Introduction to LaTeX

Liu Yihao

SJTU-UMJI Technology Department

June 22, 2021

Table of Contents

Liu Yihao

Appendix
Symbol table
Package List
Reference

- Lecture I: Hello, LaTeX
- Lecture II: Text
- Lecture III: Maths
- Lecture IV: Graphs, Tables and Code
- Lecture V: Beamer
- Appendix

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Lecture I

Hello, LaTeX

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started
- What is LaTeX
- Distributions and IDEs
- Documentation

2 Layout of a Document

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

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

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

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.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started
- What is LaTeX
- Distributions and IDEs
- Documentation

2 Layout of a Document

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Installation of LaTeX

Though there are some other distributions of LaTeX(like MikTeX), TeXLive is recommended in this lecture.

Windows & Linux

Download TeXLive on the tuna mirrors

https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/

MacOS

Download MacTeX on the tuna mirrors

https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/mac/mactex/

Linux (Debian/Ubuntu)

Enter the command (fast with apt source mirror)

```
sudo apt-get install texlive-full
```

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Selection of IDEs

There are various IDEs recommended that support LaTeX, for example

Texmaker

http://www.xm1math.net/texmaker/

Sublime Text

http://www.sublimetext.com/

Follow the instructions on https://www.zhihu.com/question/36038602

Visual Studio Code

https://code.visualstudio.com/

Follow the instructions on https://zhuanlan.zhihu.com/p/38178015

They all have cross-platform support for Windows, Linux and MacOS.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

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.

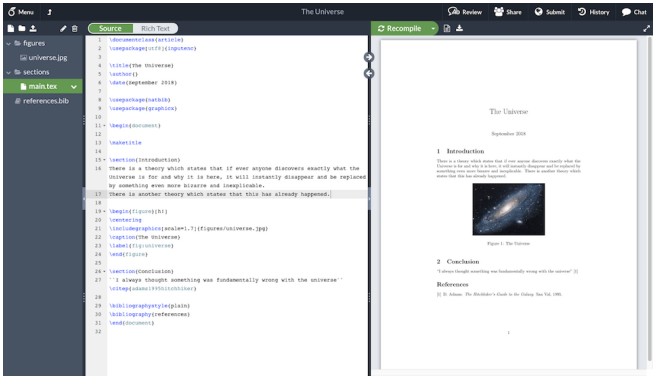


Figure 1: Layout of the Overleaf Online LaTeX Editor.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started
- What is LaTeX
- Distributions and IDEs
- Documentation

2 Layout of a Document

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

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

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.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document
- A Simple Document
- Document Classes
- The Preamble
- Document Body
- Document Sections

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

A Simple Document

A typical (simplest) LaTeX example is presented here.

Example

```
\documentclass[a4paper]{article}
\usepackage{amsmath} % Define various maths environments
\usepackage{amssymb} % Define various maths symbols
\usepackage{geometry} % Adjust the margin, paper size, and etc.
\usepackage[shortlabels]{enumerate} % Provide different style of lists
\usepackage{graphicx} % Insert image of all types
% Use other packages and setup them here
\title{A simple \LaTeX\ document}
\author{XX XXX}
\date{\today}

\begin{document}
    \maketitle
    Hello, \LaTeX !
\end{document}
```

Code started with `\` is called a command, and a pair of `\begin{}` and `\begin{}` is called an environment.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document
- A Simple Document
- Document Classes
- The Preamble
- Document Body
- Document Sections

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

All Begins with documentclass

Definition

In a LaTeX file, the first line must be

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

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

article	Write a report or an science article
report	Write a report
beamer	Produce a lecture silde like this!

Some options can be added, for example, a typical case can be

```
\documentclass[11pt,twoside,a4paper]{article}
```

Some details about the article class will be introduced on the next page. More features about other classes and options can be found in the LaTeX Document on your own.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

The article Class

The article class is one of the most basic class in LaTeX, it provides you with some normalized structure and format for report writing. So usually you will use the following command as the first line of your tex document:

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

Some of the options values are listed below (the default values are alerted)

- 10pt, 11pt, 12pt or other sizes - the font size of the document
- a4paper, a5paper, letterpaper - the size of paper
- fleqn - make the math equations left aligned (default middle aligned)
- leqno - display the serial numbers of math equations on the left (default on the right)
- titlepage, notitlepage - whether to make the title an entire page
- onecolumn, twocolumn - the number of columns of the document
- twoside, oneside - influence the position of something on the page

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Other classes

This project is open sourced and you can read the source code Link to learn much (I promise) about the `beamer` class and some very interesting features of LaTeX itself. There may also be a lecture about the `beamer` class in the future.

When writing a long report, `report` class can be used to provide some more layers of document (such as `chapter`) and different type settings. It's very similar to the `article` class, so it won't be specified.

There are some other document classes such as `minimal`, `book`, `letter` and etc., but I think you may never use them.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document
- A Simple Document
- Document Classes
- The Preamble
- Document Body
- Document Sections

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

The Preamble of a Document

As in the simple example of a document, you should notice that there is a pair of

Command

```
\begin{document}
  % some contents
\end{document}
```

This is called the body of the document, and everything before the body, including the `\documentclass` line, is called the preamble of the document.

In the preamble, you define the type of document you are writing and the language, load extra packages you will need, and set several parameters. For instance, a simplified document of the example above preamble would look like this:

Example

```
\documentclass[a4paper]{article}
\title{A simple \LaTeX\ document}
\author{XX XXX}
\date{\today}
```

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Title, Author and Date

It's very useful to generate a title on the first page of a document, in order to achieve it, these commands should first be added in the preamble.

Example

```
1 \title{title}
2 \author{author name}
3 \date{\today}
```

You can simply use `\date{\today}` to display your system date now.

Then in the body (will be introduced in the next section), use the command `\maketitle` to generate the title, or title page if you added the option `titlepage` in the `\documentclass`.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Magic of Packages

LaTeX is a macro-based language, where most of useful commands are not built-in commands. These commands are defined in various packages, which should be included in the preamble.

Command

```
\usepackage[options]{package}
```

There are some very useful packages that you may ALWAYS include:

- `amsmath` Define various maths environments
- `amssymb` Define various maths symbols
- `geometry` Adjust the margin, paper size, and etc.
- `enumitem` Generate a list like this!
- `graphicx` Insert images of all types

The usages of these and more packages will be introduced further.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Common Packages

Here I provide a list of commonly used packages, you can start from using them after the lecture.

```
\usepackage{hyperref}    % Extensive support for hypertext
\usepackage{float}       % Improved interface for floating objects
\usepackagep[margin=2.5cm]{geometry} % Flexible document dimensions
\usepackage[shortlabels]{enumerate}  % Enumerate with redefinable labels
\usepackage{multirow}    % Tabular cells spanning multiple rows
\usepackage{multicol}    % Intermix single and multiple columns
\usepackage{ulem}        % Package for underlining
\usepackage{graphicx}    % Enhanced support for graphics
\usepackage{subfig}      % Figures broken into subfigures
\usepackage{amsmath}     % AMS mathematical facilities
\usepackage{amssymb}     % AMS symbols
\usepackage{amsfonts}    % AMS fonts
\usepackage{mathrsfs}    % Support for using RSFS fonts in maths
\usepackage{latexsym}    % LaTeX symbols
\usepackage{verbatim}    % Reimplementation of LaTeX verbatim
```

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document
- A Simple Document
- Document Classes
- The Preamble
- Document Body
- Document Sections

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Main Body of Document

The main body of your document which starts with `\begin{document}` and ends with `\end{document}` can be also called the `document` environment. All of the contents you'd like to display should be in it, and it MUST be unique in the whole file.

Example

```
\begin{document}
    \maketitle
    \newpage   %start the following contents in a new page
    \tableofcontents %automatically generate a table of content
    \newpage
    Hello, \LaTeX !
    % TODO: Add more contents
    ...
\end{document}
```

The position and order of title page and table of contents can be arbitrary, and there can be multiple table of contents in one document.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

The abstract Environment

When you are writing a paper, an abstract is often necessary in the beginning of the document.

Example

```
\begin{abstract}
  This is a lecture about how to getting start in \LaTeX!
\end{abstract}
```

Abstract

This is a lecture about how to getting start in LaTeX!

The styling of the abstract will be based on the documentclass you are using. The example shows an abstract in the beamer class, which will be slightly different from that in the article class.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Comments

As in other programming languages, comments are useful when you want to make your code readable. Adding a `%` can make the whole line after it into a comment.

Example

```
% This is a comment
```

If you need multiline comments, use the `comment` environment provided by the `comment` package. (Add `\usepackage{comment}` to your preamble.)

Example

```
\begin{comment}
 some comments
 some other comments
\end{comment}
```

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Note that in the compiling, anything after a % is omitted, including the newline character, so there is no space between "comment" and "no" in the second line.

Example

```
A line
with space between ``line'' and ``with''

A line ended with comment% comments
no space between ``comment'' and ``no''
```

A line with space between "line" and "with"
A line ended with commentno space between "comment" and "no"

PS: One newline, or any number of space and tab characters are usually considered as a single "spacing" in LaTeX compilers. Two or more continuous newlines will cause a line break. We'll discuss it later in the lecture.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document
- A Simple Document
- Document Classes
- The Preamble
- Document Body
- Document Sections

3 Learn More

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Sections

Commands to organize a document vary depending on the document type, the simplest form of organization is the sectioning, available in all formats.

Command

```
\section{name}
\subsection{name}
\subsubsection{name}
```

```
\section*{name}
\subsection*{name}
\subsubsection*{name}
```

The default style (can be changed with `\renewcommand`) of sections is like

1 Example Section Name
1.1 Example Subsection Name
1.1.1 Example Subsubsection Name

If an asterisk (*) is added, the sequence number will be hidden, and it won't be added to the table of contents.

Note: (Sub)sections are `commands`, and the whole contents between two (sub)sections is belonged to the former (sub)section.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Other Structures - Chapter, Part and Paragraph

Command

```
\chapter{name}
\part{name}
\paragraph{name}
\subparagraph{name}
```

```
\chapter*{name}
\part*{name}
\paragraph*{name}
\subparagraph*{name}
```

In document classes such as `report` and `book`, some outer structures of `section` (`\chapter` and `\part`) can be used.

`\paragraph` and `\subparagraph` are used for the title of small paragraphs in a (sub)section.

If an asterisk (*) is added, the effect will be the same as in the sections (sequence numbers will be hidden).

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Levels of Document Structures

There are up to 7 levels of depth for defining sections depending on the document class:

Level	Command
-1	`\part{part}`
0	`\chapter{chapter}`
1	`\section{section}`
2	`\subsection{subsection}`
3	`\subsubsection{subsubsection}`
4	`\paragraph{paragraph}`
5	`\subparagraph{subparagraph}`

`\part` and `\chapter` are not available in some document classes.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document

3 Learn More
- Commands
- Environments

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Common Syntax of LaTeX Commands

All LaTeX commands have the following syntax

Command

```
\commandName<specialArgs>[optionalArgs]{requiredArgs}
```

`specialArgs` Seldom used in basic usage, for certain special usages in some packages

`optionalArgs` Used to define mode of the command, if not specified, LaTeX will use the default mode

`requiredArgs` Must be filled

If you want to connect a letter after a command, a space must be appended after the command or LaTeX won't be able to compile it correctly. But two commands can be directly connected since there is a `\` before each command.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Define New Commands

In LaTeX, you can define a new command (must not already exist) with

Command

```
\newcommand{\commandName}[args]{definition}
```

The definitions of new commands are usually put in the preamble. If there are no arguments, you can omit the optional `[args]`; or use `#num` to fill in the arguments.

Example

```
1  \newcommand{\examplelatexcommand}[1]{%
2  This lecture is #1!%
3  }%
4
5  \examplelatexcommand{interesting}
6  \examplelatexcommand{great}
```

This lecture is interesting! This lecture is great!

Here I use the comment character `%` in the end of each line of the definition to prevent adding newlines in the new command.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Renew Commands

You can also redefine a command (must already exist) with

Command

```
\renewcommand{\commandName}[args]{definition}
```

Example

```
1  \newcommand{\examplelatexcommand}[1]{...}%
2  \renewcommand{\examplelatexcommand}[1]{%
3  This lecture is not #1!%
4  }%
5
6  \examplelatexcommand{interesting}
7  \examplelatexcommand{great}
```

This lecture is not interesting! This lecture is not great!

`\renewcommand` is often used to change the style of `section`, `subsection` and etc., for example

Example

```
\renewcommand{\thesection}{\Roman{section}.}
\renewcommand{\thesubsection}{\Roman{section}.\arabic{subsection}}
```

This example changes the section number to capital roman numbers and subsection number to arabic numbers. Here's a list of available styles:

`\arabic`	prints the value as an Arabic number, e.g. 2.
`\alph`	prints the value as an alphabetic character (minuscule), e.g. b.
`\Alph`	prints the value as an alphabetic character (capital letter), e.g. B.
`\roman`	prints the value as a Roman number (minuscules), e.g. ii.
`\Roman`	prints the value as a Roman number (capital letters), e.g. II.
`\fnsymbol`	prints the value as a symbol in a sequence, this is meant to be used for symbolic footnotes, e.g. †.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

1 Getting Started

2 Layout of a Document

3 Learn More
- Commands
- Environments

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Common Syntax of LaTeX Environments

All LaTeX environments have the following syntax

Command

```
\begin{environmentName}<specialArgs>[optionalArgs]{requiredArgs}
    % ...
\end{environmentName}
```

`specialArgs`, `optionalArgs`, `requiredArgs` are similar to those in a `command`

It is recommended to have a indent in each environment or your tex codes will be difficult to read by others or even yourself.

