choose the columns to be drawn
- `\bottomrule` - Bottom line

Horizontal lines

There are four commands to create a horizontal line in a table.

- `\toprule` - Top line
- `\midrule` - Middle line
- `\cmidrule{2-3}` - Middle line that you can choose the columns to be drawn
- `\bottomrule` - Bottom line

Example

```
\begin{tabular}{ccccc}
  \toprule
  \multirow{4}{*}{Table} & Title 1 & Title 2 & Title 3 & Title 4 \\
  \cmidrule{2-5}
  & \multicolumn{2}{c}{Text 1} &
  \multicolumn{2}{c}{\multirow{3}{*}{Text 3}} \\
  \cmidrule{2-3}
  & \multicolumn{2}{c}{Text 2} & \multicolumn{2}{c}{} \\
  \cmidrule{2-3}
  & Text 4 & Text 5 & \multicolumn{2}{c}{} \\
  \bottomrule
\end{tabular}
```

Example

Table	Title 1	Title 2	Title 3	Title 4
	Text 1		Text 3	
	Text 2			
	Text 4	Text 5		

Just leave blank in the rest rows of `\multirow`.

Table Generators

With `\multirow` and `\multicolumn`, we can almost draw tables of any style, but this coding process can never be as easy as the graphic one, like making tables in Word or Excel. Is there any ways to convert graphic tables into LaTeX codes directly?

- Use LaTeX Table Generator: http://www.tablesgenerator.com/
- LaTeX Complex Table Editor: https://www.latex-tables.com/
- Excel2latex: https://ctan.org/tex-archive/support/excel2latex/

The `table` Environment

The `table` environment is used to arrange the place of a tabular, similar to the `figure` environment. Here is a template of how to use the environment.

Command

```
\begin{table}[position]
  \centering
  \caption{caption}
  \begin{tabular}{format}
    ...
  \end{tabular}
  \label{table:label}
\end{table}
```

The `position`, `caption`, `label` are same as those in the `figure` environment. It's recommended to put the caption above the table.

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

1. Change `[h]` to `[!h]`

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

1. Change `[h]` to `[!h]`
2. Change `[!h]` to `[!H]`

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

1. Change `[h]` to `[!h]`
2. Change `[!h]` to `[!H]`
3. Use `\newpage` to move the following content to the next page

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

1. Change `[h]` to `[!h]`
2. Change `[!h]` to `[!H]`
3. Use `\newpage` to move the following content to the next page

Recall the Positions

We usually want to place the graphs or tables just below or above the content where we mention them, but even when we type `[h]` in position, you can not ensure that it will appear at the ideal position, and there are several methods to make up for this. You can try them one by one:

1. Change `[h]` to `[!h]`
2. Change `[!h]` to `[!H]`
3. Use `\newpage` to move the following content to the next page

Usually you don't need to pay too much attention about where the figures and tables are exactly are because you can use `\ref` to reference them. And the numbering of figures and tables will strictly follow the order of their code.

References

- Wikibook. https://en.wikibooks.org/wiki/LaTeX.

- Overleaf Knowledge Base. https://www.overleaf.com/learn.

- tc-imba, zhang et al. LaTeX Lecture.
  https://github.com/SJTU-UMJI-Tech/LaTeX.

Resources

- LaTeX Cheet Sheet
- The Comprehensive TeX Archive Network
- Awesome LaTeX