Getting Started
What is LaTeX
Distributions and IDEs
Documentation
Layout of a Document
A Simple Document
Document Classes
The Preamble
Document Body
Document Sections
Learn More
Commands
Environments

Define New Environments and Renew Environments

You can define a new environment (must not already exist) with

Command

```
\newenvironment{environmentName}[args]{before begin}{after end}
```

The difference of defining an `environment` from defining a `command` is that you should specify two code blocks, one is inserted before the `\begin` clause and the other is inserted after the `\end` clause.

Another issue is that arguments can only been used in the first of them (before `\begin`). If you need to save some arguments, use `\newcommand` to define a macro, but it may cause problems in nested usages.

Redefine an environment (must already exist) with

Command

```
\renewenvironment{environmentName}[args]{before begin}{after end}
```

For example, the examples in this lecture are provided by a self-defined `latexexample` environment:

Example

```
\newenvironment{latexexample}
{\VerbatimOut{\jobname.tmp}}
{\endVerbatimOut
\begin{example}
~
\inputminted{latex}{\jobname.tmp}

\input{\jobname.tmp}
\end{example}
}
\begin{latexexample}
  some code here
\end{latexexample}
```

It is a `verbatim` environment, which accepts LaTeX code as plain text and deals with them later.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Lecture II

Text

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text
- Special Characters and Accents
- Fonts
- Underline

5 Typesetting

6 Learn more - multi languages and scope in LaTeX

Special Characters

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

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

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 (...)

Accent on letters

Sometimes you may need an accent form of a letter, here is an example of letter o

\`{o}	ò	\'{o}	ó	\^{o}	ô	\"{o}	ö	\~{o}	õ
\={o}	ō	\.{o}	ȯ	\u{o}	ŏ	\v{o}	ǒ	\H{o}	ő
\t{oo}	o͡o	\r{o}	o̊	\c{o}	o̧	\d{o}	ọ	\b{o}	o̲

Something interesting

You may be curious about how to print words like LaTeX, actually it's defined as a command.

- \TeX - TeX
- \LaTeX - LaTeX
- \LaTeXe - LaTeX 2$_\varepsilon$

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text
- Special Characters and Accents
- Fonts
- Underline

5 Typesetting

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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}

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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

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

- \mathbf - **Sample Text**
- \mathit - *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}`.

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)

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text
- Special Characters and Accents
- Fonts
- Underline

5 Typesetting

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting
- Enumeration
- Alignment
- Spaces, lines and pages
- Minipage and Multicolumn

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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.], ...

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Itemize

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

Command

```
\usepackage{enumerate}
\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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting
- Enumeration
- Alignment
- Spaces, lines and pages
- Minipage and Multicolumn

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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}
```

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting
- Enumeration
- Alignment
- Spaces, lines and pages
- Minipage and Multicolumn

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Spaces may be confusing

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

- ␣ - the basic space in LaTeX (printed in yellow since it's transparent). Note 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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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)
- \\[offset] - begin a new line with an vertical offset, offset is the size of needed space (not recommended, using `\vspace` instead.)
- \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).

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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).

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting
- Enumeration
- Alignment
- Spaces, lines and pages
- Minipage and Multicolumn

6 Learn more - multi languages and scope in LaTeX

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Minipage

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

Example

```
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
\vfill % Fill vertical space
```

```
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
\hfill % Fill horizontal space
\begin{minipage}{0.32\linewidth}
  % ...
\end{minipage}
```

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

The code above generate six minipages in a grid of 3 columns $\times$ 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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

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.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting

6 Learn more - multi languages and scope in LaTeX
- Multiple Languages
- Scope

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Spelling languages

If you want to use a spelling language with characters similar to English, package babel can be used (exactly the same name as babel).

Command

```
\usepackage[languages]{babel}
```

- languages - a list of languages, the last one to be the default language

Example

```
\usepackage[greek,english]{babel}
\textgreek{abcdefgABCDEFG}
```

Then LaTeX will print αβςδεφγAB῍ΔΕΦΓ
Of course, you can use some simple commands to print these greek letters directly, such as `\alpha`, `\beta` and etc, which is more convenient only when few of them are needed.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Chinese

The Chinese TeX Community maintains a package called ctex for inputing Chinese in LaTeX. Note that it is only a package, which is shipped with most modern TeX Suites, not the CTEX Suite. I don't think it's a good choice to use the CTEX Suite directly.

Command

```
\usepackage{ctex}
```

The default LaTeX compiler pdflatex doesn't have support on Chinese input with ctex package, xelatex is a recommended modern LaTeX compiler as a replacement.

However, the ctex package is too heavy and it can slow down the total compilation speed seriously.

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

4 Polishing the plain text

5 Typesetting

6 Learn more - multi languages and scope in LaTeX
- Multiple Languages
- Scope

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

Usage of scope in LaTeX

First, you should realize the meaning of "scope" in programming. Let's start with a simple example in C/C++ (assuming you know that):

```
1  int main()
2  { // The scope "main" of function main
3    int a = 1; // int a is defined in scope "main"
4    for (int i = 0; i < 10; i++)
5    { // The scope "for" of the for loop
6      int b = i; // int b and i are both defined in scope "for"
7      a += b; // int a can be visited here!
8    }
9    { // The scope "other", we can directly define a scope like this
10     int c; // int c is defined in scope "other"
11     c = a; // int a can be visited here!
12   }
13   a -= c // error: c is not in scope "main", can't be visited!
14 }
```

Polishing the plain text
Special Characters and Accents
Fonts
Underline
Typesetting
Enumeration
Alignment
Spaces, lines and pages
Minipage and Multicolumn
Learn more - multi languages and scope in LaTeX
Multiple Languages
Scope

In the example of C/C++, we use brackets {} to define a scope, which is just the same in LaTeX. In addition, notice that an environment or a command also defines a scope.

Example

```
1  black (default) text \\
2  \color{blue}
3  blue text \\
4  { \color{brown} brown text }
5  \begin{center}
6    \color{red}
7    centered red text
8  \end{center}
9  \textbf{ \color{brown}
10 bold brown text } \\
11 blue text
```

black (default) text
blue text
brown text

centered red text

bold brown text
blue text

With the usage of scopes, you can flexibly change the color, font or anything else you wish in a self-defined range of the document.

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Lecture III

Maths

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX
- Math Expressions
- Math Environments
- Spacing in Math Mode
- Basic Math Commands
- Matrices and Arrays

8 Useful Maths Packages

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Introduction

Basic equations in LaTeX can be easily "programmed", for example: [1]

Example

```
1  The well known Pythagorean theorem \(x^2 + y^2 = z^2\) was
2  proved to be invalid for other exponents.
3  Meaning the next equation has no integer solutions:
4
5  \[ x^n + y^n = z^n \]
```

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:

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

[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 (including myself in the past) 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:

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	$\lVert x+y \rVert$	`\lVert x+y \rVert`
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

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

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

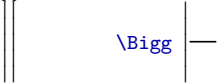
Notice that to insert the parentheses or brackets, the `\left` and `\right` commands are used. Even if you are using only one bracket, both commands are mandatory, you can use invisible brackets `\left.` or `\right.` for this.

Example

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

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

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 ]`
bigg		`\bigg \{ \bigg \}`
Bigg		`\Bigg |` —

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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

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

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

```
1 The mass-energy equivalence is described by the famous equation
2 \[E=mc^2\]
3 discovered in 1905 by Albert Einstein.
4 In natural units ($c$ = 1), the formula expresses the identity
5 \begin{equation}
6 E=m
7 \end{equation}
```

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

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX
- Math Expressions
- Math Environments
- Spacing in Math Mode
- Basic Math Commands
- Matrices and Arrays

8 Useful Maths Packages

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

The equation Environment

An equation environment contains a set of maths equations

Command

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

Example

$$\mathrm{rot}\, F = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right)\hat{n_x} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right)\hat{n_y} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\hat{n_z} \tag{2}$$

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).

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

The LaTeX script of the equation above is quite long, but not so difficult as you think so. All of the useless spaces are omitted, so please pay attention to the necessary spaces (marked in ␣).

```
\begin{equation}
    \mathop{\rm␣rot}F=\left(\frac{\partial␣F_z}{\partial␣y}
        -\frac{\partial␣F_y}{\partial␣z}\right)\hat{n_x}
        +\left(\frac{\partial␣F_x}{\partial␣z}
        -\frac{\partial␣F_z}{\partial␣x}\right)\hat{n_y}
        +\left(\frac{\partial␣F_y}{\partial␣x}
        -\frac{\partial␣F_x}{\partial␣y}\right)\hat{n_z}
\end{equation}
```

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.

It is written in plain-LaTeX, and things can even be easier with packages like `physics`, which will be demonstrated later.

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}
```

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

& 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

```
1  Equations:
2  $
3    \left\lbrace\begin{aligned}
4      x+y &= 1 \\ x-y &= 1
5    \end{aligned}\right.
6    \Longrightarrow
7    \left\lbrace\begin{aligned}
8      x &= 1 \\ y &= 0
9    \end{aligned}\right.
10 $
```

Equations: $\begin{cases} x+y=1 \\ x-y=1 \end{cases} \Longrightarrow \begin{cases} x=1 \\ y=0 \end{cases}$

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}
```

$$F = 1+2+3+4+5 \tag{4}$$
$$= 15 \tag{5}$$

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*}
```

$$a+b \Leftrightarrow b+a$$
$$(a+b)+c \Leftrightarrow a+(b+c)$$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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{align*}
  \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{align*}
```

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

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}

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

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

$$\begin{cases} x+y &= 1 \\ x-y &= 1 \end{cases} \tag{7}$$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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}
```

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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}
```

$$\begin{gathered} 2x - 5y = 8 \\ 3x^2 + 9y = 3a + c \end{gathered} \tag{10}$$

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

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

$$p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \quad (11)$$

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

For equations equal or longer then three lines,

Example

```
\begin{multline*}
  a+b+c=1 \\
  b+c=2 \\
  c+d=1 \\
  d=3
\end{multline*}
```

$$a+b+c=1$$
$$b+c=2$$
$$c+d=1$$
$$d=3$$

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

The flalign Environment

For equations aligned left, use the flalign environment. It is similar to the align environment.

Example

```
1  \begin{flalign}
2    a+b &=1=& b+a \\
3    b   &=2=& c
4  \end{flalign}
```

$$a+b=1= \qquad b+a \tag{12}$$
$$b=2= \qquad c \tag{13}$$

You may notice that the columns are flushed left (start from the left most position) and the right most column is flushed right, different from the align environment.

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX
- Math Expressions
- Math Environments
- Spacing in Math Mode
- Basic Math Commands
- Matrices and Arrays

8 Useful Maths Packages

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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
\[
S = \{ z \in \mathbb{C}\, |\, |z| < 1 \} \quad
\textrm{and} \quad S_2=\partial{S}
\]
```

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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*}
```

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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
\begin{equation}
  a \times b = c
\end{equation}
your body paragraph is supposed to be typed here
```

your body paragraph is supposed to be typed here

$$a \times b = c \tag{14}$$

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

```
\setlength\abovedisplayskip{0em}
\setlength\belowdisplayskip{0em}
your body paragraph is supposed to be typed here
\begin{equation}
  a \times b = c
\end{equation}
your body paragraph is supposed to be typed here
```

your body paragraph is supposed to be typed here

$$a \times b = c \tag{15}$$

your body paragraph is supposed to be typed here

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX
- Math Expressions
- Math Environments
- Spacing in Math Mode
- Basic Math Commands
- Matrices and Arrays

8 Useful Maths Packages

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:
\[ \binom{n}{k} = \frac{n!}{k!(n-k)!} \]
```

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.

Example

```
1  When displaying fractions in-line, for example $\frac{3x}{2}$
2  you can set a different display style:  $ \displaystyle \frac{3x}{2} $.
3  Or you can use $\dfrac{3x}{2}$. This is also true the other way around
4  \[ f(x)=\binom{n}{x}=\frac{n!}{x!(n-x)!} \quad \textrm{and} \quad
5     f(x)=\textstyle\binom{n}{x}=\frac{n!}{x!(n-x)!} \]
```

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

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

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

```
1  The fractions can be nested
2  \[ \frac{1+\frac{a}{b}}{1+\frac{1}{1+\frac{1}{a}}} \]
3  Now a wild example
4  \[ a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_3+\cdots}}} \]
```

The fractions can be nested

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

Now a wild example

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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 roman style, try `commath` or `physics` package. Either of them provides some macros to insert the "d" you want simply.

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Multiple Integrals

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

Example

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

$$\iint_V \mu(u,v)\,du\,dv$$
$$\iiint_V \mu(u,v,w)\,du\,dv\,dw$$
$$\iiiint_V \mu(t,u,v,w)\,dt\,du\,dv\,dw$$
$$\idotsint_V \mu(u_1,\dots,u_k)\,du_1\dots du_k$$

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*}
```

$$\oint_V f(s)\,ds$$
$$\oiint_V f(s,t)\,ds\,dt$$

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^^I
\[ \lim_{x\to\infty} f(x) \]
```

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

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

Example

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

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

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

Example

```
1  Product $\prod_{i=a}^{b} f(i)$ inside text
2  \[ \prod_{i=a}^{b} f(i) \]
```

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

```
1  Integral $\int_{a}^{b} x^2 dx$ inside text \par
2  Improved integral $\int\limits_{a}^{b} x^2 dx$ inside text \par
3  Use limits in display mode \[ \int\limits_{a}^{b} x^2 dx \]
```

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

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

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

```
1  $\int\frac{1}{2}dx - \mathlarger{ \int\frac{1}{2}dx}$
```

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

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	$\dots\ \cdot\ \cdots\ \vdots\ \ddots$	`\dots \ \cdot \ \cdots \ \vdots \ \ddots`
Arrows	$\rightarrow\ \leftarrow\ \leftrightarrow$	`\rightarrow \ \leftarrow \ \leftrightarrow`
	$\Rightarrow\ \Leftarrow\ \Leftrightarrow$	`\Rightarrow \ \Leftarrow \ \Leftrightarrow`
	$\longleftarrow\ \Longrightarrow$	`\longleftarrow \ \Longrightarrow`

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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

```
1  Let \( \mathcal{T} \) be a topological space, a basis is defined as
2  \[ \mathcal{B} = \{B_{\alpha} \in \mathcal{T}\, |\,
3     U = \bigcup B_{\alpha} \forall U \in \mathcal{T} \} \]
```

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

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

[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*}
```

$$RQSZ$$
$$\mathcal{RQSZ}$$
$$\mathfrak{RQSZ}$$
$$\mathbb{RQSZ}$$

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*}
```

$$3x^2 \in R \subset Q$$
$$3x^2 \in R \subset Q$$
$$3\mathrm{x}^2 \in \mathrm{R} \subset \mathrm{Q}$$
$$3\mathit{x}^2 \in \mathit{R} \subset \mathit{Q}$$
$$\mathbf{3x^2 \in R \subset Q}$$
$$3\mathsf{x}^2 \in \mathsf{R} \subset \mathsf{Q}$$
$$3\mathtt{x}^2 \in \mathtt{R} \subset \mathtt{Q}$$

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 $x \in \mathrm{M}_{\mathbb{R}}$

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

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

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX
- Math Expressions
- Math Environments
- Spacing in Math Mode
- Basic Math Commands
- Matrices and Arrays

8 Useful Maths Packages

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}
```

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{16}$$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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}
```

$$\left\lceil \begin{matrix} 1 & 2 & 3 \\ a & b & c \end{matrix} \right\rceil \tag{17}$$

$$\left\langle \begin{matrix} 1 & 2 & 3 \\ a & b & c \end{matrix} \right\rangle \tag{18}$$

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

```
1  Trying to typeset an inline matrix here
2  $\begin{pmatrix}
3    a & b \\
4    c & d
5  \end{pmatrix}$
6  but it looks too big, so let's try
7  $\big(\begin{smallmatrix}
8    a & b \\
9    c & d
10 \end{smallmatrix}\big)$ instead.
```

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

$$\chi(\lambda) = \begin{vmatrix} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{vmatrix} \tag{19}$$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX

8 Useful Maths Packages
- Common Packages
- The physics Package
- The systeme Package
- The gauss Package

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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

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

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

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}
```

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX

8 Useful Maths Packages
- Common Packages
- The physics Package
- The systeme Package
- The gauss Package

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

The physics Package

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

Command

```
1  \usepackage{physics}
```

The goal of this package is to make typesetting equations for physics simpler, faster, and more human-readable. But it can also be used in various maths circumstances.

To that end, the commands included in this package have names that make the purpose of each command immediately obvious and remove any ambiguity while reading and editing physics code.

The documentation of the physics package can be found in http://mirrors.ctan.org/macros/latex/contrib/physics/physics.pdf.

Recall the equation:

```
\begin{equation}
    \mathop{\rm rot}F=\left(\frac{\partial F_z}{\partial y}
        -\frac{\partial F_y}{\partial z}\right)\hat{n_x}
        +\left(\frac{\partial F_x}{\partial z}
        -\frac{\partial F_z}{\partial x}\right)\hat{n_y}
        +\left(\frac{\partial F_y}{\partial x}
        -\frac{\partial F_x}{\partial y}\right)\hat{n_z}
\end{equation}
```

Now we can rewrite it with the commands defined in the physics package.

Example

```
\begin{equation}
  \mathop{\rm rot}F=\qty(\pdv{F_z}{y}-\pdv{F_y}{z})\hat{n_x}+
                    \qty(\pdv{F_x}{z}-\pdv{F_z}{x})\hat{n_y}+
                    \qty(\pdv{F_y}{x}-\pdv{F_x}{y})\hat{n_z}
\end{equation}
```

$$\mathrm{rot}\, F = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right)\hat{n_x} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right)\hat{n_y} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\hat{n_z} \tag{20}$$

Automatic Bracing

When typesetting maths equations, you may use something like `\left(` and `\right)` to make the braces taller than the typical ones. The `physics` package provides some macros to simplify and replace them.

`\quantity`

`\qty(\typical)` → (■)
`\qty(\tall)` → 
`\qty(\grande)` → 
automatic ( ) braces

`\qty[\typical]` → [■]
automatic [ ] braces

`\qty|\typical|` → |■|
automatic | | braces

`\qty{\typical}` → {■}
automatic { } braces

`\qty\big{}` →
`\qty\Big{}` → 
`\qty\bigg{}` → 
`\qty\Bigg{}` →
manual sizing (works with any of the above bracket types)

`\pqty{}` ↔ `\qty()`
`\bqty{}` ↔ `\qty[]`
`\vqty{}` ↔ `\qty||`
`\Bqty{}` ↔ `\qty{}`
alternative syntax; robust and more LaTeX-friendly

\absolutevalue	\abs{a} → $\lvert a \rvert$	automatic sizing; equivalent to \qty |a|
	\abs\Big{a} → $\Big\lvert a \Big\rvert$	inherits manual sizing syntax from \qty
	\abs*{\grande} → $\lvert$ [redacted] $\rvert$	star for no resize
\norm	\norm{a} → $\lVert a \rVert$	automatic sizing
	\norm\Big{a} → $\Big\lVert a \Big\rVert$	manual sizing
	\norm*{\grande} → $\lVert$ [redacted] $\rVert$	star for no resize
\evaluated	\eval{x}_0^\infty → $x\Big\rvert_0^\infty$	vertical bar for evaluation limits
	\eval(x|_0^\infty → $\left(x\right\rvert_0^\infty$	alternate form
	\eval[x|_0^\infty → $\left[x\right\rvert_0^\infty$	alternate form
	\eval[\venti|_0^\infty → $\left[\text{[redacted]}\right\rvert_0^\infty$	automatic sizing
	\eval*[\venti|_0^\infty → $\left[\text{[redacted]}\right\rvert_0^\infty$	star for no resize
\order	\order{x^2} → $\mathcal{O}\left(x^2\right)$	order symbol; automatic sizing and space handling
	\order\Big{x^2} → $\mathcal{O}\Big(x^2\Big)$	manual sizing
	\order*{\grande} → $\mathcal{O}($ [redacted] $)$	star for no resize

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Vector Notation

You may use `\mathbf` to make bold maths symbols, However, it won't always work. For example, with `\mathbf{\alpha}` you may have α, which is actually not bold. These commands will help provide the correct $\boldsymbol{\alpha}$.

`\vectorbold`	`\vb{a}` → $\mathbf{a}$	upright/no Greek
	`\vb*{a}`, `\vb*{\theta}` → $\boldsymbol{a}$, $\boldsymbol{\theta}$	italic/Greek
`\vectorarrow`	`\va{a}` → $\vec{\mathbf{a}}$	upright/no Greek
	`\va*{a}`, `\va*{\theta}` → $\vec{\boldsymbol{a}}$, $\vec{\boldsymbol{\theta}}$	italic/Greek
`\vectorunit`	`\vu{a}` → $\hat{\mathbf{a}}$	upright/no Greek
	`\vu*{a}`, `\vu*{\theta}` → $\hat{\boldsymbol{a}}$, $\hat{\boldsymbol{\theta}}$	italic/Greek

There are also some shorthand for vector operations.

`\dotproduct`	`\vdot` → $\cdot$ as in $\mathbf{a} \cdot \mathbf{b}$	note: `\dp` is a protected TeX primitive
`\crossproduct`	`\cross` → $\times$ as in $\mathbf{a} \times \mathbf{b}$	alternate name
	`\cp` → $\times$ as in $\mathbf{a} \times \mathbf{b}$	shorthand name

The default del (nabla) symbol ∇ used in `physics` vector notation can be switched to appear with an arrow $\vec{\nabla}$ by including the option `arrowdel` in the document preamble → `\usepackage[arrowdel]{physics}`.

`\divergence`

`\div` → $\nabla\cdot$

note: `amsmath` symbol ÷ renamed `\divisionsymbol`

`\div{\vb{a}}` → $\nabla \cdot \mathbf{a}$ default mode

`\div(\vb{a}+\tall)` → $\nabla \cdot \left(\mathbf{a} + \square\right)$ long-form

`\div[\vb{a}+\tall]` → $\nabla \cdot \left[\mathbf{a} + \square\right]$

`\curl`

`\curl` → $\nabla\times$

`\curl{\vb{a}}` → $\nabla \times \mathbf{a}$ default mode

`\curl(\vb{a}+\tall)` → $\nabla \times \left(\mathbf{a} + \square\right)$ long-form

`\curl[\vb{a}+\tall]` → $\nabla \times \left[\mathbf{a} + \square\right]$

`\laplacian`

`\laplacian` → ∇^2

`\laplacian{\Psi}` → $\nabla^2 \Psi$ default mode

`\laplacian(\Psi+\tall)` → $\nabla^2 \left(\Psi + \square\right)$ long-form

`\laplacian[\Psi+\tall]` → $\nabla^2 \left[\Psi + \square\right]$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Operators

The standard set of trig functions is redefined in `physics` to provide automatic braces that behave like `\qty()`. In addition, an optional power argument is provided. This behavior can be switched off by including the option `notrig` in the preamble → `\usepackage[notrig]{physics}`.

For example,

`\sin`	`\sin(\grande)` → $\sin\left(\blacksquare\right)$	automatic braces; old `\sin` renamed `\sine`
	`\sin[2](x)` → $\sin^2(x)$	optional power
	`\sin x` → $\sin x$	can still use without an argument

Similar behavior has also been extended to the following functions:

`\exp(\tall)`	$\exp\left(\blacksquare\right)$		`\exponential`
`\log(\tall)`	$\log\left(\blacksquare\right)$		`\logarithm`
`\ln(\tall)`	$\ln\left(\blacksquare\right)$	old definitions ⇒	`\naturallogarithm`
`\det(\tall)`	$\det\left(\blacksquare\right)$		`\determinant`
`\Pr(\tall)`	$\Pr\left(\blacksquare\right)$		`\Probability`

There are also some new operators:

`\trace` or `\tr`	`\tr\rho` → $\tr\rho$ also `\tr(\tall)` → $\tr\left(\blacksquare\right)$	trace; same bracing as trig functions
`\Trace` or `\Tr`	`\Tr\rho` → $\Tr\rho$	alternate
`\rank`	`\rank M` → $\rank M$	matrix rank
`\erf`	`\erf(x)` → $\erf(x)$	Gauss error function
`\Res`	`\Res[f(z)]` → $\Res[f(z)]$	residue; same bracing as trig functions
`\principalvalue`	`\pv{\int f(z) \dd{z}}` → $\pv{\int f(z)\dd{z}}$ `\PV{\int f(z) \dd{z}}` → $\PV{\int f(z)\dd{z}}$	Cauchy principal value alternate
`\Re`	`\Re{z}` → $\Re{z}$	old `\Re` renamed to `\real` → $\real$
`\Im`	`\Im{z}` → $\Im{z}$	old `\Im` renamed to `\imaginary` → $\imaginary$

Quick Quad Text

This set of commands produces text in math-mode padded by \quad spacing on either side. This is meant to provide a quick way to insert simple words or phrases in a sequence of equations. Each of the following commands includes a starred version which pads the text only on the right side with \quad for use in aligned environments such as cases.

\qqtext	\qq{}	general quick quad text with argument
	\qq{word or phrase} → word or phrase	normal mode; left and right \quad
	\qq*{word or phrase} → word or phrase	starred mode; right \quad only

Some special macros:

\qcomma or \qc →,	right \quad only
\qcc → c.c.	complex conjugate; left and right \quad unless starred \qcc* → c.c.
\qif → if	left and right \quad unless starred \qif* → if

\qthen, \qelse, \qotherwise, \qunless, \qgiven, \qusing, \qassume, \qsince, \qlet, \qfor, \qall, \qeven, \qodd, \qinteger, \qand, \qor, \qas, \qin

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Derivatives

The default differential symbol d which is used in `\differential` and `\derivative` can be switched to an italic form d by including the option `italicdiff` in the preamble → `\usepackage[italicdiff]{physics}`.

`\differential`	`\dd` → d	
	`\dd x` → $\mathrm{d}x$	no spacing (not recommended)
	`\dd{x}` → $\,\mathrm{d}x\,$	automatic spacing based on neighbors
	`\dd[3]{x}` → d^3x	optional power
	`\dd(\cos\theta)` → $\mathrm{d}(\cos\theta)$	long-form; automatic braces
`\derivative`	`\dv{x}` → $\dfrac{\mathrm{d}}{\mathrm{d}x}$	one argument
	`\dv{f}{x}` → $\dfrac{\mathrm{d}f}{\mathrm{d}x}$	two arguments
	`\dv[n]{f}{x}` → $\dfrac{\mathrm{d}^n f}{\mathrm{d}x^n}$	optional power
	`\dv{x}(\grande)` → $\dfrac{\mathrm{d}}{\mathrm{d}x}\left(\blacksquare\right)$	long-form; automatic braces, spacing
	`\dv*{f}{x}` → $\mathrm{d}f/\mathrm{d}x$	inline form using `\flatfrac`

`\partialderivative`

`\pderivative{x}` $\rightarrow \frac{\partial}{\partial x}$	alternate name
`\pdv{x}` $\rightarrow \frac{\partial}{\partial x}$	shorthand name
`\pdv{f}{x}` $\rightarrow \frac{\partial f}{\partial x}$	two arguments
`\pdv[n]{f}{x}` $\rightarrow \frac{\partial^n f}{\partial x^n}$	optional power
`\pdv{x}(\grande)` $\rightarrow \frac{\partial}{\partial x}\left(\blacksquare\right)$	long-form
`\pdv{f}{x}{y}` $\rightarrow \frac{\partial^2 f}{\partial x \partial y}$	mixed partial
`\pdv*{f}{x}` $\rightarrow \partial f/\partial x$	inline form using `\flatfrac`

`\variation`

`\var{F[g(x)]}` $\rightarrow \delta F[g(x)]$	functional variation (works like `\dd`)
`\var(E-TS)` $\rightarrow \delta(E-TS)$	long-form

`\functionalderivative`

`\fdv{g}` $\rightarrow \frac{\delta}{\delta g}$	functional derivative (works like `\dv`)
`\fdv{F}{g}` $\rightarrow \frac{\delta F}{\delta g}$	
`\fdv{V}(E-TS)` $\rightarrow \frac{\delta}{\delta V}(E-TS)$	long-form
`\fdv*{F}{x}` $\rightarrow \delta F/\delta x$	inline form using `\flatfrac`

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

Matrices

The following matrix macros produce unformatted rows and columns of matrix elements for use as separate matrices as well as blocks within larger matrices. For example, the command `\identitymatrix{2}` which has also has the shortcut `\imat{2}` produces the elements of a 2×2 identity matrix $\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}$ without braces or grouping. This allows the command to also be used within another matrix, as in:

Example

```
\begin{equation}
  \begin{pmatrix}
    \imat{2} \\
    a & b
  \end{pmatrix}
\end{equation}
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{pmatrix} \tag{21}$$

To specify elements on the right of left sides of our `\imat{2}` sub-matrix we use the grouping command `\matrixquantity` or `\mqty` to effectively convert `\imat{2}` into a single matrix element of a larger matrix:

Example

```
\begin{equation}
  \begin{pmatrix}
    \mqty{\imat{2}} & \mqty{a\\b} \\
    \mqty{c & d}    & e
  \end{pmatrix}
\end{equation}
```

$$\begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ c & d & e \end{pmatrix} \tag{22}$$

The extra `\mqty` groups were required in this case in order to get the a and b elements to behave as a single element, since `\mqty{\imat{2}}` also acts like a single matrix element (the same can be said of the grouped c and d elements). Finally, the outermost `pmatrix` environment could have also been replaced with the `physics` macro `\mqty()`, allowing the above example to be written on one line:

Example

```
\begin{equation}
  \mqty(
    \mqty{\imat{2}} & \mqty{a\\b} \\
    \mqty{c & d}    & e
  )
\end{equation}
```

$$\begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ c & d & e \end{pmatrix} \tag{23}$$

The matrix commands are listed below:

`\matrixquantity`

`\mqty{a & b \\ c & d}` → $\begin{matrix} a & b \\ c & d \end{matrix}$ groups a set of matrix elements into a single object

`\mqty(a & b \\ c & d)` → $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ parentheses

`\mqty*(a & b \\ c & d)` → $\left(\begin{matrix} a & b \\ c & d \end{matrix}\right)$ alternate parentheses

`\mqty[a & b \\ c & d]` → $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ square brackets

`\mqty|a & b \\ c & d|` → $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ vertical bars

`\pmqty{}` ↔ `\mqty()`
`\Pmqty{}` ↔ `\mqty*()`
`\bmqty{}` ↔ `\mqty[]`
`\vmqty{}` ↔ `\mqty||`
alternative syntax; robust and more LaTeX-friendly

`\smallmatrixquantity`

`\smqty{a & b \\ c & d}` → $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ the `smallmatrix` form of `\mqty`
`\smqty()` or `\spmqty{}` small version of `\mqty()`
`\smqty*()` or `\sPmqty{}` small version of `\mqty*()`
`\smqty[]` or `\sbmqty{}` small version of `\mqty[]`
`\smqty||` or `\svmqty{}` small version of `\mqty||`

`\matrixdeterminant`

`\mdet{a & b \\ c & d}` → $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ matrix determinant

`\smdet{a & b \\ c & d}` → $\left|\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right|$ small matrix determinant

`\identitymatrix`

`\imat{n}` elements of $n \times n$ identity matrix

`\smqty(\imat{3})` → $\left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$ formatted with `\mqty` or `\smqty`

`\xmatrix` `\xmat{x}{n}{m}` — elements of $n \times m$ matrix filled with x

`\smqty(\xmat{1}{2}{3})` $\rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ formatted with `\mqty` or `\smqty`

`\smqty(\xmat*{a}{3}{3})` $\rightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ star for element indices

`\smqty(\xmat*{a}{3}{1})` $\rightarrow \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ as a vector with indices

`\smqty(\xmat*{a}{1}{3})` $\rightarrow \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}$

`\zeromatrix` `\zmat{n}{m}` — $n \times m$ matrix filled with zeros

`\smqty(\zmat{2}{2})` $\rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ equivalent to `\xmat{0}{n}{m}`

`\paulimatrix` `\pmat{n}` — n^{th} Pauli matrix, $n \in \{0, 1, 2, 3 \text{ or } x, y, z\}$

`\smqty(\pmat{0})` $\rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

`\smqty(\pmat{1})` $\rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

`\smqty(\pmat{2})` $\rightarrow \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

`\smqty(\pmat{3})` $\rightarrow \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

`\diagonalmatrix` `\dmat{a,b,c,...}` — specify up to eight diagonal or block diagonal elements

`\mqty(\dmat{1,2,3})` $\rightarrow \begin{pmatrix} 1 & & \\ & 2 & \\ & & 3 \end{pmatrix}$

`\mqty(\dmat[0]{1,2})` $\rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ optional argument to fill spaces

`\mqty(\dmat{1,2&3\\4&5})` $\rightarrow \begin{pmatrix} 1 & & \\ & 2 & 3 \\ & 4 & 5 \end{pmatrix}$ enter matrix elements for each block as a single diagonal element

`\antidiagonalmatrix` `\admat{a,b,c,...}` — same as syntax as `\dmat`

`\mqty(\admat{1,2,3})` $\rightarrow \begin{pmatrix} & & 1 \\ & 2 & \\ 3 & & \end{pmatrix}$

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX

8 Useful Maths Packages
- Common Packages
- The physics Package
- The systeme Package
- The gauss Package

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}
```

$$\begin{cases} 2a-3b+4c=2 \\ a+8b+5c=8 \\ -a+2b+c=-5 \end{cases} \tag{24}$$

It also works for subscripts.

Example

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

$$\begin{cases} 4x_1 - x_2 = 3 \\ -x_1 + 5x_2 = -1 \end{cases} \tag{25}$$

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}
```

$$\begin{cases} 2x + 3y = 0 \\ x - z + 9 = 0 \\ 3x - y + z + 2 + 5 - 7 = 0 \end{cases} \tag{26}$$

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}
```

$$\begin{cases} (2+\sqrt{2})x-(1-\sqrt{2})y=1 \\ x+(1+\sqrt{2})y=-1 \end{cases} \tag{27}$$

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.

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

7 Use Maths in LaTeX

8 Useful Maths Packages
- Common Packages
- The physics Package
- The systeme Package
- The gauss Package

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

The gauss Package

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

Command

```
1  \usepackage{gauss}
```

This package provides LaTeX-macros for typesetting operations on a matrix. By an "operation on a matrix" we understand a row operation or a column operation. It is named gauss because Gauss Elimination is a widely used application of matrix operations.

The documentation of the systeme package can be found in http://mirrors.ctan.org/macros/latex/contrib/gauss/gauss-doc.pdf.

Use Maths in LaTeX
Math Expressions
Math Environments
Spacing in Math Mode
Basic Math Commands
Matrices and Arrays
Useful Maths Packages
Common Packages
The physics Package
The systeme Package
The gauss Package

For example, if you are taking VV285 or working with other linear algebra stuffs in LaTeX, you may use the `gmatrix` environment provided by the `gauss` package.

Example

```
\begin{equation}
  \begin{array}{ccc|}
    4 & 2 & -2 \\
    -3 & 1 & 0 \\
    1 & 4 & 2
  \end{array}
  \begin{gmatrix}
    -2 \\ 6 \\ -9
    \rowops
    \swap{0}{2}
    \add[*(3)]{0}{1}
  \end{gmatrix}
\end{equation}
```

$$\begin{array}{ccc|c} 4 & 2 & -2 & -2 \\ -3 & 1 & 0 & 6 \\ 1 & 4 & 2 & -9 \end{array} \begin{array}{l} \leftarrow \\ \\ \leftarrow \end{array} \begin{array}{l} \\ \\ +\,*(3) \end{array} \tag{28}$$

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Lecture IV

Graphs, Tables and Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs
- Include Graphs
- Figures
- Draw Graphs

10 Tables

11 Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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

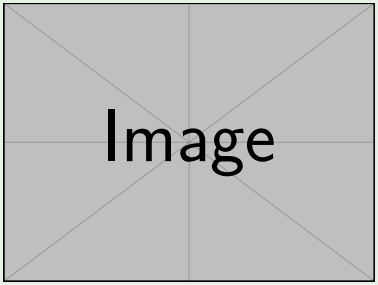
```
\includegraphics[options]{filename}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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

```
1  \includegraphics[width=0.4\textwidth]{example-image}
```



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.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Options of Include Graphs

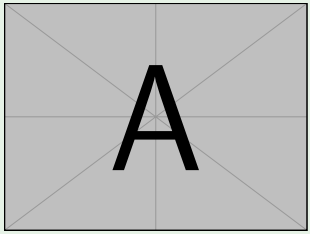
Here some useful `options` are listed:

- `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

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



Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs
- Include Graphs
- Figures
- Draw Graphs

10 Tables

11 Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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)

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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=0.8\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}.
```

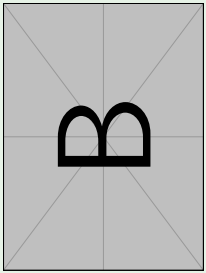


Figure 2: Example Image B rotated by 90 degree.

B was shown in Figure 2.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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}`.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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.

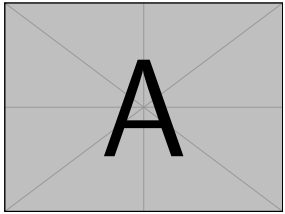
Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

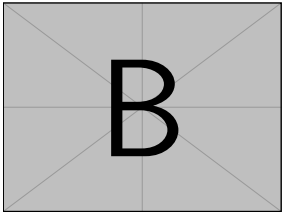
```latex
\begin{figure}
    \centering
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[width=\textwidth]{example-image-a}
        \caption{Example Image A.}
        \label{fig:subcaption-a}
    \end{subfigure}
    ~
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[width=\textwidth]{example-image-b}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

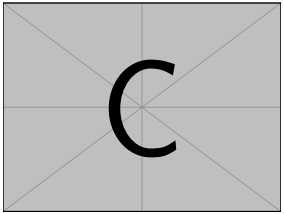
Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing



(a) Example Image A.



(b) Example Image B.



(c) Example Image C.

Figure 3: Example Images

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs
- Include Graphs
- Figures
- Draw Graphs

10 Tables

11 Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The tikz and pgf packages

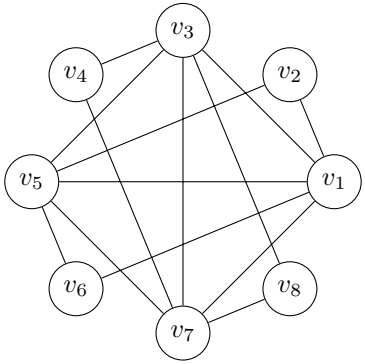
The tikz and pgf packages can help you draw graphs in LaTeX for example:

Example

```
\begin{tikzpicture}[scale=2, bend angle=22.5]
\tikzstyle{every node}=[draw,shape=circle];
\foreach \i in {1,...,8}
{
\path (45*\i-45:1cm) node (v\i) {$v_\i$};
}
\draw
(v1) -- (v2) (v3) -- (v4) (v5) -- (v6) (v7) -- (v8)
(v1) -- (v3) (v3) -- (v5) (v5) -- (v7) (v7) -- (v1)
(v2) -- (v5) (v4) -- (v7) (v6) -- (v1) (v8) -- (v3)
(v1) -- (v5) (v3) -- (v7);
\end{tikzpicture}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

This will generate a simple graph which consists of eight nodes:



There may be a lecture about tikz and pgf in the future. If you are now interested in it, please refer to the pgf manuel by `texdoc tikz` or `texdoc pgf`.

Another example:

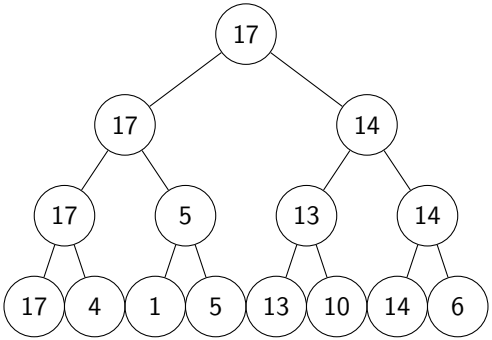
Example

```
\begin{tikzpicture}[scale=0.8]
\tikzstyle{every node}=[draw,shape=circle,minimum size=0.8cm];
\node {17}[sibling distance=4cm]
child { node {17}[sibling distance=2cm]
    child {
        node {17}[sibling distance=1cm]
        child { node {17} }
        child { node {4} }
    }
    child {
        node {5}[sibling distance=1cm]
        child { node {1} }
        child { node {5} }
    }
}
child { node {14}[sibling distance=2cm]
    child {
        node {13}[sibling distance=1cm]
        child { node {13} }
        child { node {10} }
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

```
21        }
22        child {
23            node {14}[sibling distance=1cm]
24            child { node {14} }
25            child { node {6} }
26        }
27    };
28    \end{tikzpicture}
```

This will generate a binary tree:



Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs

10 Tables
- Tabulars
- Tables

11 Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The tabular Environment

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

Command

```
\usepackage{array}
```

Example

```
\begin{tabular}{|l|c|r|}
  \hline
  Title 1 & Title 2 & Title 3 \\
  \hline
  1 & 2 & 3 \\
  \hline
\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.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Column Format

Command

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

`format` can be set as follow:

- `|` - represents a vertical separate line between two columns
- `l` - align left in this column
- `c` - align center in this column
- `r` - align right in this column

Example

`|l|l|l|`

Title 1	Title 2	Title 3
1	2	3

`||c|cc||`

Title 1	Title 2	Title 3
1	2	3

With the help of the `array` package, more formats are available:

- `p{width}` - Equivalent to `\parbox[t]{width}`, vertically aligned bottom
- `b{width}` - Equivalent to `\parbox[b]{width}`, vertically aligned top
- `m{width}` - Equivalent to `\parbox{width}`, vertically aligned middle
- `>{decl.}` - Can be used before a letter option, inserts `decl` before the column.
- `<{decl.}` - Can be used after a letter option, inserts `decl` after the column.

`t` and `b` may be very confusing, but that's how they work in `\parbox`. With these new formats, the columns can be defined more flexibly.

Example

```
1  \begin{tabular}
2  {|p{1.2cm}|b{1.2cm}|m{1.2cm}|}
3    \hline
4    Aligned Bottom & Aligned Top &
5    Aligned Middle \\
6    \hline
7    1 & 2 & 3 \\
8    \hline
9  \end{tabular}
```

Aligned Bottom	Aligned Top	Aligned Middle
1	2	3

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

`t`, `b` and `m` only affect the vertical alignment. If you want to control the width and make the text horizontally centered as well, you can use `>{\centering}` to insert a `\centering` before the text in that column. You can also insert `>{$}` and `<{$}` to generate a column in math mode.

Example

```
\begin{tabular}{|>{\centering}m{2cm}|>{$}b{2cm}<{$}|}
  \hline
  Row of Text &
  \text{Row of Maths} \\
  \hline
  First & x \\
  Second & x^2 \\
  \hline
\end{tabular}
```

Row of Text	Row of Maths
First	x
Second	x^2

If a column type will be used many times, and also very long, you can define a new column type by yourselves. You can use

Command

```
\newcolumntype{new type}{>{some declarations}{old type}<{some more declarations}}
```

If you want to repeat a format for multiple times, you can use `*{num}{format}`. Here's an example of the usage of `\newcolumntype` with multiple columns form.

Example

```
\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{L}{>{$}l<{$}}
\newcolumntype{R}{>{$}r<{$}}

\begin{tabular}{|L| *{2}{C|} R|}
  \hline
  \text{First} & \text{Second} &
  \text{Second} & \text{Third} \\
  \hline
  x & x^2 & x^2 & x^3 \\
  \hline
  y & y^2 & y^2 & y^3 \\
  \hline
\end{tabular}
```

First	Second	Second	Third
x	x^2	x^2	x^3
y	y^2	y^2	y^3

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Horizontal Lines

We usually need horizontal lines in tables. As shown in the examples above, you can add a `\hline` at the beginning of a row.

If you only want to draw a partial line, use `\cline[start-end]`.

Example

```
\begin{tabular}{c|l|c|r}
  \hline\hline
  & Title 1 & Title 2 & Title 3 \\
  \cline{2-4}
  Table & 1 & 2 & 3 \\
  \cline{2-4}
  & 4 & 5 & 6 \\
  \hline\hline
\end{tabular}
```

	Title 1	Title 2	Title 3
Table	1	2	3
	4	5	6

Here we draw a table with a multirow, but it only works with multirows of odd row number. A more convenient method of drawing multirows will be introduced.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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 `|` (eg. `c|`)
- `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.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\centering
\begin{tabular}{|c|c|c|c|c|}
  \hline
  \multirow{4}{*}{Table} & Title 1 & Title 2 & Title 3 & Title 4 \\
  \cline{2-5}
  & \multicolumn{2}{c|}{Text 1} &
  \multicolumn{2}{c|}{\multirow{3}{*}{Text 3}} \\
  \cline{2-3}
  & \multicolumn{2}{c|}{Text 2} & \multicolumn{2}{c|}{} \\
  \cline{2-3}
  & Text 4 & Text 5 & \multicolumn{2}{c|}{} \\
  \hline
\end{tabular}
```

<table>
<tr><td rowspan="4">Table</td><td>Title 1</td><td>Title 2</td><td>Title 3</td><td>Title 4</td></tr>
<tr><td colspan="2">Text 1</td><td colspan="2" rowspan="3">Text 3</td></tr>
<tr><td colspan="2">Text 2</td></tr>
<tr><td>Text 4</td><td>Text 5</td></tr>
</table>

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

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Coloring Rows and Columns

The rows and columns can also be colored, with the `colortbl` package. You may also need the `xcolor` package to define new colors.

Command

```
\usepackage{xcolor}
\usepackage{colortbl}
```

Some commands are provided by these packages

Command

```
\definecolor{name}{system}{definition}
\rowcolor{color}
\columncolor{color}
```

Here `system` can be `rgb/hsb/cmyk/gray`. Please refer the lecture about defining colors.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\definecolor{mygray}{gray}{.9}
\definecolor{barblue}{RGB}{153,204,254}
\centering
\begin{tabular}{|*2{>{\columncolor{mygray}}c|c|}{}}
  \hline\rowcolor{barblue}
  Title 1 & Title 2 & Title 3 & Title 4 \\
  \hline
  Text 1 & Text 2 & Text 3 & Text 4 \\
  \hline
  Text 5 & Text 6 & Text 7 & Text 8 \\
  \hline
\end{tabular}
```

Title 1	Title 2	Title 3	Title 4
Text 1	Text 2	Text 3	Text 4
Text 5	Text 6	Text 7	Text 8

Note that the `\rowcolor` overwrites the `\columncolor`.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Styling Rows and Columns

Styling columns can be easily achieved by prepending styles in the `>{decl.}` introduced before.

However, styling rows is much more complicated. You should only consider using this method when your table is really too large to style them one by one.

First, you may define these in the preamble of your document:

Command

```
\newcolumntype{+}{>{\global\let\currentrowstyle\relax}}
\newcolumntype{^}{>{\currentrowstyle}}
\newcommand{\rowstyle}[1]{\gdef\currentrowstyle{#1}%
  #1\ignorespaces
}
```

Then you should add a `+` before the first column definition and a `^` before any other column definitions. (You can change the symbols `+` and `^` in the definition above.)

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\newcolumntype{+}{>{\global\let\currentrowstyle\relax}}
\newcolumntype{^}{>{\currentrowstyle}}
\newcommand{\rowstyle}[1]{\gdef\currentrowstyle{#1}%
  #1\ignorespaces
}
\centering
\begin{tabular}{|+>{\ttfamily}c|^c|^>{\ttfamily}c|^c|}
  \hline\rowstyle{\bfseries\sffamily}
  Title 1 & Title 2 & Title 3 & Title 4 \\
  \hline
  Text 1 & Text 2 & Text 3 & Text 4 \\
  \hline
  Text 5 & Text 6 & Text 7 & Text 8 \\
  \hline
\end{tabular}
```

Title 1	Title 2	Title 3	Title 4
Text 1	Text 2	Text 3	Text 4
Text 5	Text 6	Text 7	Text 8

Note that the `\rowstyle` also overwrites the column style set in the formats.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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/

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs

10 Tables
- Tabulars
- Tables

11 Code

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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
  \begin{tabular}{format}
    ...
  \end{tabular}
  \caption{caption}
  \label{table:label}
\end{table}
```

The position, caption, label are same as those in the figure environment.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

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.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

figure and table in Two-column Documents

If you are writing a document using two columns (i.e. you started your document with something like `\documentclass[twocolumn]{article}`), you might have noticed that you can't use floating elements that are wider than the width of a column (using a LaTeX notation, wider than `0.5\textwidth`), otherwise you will see the figure or table overlapping with text.

If you really have to use such wide elements, the only solution is to use the "starred" variants of the floating environments:

Command

```
\begin{figure*}[position]
  ...
\end{figure*}
```

```
\begin{table*}[position]
  ...
\end{table*}
```

Those "starred" versions work like the standard ones, but they will be as wide as the page, so you will get no overlapping.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The array Environment

When you use tabular in maths environment, the text format in the tabular won't be italic. However, there is a replacement of tabular, which is the array environment.

Command

```
1  \begin{array}{format}
2    ...
3  \end{array}
```

The options and usages of these two environment are exactly the same.

Though the environment is not provided by the array package (it's built-in one), you are also recommended to use this package for enhancements.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

List of Floats

The figures and tables are all called floats. Captions can be listed at the beginning of a paper or report in a "List of Figures" or a "List of Tables" section with the commands:

Command

```
\listoffigures
\listoftables
```

The caption used for each figure will appear in these lists, along with the figure numbers, and page numbers that they appear on.

The `\caption` command also has an optional parameter, which is used for the List of Tables or List of Figures.

Command

```
\caption[short]{long}
```

Typically the `short` description is for the caption listing, and the `long` description will be placed beside the figure or table.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs

10 Tables

11 Code
- Pseudo Code
- Code Listing

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The algorithm Environment

LaTeX has several packages for typesetting algorithms in form of "pseudocode". They provide stylistic enhancements over a uniform style (i.e., all in typewriter font) so that constructs such as loops or conditionals are visually separated from other text. The pseudocode is usually put in an algorithm environment. Include it by adding the command to your document's preamble.

Command

```
\usepackage{algorithm}
```

Then you can use the algorithm environment, which acts similar as the figure and table environments.

Command

```
\begin{algorithm}[position]
  \caption{caption}
  \label{algorithm:label}
  <the actual pseudocode environment>
\end{algorithm}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The algorithmic Package

One of the packages, the algorithmic, defines the algorithmic environment. Include it by adding the command to your document's preamble.

Command

```
\usepackage{algorithmic}
```

The basic commands are:

Command

```
\STATE <text>
\IF{<condition>} \STATE {<text>} \ELSE \STATE{<text>} \ENDIF
\IF{<condition>} \STATE {<text>} \ELSIF{<condition>} \STATE{<text>} \ENDIF
\FOR{<condition>} \STATE {<text>} \ENDFOR
\FOR{<condition> \TO <condition> } \STATE {<text>} \ENDFOR
\FORALL{<condition>} \STATE{<text>} \ENDFOR
\WHILE{<condition>} \STATE{<text>} \ENDWHILE
\REPEAT \STATE{<text>} \UNTIL{<condition>}
\LOOP \STATE{<text>} \ENDLOOP
\REQUIRE <text>, \ENSURE <text>, \RETURN <text>, \PRINT <text>
\AND, \OR, \XOR, \NOT, \TO, \TRUE, \FALSE, \COMMENT{<text>}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\begin{algorithm}[H]
  \caption{Calculate $y = x^n$}
  \label{algorithm:n-square}
  \begin{algorithmic}
    \REQUIRE $n \geq 0 \vee x \neq 0$
    \ENSURE $y = x^n$
    \STATE $y \leftarrow 1$
    \IF{$n < 0$}
      \STATE $X \leftarrow 1 / x$
      \STATE $N \leftarrow -n$
    \ELSE
      \STATE $X \leftarrow x$
      \STATE $N \leftarrow n$
    \ENDIF
    \WHILE{$N \neq 0$}
      \IF{$N$ is even}
        \STATE $X \leftarrow X \times X$
        \STATE $N \leftarrow N / 2$
      \ELSE[$N$ is odd]
        \STATE $y \leftarrow y \times X$
        \STATE $N \leftarrow N - 1$
      \ENDIF
    \ENDWHILE
  \end{algorithmic}
\end{algorithm}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y \leftarrow 1$
  if $n < 0$ **then**
    $X \leftarrow 1/x$
    $N \leftarrow -n$
  else
    $X \leftarrow x$
    $N \leftarrow n$
  end if
  while $N \neq 0$ **do**
    if N is even **then**
      $X \leftarrow X \times X$
      $N \leftarrow N/2$
    else[N is odd]
      $y \leftarrow y \times X$
      $N \leftarrow N - 1$
    end if
  end while

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The algorithmicx Package

Another package algorithmicx provides more functionalities, but it is not compatible with the algorithmic package. Include it by adding the command to your document's preamble.

Command

```
\usepackage{algpseudocode}
```

Note that \usepackage{algorithmicx} only defines some common macros and it is not enough. Don't insert \usepackage{algorithmic} in this situation.

The main difference of these two packages is that all of the command name are changed, so that only the first letter in a word is capital. For example, \STATE is changed to \State and \ENDFOR is changed to \EndFor.

The command \begin{algorithmic} can be given the optional argument of a positive integer, which if given will cause line numbering to occur at multiples of that integer. E.g. \begin{algorithmic}[5] will enter the algorithmic environment and number every fifth line.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\begin{algorithm}[H]
  \caption{Euclid' s algorithm}
  \label{algorithm:euclid}
  \begin{algorithmic}[1]
    \Procedure{Euclid}{$a,b$}\Comment{The g.c.d. of a and b}
    \State $r\gets a\bmod b$
      \While{$r\not=0$}\Comment{We have the answer if r is 0}
        \State $a\gets b$
        \State $b\gets r$
        \State $r\gets a\bmod b$
      \EndWhile\label{euclidendwhile}
      \State \textbf{return} $b$\Comment{The gcd is b}
    \EndProcedure
  \end{algorithmic}
\end{algorithm}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Algorithm 2 Euclid′ s algorithm

```
1: procedure EUCLID(a, b)                    ▷ The g.c.d. of a and b
2:     r ← a mod b
3:     while r ≠ 0 do                        ▷ We have the answer if r is 0
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     end while
8:     return b                              ▷ The gcd is b
9: end procedure
```

Algorithms can also be listed like figures and tables, by the command:

Command

```
\listofalgorithms
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

9 Graphs

10 Tables

11 Code
- Pseudo Code
- Code Listing

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The verbatim Environment

The default tool to display code in LaTeX is verbatim, which generates an output in monospaced font.

Example

```
1  \begin{verbatim}
2  Text enclosed inside \texttt{verbatim} environment
3  is printed directly
4  and all \LaTeX{} commands are ignored.
5  \end{verbatim}
```

```
Text enclosed inside \texttt{verbatim} environment
is printed directly
and all \LaTeX{} commands are ignored.
```

There's a starred version (verbatim*) whose output is slightly different.

```
Text␣enclosed␣inside␣\texttt{verbatim}␣environment
is␣printed␣directly
and␣all␣\LaTeX{}␣commands␣are␣ignored.
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The verb Command

Verbatim-like text can also be used inline with the command \verb

Example

```
1  In the directory \verb|C:\Windows\system32| you can find a lot of Windows
2  system applications.
3
4  The \verb+\ldots+ command produces \ldots
```

In the directory C:\Windows\system32 you can find a lot of Windows system applications.
The \ldots command produces ...

The command \verb|C:\Windows\system32| prints the text inside the delimiters | in verbatim format. Any character, except letters and *, can be used as delimiter. For instance \verb+\ldots+ uses + as delimiter.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The listings Package

A better form of code listing can be done by the listings package. To use it, simply insert the command in the preamble of your document.

Command

```
\usepackage{listings}
```

It provides a lstlisting environment.

Command

```
\begin{lstlisting}[language=name]
  ...
\end{lstlisting}
```

You can also input source code from file.

Command

```
\lstinputlisting[language=name]{filename}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
1  \begin{lstlisting}[language=Python]
2  import numpy as np
3
4  def incmatrix(genl1,genl2):
5      m = len(genl1)
6      n = len(genl2)
7  \end{lstlisting}
```

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
```

You can add code coloring and styling by some complicated configurations, see the Overleaf tutorial Link.

The documentation of the `listings` package can be found in Link.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The minted Package

All of the code in this lecture are highlighted by the minted package. To use it, simply insert the command in the preamble of your document.

Command

```
\usepackage{minted}
```

This is a very special package, it depends a program out of LaTeX called pygmentize, which is a code highlighting package written in Python.

You can install the package through pip (assuming you have Python 2 or 3 and pip installed) in your terminal:

Command

```
pip install Pygments
```

And then you can examine in your terminal whether pygmentize is your PATH by directly running it. You also need to add an option -shell-escape to your LaTeX compiler because LaTeX need this permission to run other programs on shell.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The minted Environment

You can use the minted environment to insert a block of code in the specific language.

Command

```
\begin{minted}[options]{language}
  ...
\end{minted}
```

You can use the command in the terminal to find the supported languages.

Command

```
pygmentize -L lexers
```

There is also a list of languages on the online document Link. Note that if you want to insert plain text, use the text language which doesn't have any highlight.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

The Inline `minted`

For a single line of source code, you can alternatively use a shorthand notation:

Command

```
\mint[options]{language}|...|
```

Here we use a pair of |, same as the usage of the `\verb` command, which is also an inline verbatim command.

Or you can also use

Command

```
\mintinline[options]{language}|...|
```

Here | can also be replaced with {}, a pair of +, etc., the key is there should not exist the same delimiter inside the code.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Input File with `minted`

When you have a source code file alone, you can use the command to input the file.

Command

```
\inputminted[options]{language}{filename}
```

There are some commonly used options (not only for this command):

- `linenos` - Turn on line numbers
- `breaklines` - Automatically break long lines in `minted` environment and `\mint`, and wrap longer lines in `\mintinline`.
- `fontsize` - The size of the font to use, as a size command, e.g. `fontsize=\footnotesize`.
- `tabsize` - The number of spaces a tab is equivalent to. (default is 8, but often set to 4)
- `firstline` - The first line to show. (default is 1, useful when showing part of a file)
- `lastline` - The last line to show. (default is the last line of the input)

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Using Different Styles

You can use various styles of highlighting scheme provided by `pygmentize`.

Command

```
\usemintedstyle{name}
```

You can use the command in the terminal to find the supported styles.

Command

```
pygmentize -L styles
```

There is also a demo of languages and styles on the online demo Link. The `autumn` style is used in this lecture.

In the end, `XeLaTeX` might be the best LaTeX compiler working with the `minted` package. It also supports typesetting with Chinese, if you meet problems when using the default `pdflatex` compiler, switch into `XeLaTeX` may solve your issues.

The documentation of the `minted` package can be found in Link.

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Custom Floats

If tables and figures are not adequate for your needs, then you always have the option to create your own! Examples of such instances could be source code examples, or maps. For a program float example, one might therefore wish to create a `float` named program. The package `float` is your friend for this task. All commands to set up the new float must be placed in the preamble, and not within the document.

Command

```
\usepackage{float}
\floatstyle{style}
\newfloat{type}{placement}{ext}[outer counter]
\floatname{type}{floatname}
```

The default name that appears at the start of the caption is the type. If you wish to alter this, use `\floatname{type}{floatname}`.

For the `\floatstyle` command, `style` can be:

- `plain` - the normal style for LaTeX floats, but the caption is always below the content.
- `plaintop` - the normal style for LaTeX floats, but the caption is always above the content.
- `boxed` - a box is drawn that surrounds the float, and the caption is printed below.
- `ruled` - the caption appears above the float, with rules immediately above and below. Then the float contents, followed by a final horizontal rule.

For the `\newfloat` command,

- `type` - the new name you wish to call your float, in the example, "program".
- `placement` - t, b, p, or h (as previously described in Placement), where letters enumerate permitted placements.
- `ext` - the file name extension of an auxiliary file for the list of figures (or whatever). LaTeX writes the captions to this file.
- `outer counter` - the presence of this parameter indicates that the counter associated with this new float should depend on outer counter, for example "chapter".

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Example

```
\documentclass{article}
\usepackage{float}
\floatstyle{ruled}
\newfloat{program}{thp}{lop}
\floatname{program}{Program}

\begin{document}

\begin{program}[H]
  \begin{minted}{java}
class HelloWorldApp {
  public static void main(String[] args) {
    //Display the string
    System.out.println("Hello World!");
  }
}
  \end{minted}
  \caption{The Hello World! program in Java.}
  \label{program:hello-world}
\end{program}

\end{document}
```

Graphs
Include Graphs
Figures
Draw Graphs
Tables
Tabulars
Tables
Code
Pseudo Code
Code Listing

Program 1 The Hello World! program in Java.

```
1  class HelloWorldApp {
2    public static void main(String[] args) {
3      //Display the string
4      System.out.println("Hello World!");
5    }
6  }
```

You can also reference the custom floats.

Example

```
1  Program \ref{program:hello-world} is the Hello World! program in Java.
```

Program 1 is the Hello World! program in Java.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Lecture V

Beamer Slides

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction
- Beamer Document
- Beamer Structure

13 Overlay and Animation

14 Special Structures

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Why beamer?

For LaTeX users, beamer has a number of advantages over PowerPoint or other presentation software: [1]

- If you are creating slides from a larger document, you can simply re-use your LaTeX source material from that document.
- If you need mathematical content in your slides, you have the wealth of mathematical constructs in LaTeX to draw upon.
- The slides you create are multi-platform.

beamer allows you to create slides featuring overlays, animation and so on in LaTeX. You simply insert some calls to beamer macros in your LaTeX source file, and compile it into a pdf file. You can then use a pdf viewer to present your slides.

[1] http://heather.cs.ucdavis.edu/~matloff/beamer.html

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The beamer Class

In order to use beamer, you should use the following command as the first line of your tex document:

Command

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

Then you can create frames with the frame environment or the \frame command in the document body.

Example

```
1  \documentclass[options]{beamer}
2  \begin{document}
3  \begin{frame}
4    some content
5  \end{frame}
6  \frame{some content}
7  \end{document}
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The Title Page

You can add title, author, date and some other information in the preamble of the document, similar to the document class `article`.

Example

```
\title{Introduction to \LaTeX}
\author{Liu Yihao}
\date{\today}
\institute{SJTU-UMJI Technology Department}
```

Then you can use the `\titlepage` command to generate a title page.

Command

```
\begin{frame}
  \titlepage
\end{frame}
```

This is how the first page of this document is generated.

There are some more options for the title page than the ones presented. The next example is a complete one, most of the commands are optional. [1]

Example

```
\documentclass{beamer}
\usetheme{Boadilla}
\usecolortheme{seahorse}

\title[About Beamer] %optional
{About the Beamer class in presentation making}
\subtitle{A short story}
\author[Arthur, Doe] % (optional, for multiple authors)
{A.~B.~Arthur\inst{1} \and J.~Doe\inst{2}}
\institute[VFU] % (optional)
{
  \inst{1} Faculty of Physics \\ Very Famous University
  \and
  \inst{2} Faculty of Chemistry\\ Very Famous University
}
\date[VLC 2013] % (optional)
{Very Large Conference, April 2013}
\logo{\includegraphics[height=1.5cm]{example-image}}
```

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

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The distribution of each element in the title page depends on the theme, which will be introduced later. Here is a description of each command:

- `\title[short title]{title}` - The title of your presentation must be inside braces. You can set an optional shorter title in the square brackets.
- `\subtitle{subtitle}` - Subtitle can be omitted if unnecessary.
- `\author[short author]{author}` and `\institute[short institute]{institute}` - The usages can be referred to the example code. Use the `\inst` command to state the institute of each author if needed.
- `\date[short date]{date}` - You can also use `\today` as a date.
- `\logo{logo}` - You can use text or image, it will be shown on every slide.

The short versions of title, author, institute and date are often used in the headline or footline in the presentation. If omitted, the long versions will be used there.

The complete example of title page is demonstrated on the next page.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

`\frame{\titlepage}`

Outline

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2
2. Section 2
   - Subsection 2.1
     - Subsubsection 2.1.1
   - Subsection 2.2
     - Subsubsection 2.2.1
     - Subsubsection 2.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Frame Title

You may notice that some of the slides have a title (eg., "Frame Title" in this slide). You can use this command to add one:

Command

```
\begin{frame}
  \frametitle{frame title}
\end{frame}
```

Alternatively, you can add the title as an argument of the `frame` environment:

Command

```
\begin{frame}{frame title}

\end{frame}
```

It is worth noting that in `beamer` the basic container is a `frame`. A `frame` is not exactly equivalent to a slide, one `frame` may contain more than one slides.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction
- Beamer Document
- Beamer Structure

13 Overlay and Animation

14 Special Structures

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Sections and Parts

You can also structure a beamer document into sections, subsections and subsubsections. Usually subsubsections are not very useful in small presentations.

Command

```
\section{section}
\subsection{subsection}
\subsubsection{subsubsection}
```

For large presentations or lectures (such as this one), another structure called \part can be used.

Command

```
\part{part}
```

The contents of different parts are often split from each other completely, eg., the counter of figures and tables, the table of contents, etc.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Table of Contents

After dividing your presentation into sections and subsections, you can add a table of contents at the beginning of the document, or before each section, or anywhere.

Command

```
\begin{frame}{Outline}
  \tableofcontents[options]
\end{frame}
```

For example, if the document structure is

Example

```
\section{Section 1}
\subsection{Subsection 1.1}
\subsubsection{Subsubsection 1.1.1}
\subsection{Subsection 1.2}
\subsubsection{Subsubsection 1.2.1}
\subsubsection{Subsubsection 1.2.2}
\section{Section 2}
\subsection{Subsection 2.1}
\subsubsection{Subsubsection 2.1.1}
\subsection{Subsection 2.2}
\subsubsection{Subsubsection 2.2.1}
\subsubsection{Subsubsection 2.2.2}
```

An example of the default table of contents is shown on the next page.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

\frame{\tableofcontents}

Outline

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2
2. Section 2
   - Subsection 2.1
     - Subsubsection 2.1.1
   - Subsection 2.2
     - Subsubsection 2.2.1
     - Subsubsection 2.2.2

By default, all sections, subsections and subsubsections in the current part will be shown in the table and contents. You can also use options to set whether some of the sections or subsections should be shaded, or be hided. The allowed styles are show, shaded and hide. The available options are

- `sectionstyle=<style for current section>/<style for other sections>`
- `subsectionstyle=<style for current subsection>/<style for other subsections in current section>/<style for subsections in other sections>`
- `subsubsectionstyle=<style for current subsubsection>/<style for other subsubsections in current subsection>/<style for subsubsections in other subsections in current section>/<style for subsubsections in other subsections in other sections>`

The later styles can be omitted in each options, in this case, the omitted styles will be set to the last explicit style. For example, these two lines are equivalent:

- `subsectionstyle=show/shaded`
- `subsectionstyle=show/shaded/shaded`

They both cause all subsections except the current subsection in the current section to be shown in a semi-transparent way.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

There are also some shorthands of the options above, you can use them alone, or mix them with any other options.

- `currentsection` - `sectionstyle=show/shaded,subsectionstyle=show/show/shaded`
- `currentsubsection` - `subsectionstyle=show/shaded`
- `hideallsubsections` - `subsectionstyle=hide`
- `hideothersubsections` - `subsectionstyle=show/show/hide`

Some other options include

- `part=<part number>` - shows the table of contents of a specific part.
- `pausesections` - causes a `\pause` command to be issued before each section. This is useful if you wish to show the table of contents in an incremental way.
- `pausesubsections` - causes a `\pause` command to be issued before each subsection.

The `\pause` command can split a frame into two slides, which will be introduced in the next section.

Some examples are shown on the next pages.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

```
\frame{\tableofcontents[pausesection]}
```

Outline (pausesections)

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

```
\frame{\tableofcontents[pausesection]}
```

Outline (pausesections)

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2
2. Section 2
   - Subsection 2.1
     - Subsubsection 2.1.1
   - Subsection 2.2
     - Subsubsection 2.2.1
     - Subsubsection 2.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

```
\frame{\tableofcontents[currentsection]}
```

Outline (Now we are at subsubsection 1.2.1)

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2
2. Section 2
   - Subsection 2.1
     - Subsubsection 2.1.1
   - Subsection 2.2
     - Subsubsection 2.2.1
     - Subsubsection 2.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

```
\frame{\tableofcontents[currentsubsection]}
```

Outline (Now we are at subsubsection 2.2.1)

1. Section 1
   - Subsection 1.1
     - Subsubsection 1.1.1
   - Subsection 1.2
     - Subsubsection 1.2.1
     - Subsubsection 1.2.2
2. Section 2
   - Subsection 2.1
     - Subsubsection 2.1.1
   - Subsection 2.2
     - Subsubsection 2.2.1
     - Subsubsection 2.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

```
\frame{\tableofcontents[sectionstyle=show/shaded,
subsectionstyle=show/shaded/hide,subsubsectionstyle=show/shaded/hide]}
```

Outline (Now we are at subsubsection 2.2.2)

1 Section 1

2 Section 2
- Subsection 2.1
- Subsection 2.2
  - Subsubsection 2.2.1
  - Subsubsection 2.2.2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Bibliography

Like the `article` class, you can use `\cite` to create citations and add a bibliography at the end of the file.

Unfortunately, `bibtex` is not perfectly supported in `beamer`, so usually you need to typeset them by hand.

Example

```
\begin{frame}
  \frametitle{For Further Reading}
  \begin{thebibliography}{Dijkstra, 1982}
    \bibitem[Salomaa, 1973]{Salomaa1973} A.~Salomaa.
    \newblock {\em Formal Languages}. \newblock Academic Press, 1973.
    \bibitem[Dijkstra, 1982]{Dijkstra1982} E.~Dijkstra.
    \newblock Smoothsort, an alternative for sorting in situ.
    \newblock {\em Science of Computer Programming}, 1(3):223--233, 1982.
  \end{thebibliography}
\end{frame}
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Appendix

You can add an appendix by using the `\appendix` command. The command essentially just starts a new part named `\appendixname`. However, it also sets up certain hyperlinks.

All frames, all `\subsection` commands, and all `\section` commands used after this command will not be shown as part of the normal navigation bars.

Example

```
\begin{document}
\frame{\titlepage}
\section*{Outline}
\frame{\tableofcontents}
\section{Main Text}
\frame{Some text}
\section*{Summary}
\frame{Summary text}


\appendix
\section{\appendixname}
\frame{\tableofcontents}
\subsection{Additional material}
\frame{Details}
\frame{Text omitted in main talk.}
\subsection{Even more additional
↪  material}
\frame{More details}
\end{document}
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction

13 Overlay and Animation
- Overlay
- Animation

14 Special Structures

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Simple Overlay

In the introduction, it was mentioned that a frame is not equivalent to a slide.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Simple Overlay

In the introduction, it was mentioned that a frame is not equivalent to a slide.

The simplest way to verify this is to add a simple overlay with the command

Command

```
1  \pause
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Simple Overlay

In the introduction, it was mentioned that a frame is not equivalent to a slide.

The simplest way to verify this is to add a simple overlay with the command

Command

```
\pause
```

A direct example is this frame itself:

Example

```
\begin{frame}{Simple Overlay}
  some contents
  \pause
  some contents
  \pause
  some contents
\end{frame}
```

Note that the page numbers in the bottom right corner of these three slides are the same, the page counter only counts the number of frames.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Overlay Specifications

However, the \pause command only provide a basic support of splitting slides. Using commands with an overlay specification will be more flexible, which means you can have different effects with the same command on different slides.

Overlay specifications can only be written behind certain commands, not every command. \textbf is one of these commands.

Example

```
1  \textbf{This line is bold on all slides.}
2  \textbf<2>{This line is bold only on the second slide.}
```

This line is bold on all slides. This line is bold only on the second slide.

The syntax of (basic) overlay specifications is the following: They are comma-separated lists of slides and ranges. Ranges are specified like this: 2-5, which means slide two through to five.

The start or the end of a range can be omitted. For example, 3- means "slides three, four, five, and so on".

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Overlay Specifications

However, the \pause command only provide a basic support of splitting slides. Using commands with an overlay specification will be more flexible, which means you can have different effects with the same command on different slides.

Overlay specifications can only be written behind certain commands, not every command. \textbf is one of these commands.

Example

```
1  \textbf{This line is bold on all slides.}
2  \textbf<2>{This line is bold only on the second slide.}
```

This line is bold on all slides. This line is bold only on the second slide.

The syntax of (basic) overlay specifications is the following: They are comma-separated lists of slides and ranges. Ranges are specified like this: 2-5, which means slide two through to five.

The start or the end of a range can be omitted. For example, 3- means "slides three, four, five, and so on".

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

For the following commands, adding an overlay specification causes the command to be simply ignored on slides that are not included in the specification: `\textbf`, `\textit`, `\textmd`, `\textnormal`, `\textrm`, `\textsc`, `\textsf`, `\textsl`, `\texttt`, `\textup`, `\emph`, `\color`, `\textcolor`, `\alert`, `\structure`. If a command takes several arguments, like `\color`, the specification should directly follow the command.

Example

```
1  \color<2>[rgb]{1,0,0} This text is red on slides 2, otherwise black.
```

This text is red on slides 2, otherwise black.

For the following commands, adding an overlay specification causes the command to be simply ignored on slides that are not included in the specification: `\textbf`, `\textit`, `\textmd`, `\textnormal`, `\textrm`, `\textsc`, `\textsf`, `\textsl`, `\texttt`, `\textup`, `\emph`, `\color`, `\textcolor`, `\alert`, `\structure`. If a command takes several arguments, like `\color`, the specification should directly follow the command.

Example

```
1  \color<2>[rgb]{1,0,0} This text is red on slides 2, otherwise black.
```

This text is red on slides 2, otherwise black.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The onslide Command

When you want to display or hide some contents on some slides, you can use

Command

```
\onslide(modifier)<overlay specification>{text}
```

Example

```
1  \onslide<1>{(1) onslide}
2  \uncover<1>{(1) uncover}
3
4  \onslide+<2>{(2) onslide+}
5  \visible<2>{(2) visible}
6
7  \onslide*<3>{(3) onslide*}
8  \only<3>{(3) only}
```

(1) onslide (1) uncover

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The onslide Command

When you want to display or hide some contents on some slides, you can use

Command

```
\onslide(modifier)<overlay specification>{text}
```

Example

```
1  \onslide<1>{(1) onslide}
2  \uncover<1>{(1) uncover}
3
4  \onslide+<2>{(2) onslide+}
5  \visible<2>{(2) visible}
6
7  \onslide*<3>{(3) onslide*}
8  \only<3>{(3) only}
```

(2) onslide+ (2) visible

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The onslide Command

When you want to display or hide some contents on some slides, you can use

Command

```
\onslide(modifier)<overlay specification>{text}
```

Example

```
1  \onslide<1>{(1) onslide}
2  \uncover<1>{(1) uncover}
3
4  \onslide+<2>{(2) onslide+}
5  \visible<2>{(2) visible}
6
7  \onslide*<3>{(3) onslide*}
8  \only<3>{(3) only}
```

(3) onslide* (3) only

Some explanations:

- `\onslide` is equivalent to `\uncover`, the text is only shown ("uncovered") on the specified slides. On other slides, the text still occupies space and it is still typeset, but it is not shown or only shown as if transparent (can be set with the command `\setbeamercovered`).
- `\onslide+` is equivalent to `\visible`, it does almost the same as `\uncover`, but it is never transparent, but rather it is not shown at all.
- `\onslide*` is equivalent to `\only`, the text is inserted only into the specified slides. For other slides, the text is simply thrown away. In particular, it occupies no space.

There are also some similar commands:

- `\invisible<overlay specification>{text}` is opposite to `\visible`.
- `\alt<overlay specification>{text}{alternate text}` will show the text on the specified slides and the alternate text on other slides.
- `\temporal<overlay specification>{before text}{text}{after text}` will show the text on the specified slides, the before text on the slides before the interval, and the after text on the slides after the interval.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction

13 Overlay and Animation
- Overlay
- Animation

14 Special Structures

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Zooming

Zooming is necessary when you want to explain a part of a frame (or a very complicated graphic).

Command

```
\framezoom<button overlay specification><zoomed overlay specification>[options](x,y)(w,d)
```

This command should be given somewhere at the beginning of a frame. The button overlay specification is your main slide with the whole graph, and there will be a clickable area which will navigate you to the zoomed slide, specified by the zoomed overlay.

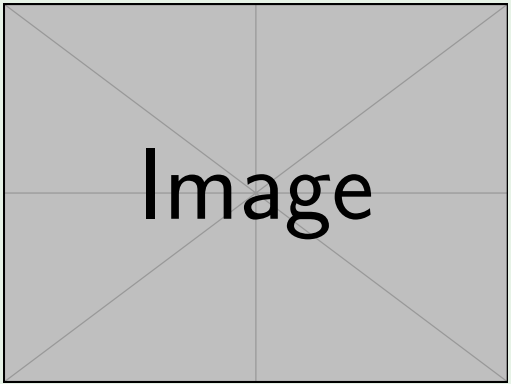
`(x,y)` is the upper left corner of the clickable area. Thus, the location `(0pt,0pt)` is at the beginning of the normal text (which excludes the headline and also the frame title). `(w,d)` is the width and depth (height) of the clickable area.

You can also add `border=width` as `options` so that there will be a border around the clickable area.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Example

```
1 \framezoom<1><2>(0cm,0cm)(4cm,3cm)
2 \framezoom<1><3>(1.5cm,3.5cm)(4cm,3cm)
3 \pgfimage[height=5cm]{example-image}
```



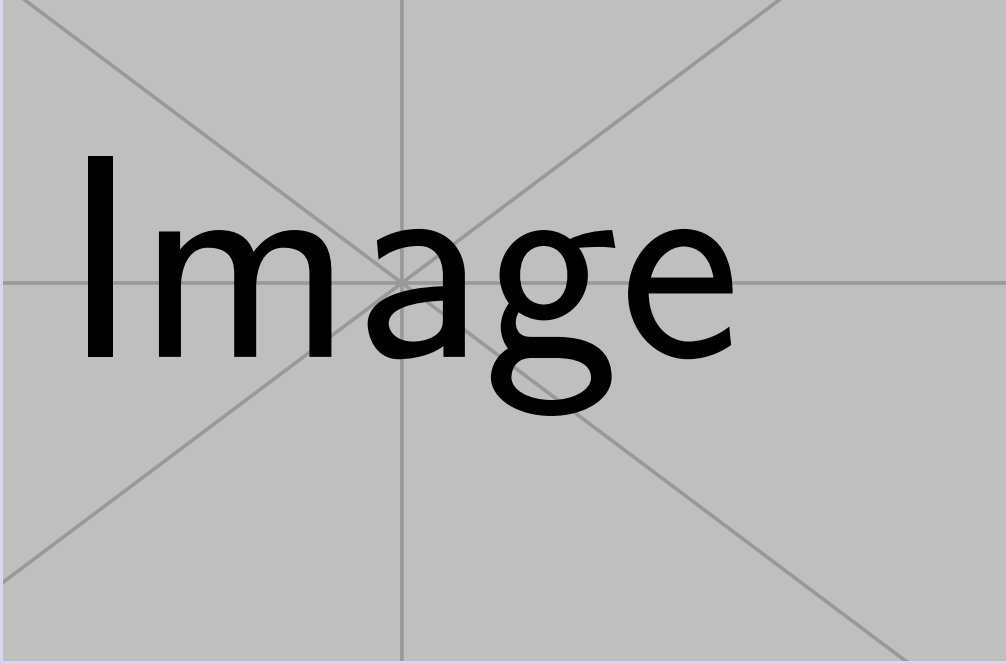
Try to click on the code and the image.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Example

```
1  \framezoom<1><2>(0cm,0cm)(
2  \framezoom<1><3>(1.5cm,3.5
3  \pgfimage[height=5cm]{examp
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame



Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Limitations of Zooming

Though `\framezoom` is very powerful, it still has some limitations.

- You can click on the zoomed slide to jump back to the origin slide, but it only works on Adobe Reader or Acrobat.
- The backend of XeLaTeX will ignore all hyperlinks without text in it, so when compiling with `xelatex`, the area is no longer clickable. You can use `pdflatex` or `lualatex` when you need the zooming feature.

  You can also redefine some macros in the `beamer` package to use `xelatex` with expected behavior, but it will require some knowledge of the inner concept of LaTeX. You can check `lecture.sty` for more details.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The againframe Command

You can use the \againframe command to "continue" frames that you previously started somewhere, but where certain details have been suppressed. You need to add a label for the frame to be repeated.

Command

```
\againframe<overlay specification>[options]{label}
```

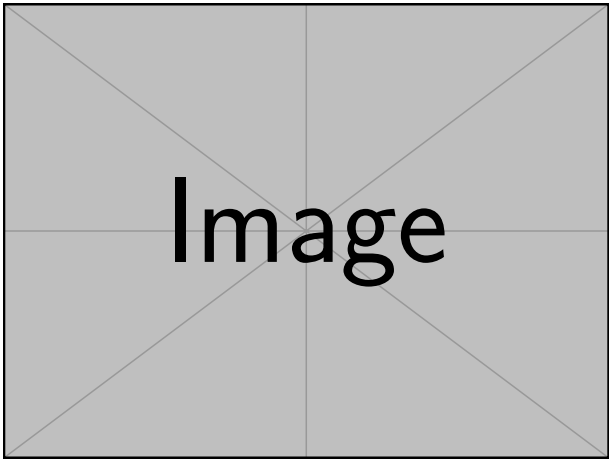
You can use this command together with the \framezoom command to put the zoomed slides at the end of the presentation.

Example

```
\begin{frame}<1>[label=zooms]
\frametitle<1>{A Complicated Picture}
\framezoom<1><2>[border](0cm,0cm)(2cm,1.5cm)
\framezoom<1><3>[border](1cm,2cm)(2cm,1.5cm)
\pgfimage[height=6cm]{example-image}
\end{frame}
% other slides
\againframe<2->[noframenumbering]{zooms}
```

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

A Complicated Picture



Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction

13 Overlay and Animation

14 Special Structures
- Blocks and Columns
- Hyperlinks and Buttons
- Fragile Frame

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The block Environment

Blocks in beamer are based on the tcolorbox, with all styles configured.

Command

```
\begin{block}<action specification>{title}
  % block contents
\end{block}
```

If the action specification is present, the given actions are taken on the specified slides.

There are three types of blocks: block, alertblock and exampleblock. The only difference is their color and style.

Example

```
\begin{block}{Normal Block}
  This is a normal block.
\end{block}
\begin{alertblock}{Alert Block}
  This is an alert block.
\end{alertblock}
\begin{exampleblock}{Example Block}
  This is an example block.
\end{exampleblock}
```

Normal Block

This is a normal block.

Alert Block

This is an alert block.

Example Block

This is an example block.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Predefined block environments

Some other block environment are predefined for ease to use. They are `theorem`, `corollary`, `definition`, `definitions`, `example`, `examples`. Here only a few of them are shown below.

Example

```
1  \begin{theorem}[additional text]
2    The additional text will be in brackets if the option is provided.
3  \end{theorem}
4  \begin{proof}[Proof Name]
5    The default title is ``Proof.'', which can be replaced by the option.
6  \end{proof}
```

Theorem (additional text)

The additional text will be in brackets if the option is provided.

Proof Name.

The default title is "Proof.", which can be replaced by the option. □

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The columns Environment

In an article class, minipage is often used to split contents into multiple columns; In beamer, you can use the columns environment as an alternative.

Command

```
\begin{columns}[options]
  \begin{column}[placement]{width}
    % contents
  \end{column}
  \column[placement]{width}{...}
\end{columns}
```

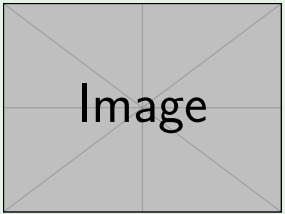
For the options in the columns environment,

- b, c and t - will cause the columns to be vertically aligned bottom, center and top.
- T - similar to t, if strange things happen in t, try this option.
- totalwidth=width - will cause the columns to occupy not the whole page width, but only width.

Example

```
\begin{columns}[c]
  \begin{column}{0.5\textwidth}
    \begin{center}
      The first line \\
      The second line
    \end{center}
  \end{column}
  \column{0.5\textwidth}{
    \includegraphics[width=0.6\textwidth]{example-image}
  }
\end{columns}
```

The first line
The second line



You should place only `column` environments or `\column` commands in the `columns` environment.

For the `placement` in the `column` environment or the `\column` command, you can overwrite the `b`, `c`, `t` and `T` in the outer `columns` environment. The default of `placement` is `t` if not specified.

The `width` is the same as other `width` in LaTeX. For example, you can use `5cm`, or `\textwidth`.

Generally speaking, there are few differences between `minipage` and `columns`, but the `columns` environment has a more user-friendly structure, and overlap is supported better in it. Despite which do you prefer, choosing one of them and sticking to it throughout a presentation is suggested.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Footnote in column

Using footnotes is usually not a good idea. They disrupt the flow of reading.

When you really need it, you can use the `\footnote` command, which is slightly different from common LaTeX.

Command

```
\footnote<overlay specification>[options]{text}
```

As usual, you can give a number as `options`, which will cause the footnote to use that number.

You can also add a `frame` as `options` so that the footnote will be shown at the bottom of the frame. This is normally the default behavior anyway, but in `minipage`, `columns` and certain blocks it makes a difference.

In a `minipage` or `column`, the footnote is usually shown as part of the minipage rather than as part of the frame.

Example

```
\begin{columns}[c,totalwidth=0.9\textwidth]
  \begin{column}{0.3\textwidth}
    The first line \footnote[frame,1]{footnote 1}
  \end{column}
  \begin{column}{0.3\textwidth}
    The second line \footnote[frame,2]{footnote 2}
  \end{column}
  \begin{column}{0.3\textwidth}
    The third line \footnote[3]{footnote 3}
  \end{column}
\end{columns}
```

The first line [1]

The second line [2]

The third line [c]

[c] footnote 3

As you can see, placing footnote at the bottom of a `column` is not a good idea, so using the `frame` option is preferred in most situations.

[1] footnote 1
[2] footnote 2

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction

13 Overlay and Animation

14 Special Structures
- Blocks and Columns
- Hyperlinks and Buttons
- Fragile Frame

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Hyperlinks and Buttons

Here is a simple three-step workflow of how to create hyperlinks and buttons in your slides:

1. You specify a target using the command `\hypertarget` or (easier) the command `\label`. In some cases, see below, this step may be skipped.
2. You render the button using `\beamerbutton` or a similar command. This will render the button, but clicking it will not yet have any effect.
3. You put the button inside a `\hyperlink` command. Now clicking it will jump to the target of the link.

The beamerbutton command

Command

\beamerbutton{text}

There are four types of buttons, the suggested usages of them are:

- \beamerbutton - a normal button
- \beamergotobutton - jump to another area of the presentation
- \beamerskipbutton - skip a well-defined part
- \beamerreturnbutton - return back to a previous part

Example

```
1  \beamerbutton{normal}   \beamergotobutton{goto}
2  \beamerskipbutton{skip} \beamerreturnbutton{return}
```

normal goto skip return

The color of the buttons will follow the current color of the block.

normal goto skip return

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The hyperlink command

Command

```
\hyperlink<overlay specification>{target}{text}
```

If the overlay specification is present, the hyperlink (including the text) is completely suppressed on the non-specified slides.

You will jump to the target you defined before with a \hypertarget command.

You can also use a LaTeX command from the hyperref package as a target, e.g., \href.

Example

```
1  You can find the source of the slides on
2  \href{https://github.com/SJTU-UMJI-Tech/LaTeX}{\beamerbutton{GitHub}}.
```

You can find the source of the slides on GitHub.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The hypertarget command

Command

`\hypertarget<overlay specification>{target}{text}`

If the overlay specification is present, the text is the target for hyper jumps only on the specified slide. On all other slides, the text is shown normally.

Example

```
1  \begin{itemize}
2    \item<1-> First item.
3    \item<2-> Second item.
4  \end{itemize}
5  \hyperlink{jumptosecond}{\beamergotobutton{Jump to second slide}}
6  \hypertarget<2>{jumptosecond}{}
```

- First item.

Jump to second slide

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

The hypertarget command

Command

`\hypertarget<overlay specification>{target}{text}`

If the overlay specification is present, the text is the target for hyper jumps only on the specified slide. On all other slides, the text is shown normally.

Example

```
1  \begin{itemize}
2    \item<1-> First item.
3    \item<2-> Second item.
4  \end{itemize}
5  \hyperlink{jumptosecond}{\beamergotobutton{Jump to second slide}}
6  \hypertarget<2>{jumptosecond}{}
```

- First item.
- Second item.

Jump to second slide

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

12 Introduction

13 Overlay and Animation

14 Special Structures
- Blocks and Columns
- Hyperlinks and Buttons
- Fragile Frame

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Fragile Frame

When you need to insert any verbatim (such as `minted`) in a frame, you have to add the option `[fragile]`, and the `\end{frame}` must be alone on a single line (except for any leading whitespace).

Example

```
\begin{frame}[fragile]
  \begin{minted}{latex}
\documentclass{beamer}
  \end{minted}
\end{frame}
```

Using this option will cause the frame contents to be written to an external file and then read back.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Frame in Fragile Frame

A more tricky situation is that you want to put `\begin{frame}` and `\end{frame}` inside a verbatim environment (like this lecture).

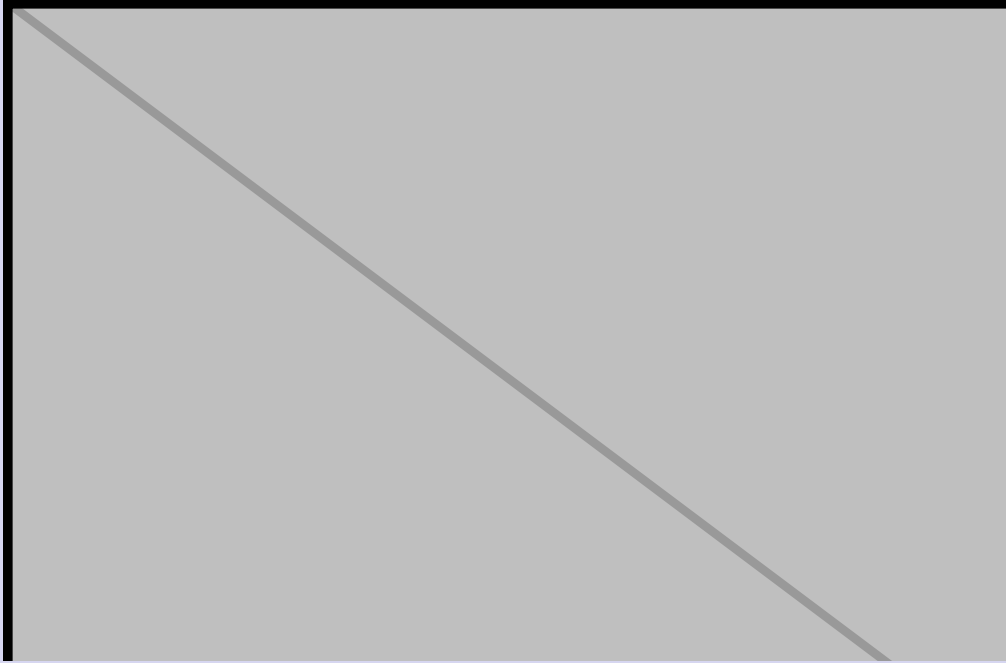
One possible solution is to define a new environment so that the new environment name won't conflict with the content in the verbatim environment.

Example

```
1 \newenvironment{fragileframe}%
2 {\begin{frame}[fragile,environment=fragileframe]}%
3 {\end{frame}}
```

Then you can use `\begin{fragileframe}` to substitute `\begin{frame}[fragile]` for all fragile frames.

Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame



Introduction
Beamer Document
Beamer Structure
Overlay and Animation
Overlay
Animation
Special Structures
Blocks and Columns
Hyperlinks and Buttons
Fragile Frame

Lecture

Appendix

Appendix
Symbol table
Package List
Reference

15 Appendix

- Symbol table
- Package List
- Reference

15 Appendix
- Symbol table
- Package List
- Reference

Symbol tables from lshort

Liu Yihao

Appendix
Symbol table
Package List
Reference

The following tables demonstrate all the symbols normally accessible from math mode. Note that some tables show symbols only accessible after loading the amssymb package in the preamble of your document[1]. If the $\mathcal{AMS}$ package and fonts are not installed on your system, have a look at CTAN:pkg/amslatex. An even more comprehensive list of symbols can be found at CTAN:info/symbols/comprehensive.

Table 1: Math Mode Accents.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{AAA}$	`\widehat{AAA}`
$\acute{a}$	`\acute{a}`	$\breve{a}$	`\breve{a}`	$\widetilde{AAA}$	`\widetilde{AAA}`
$\mathring{a}$	`\mathring{a}`				

[1] The tables were derived from `symbols.tex` by David Carlisle and subsequently changed extensively as suggested by Josef Tkadlec.

Table 2: Greek Letters.

There is no uppercase of some of the letters like \Alpha, \Beta and so on, because they look the same as normal roman letters: A, B...

α	\alpha	θ	\theta	o	o	υ	\upsilon
β	\beta	ϑ	\vartheta	π	\pi	ϕ	\phi
γ	\gamma	ι	\iota	ϖ	\varpi	φ	\varphi
δ	\delta	κ	\kappa	ρ	\rho	χ	\chi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	ψ	\psi
ε	\varepsilon	μ	\mu	σ	\sigma	ω	\omega
ζ	\zeta	ν	\nu	ς	\varsigma		
η	\eta	ξ	\xi	τ	\tau		
Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Table 3: Binary Relations.

You can negate the following symbols by prefixing them with a \not command.

$<$	<	$>$	>	$=$	=
$\leq$	\leq or \le	$\geq$	\geq or \ge	$\equiv$	\equiv
$\ll$	\ll	$\gg$	\gg	$\doteq$	\doteq
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong
$\sqsubset$	\sqsubset [a]	$\sqsupset$	\sqsupset [a]	$\bowtie$	\Join [a]
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\bowtie$	\bowtie
$\in$	\in	$\ni$	\ni , \owns	$\propto$	\propto
$\vdash$	\vdash	$\dashv$	\dashv	$\models$	\models
$\mid$	\mid	$\parallel$	\parallel	$\perp$	\perp
$\smile$	\smile	$\frown$	\frown	$\asymp$	\asymp
:	:	$\notin$	\notin	$\neq$	\neq or \ne

[a] Use the latexsym package to access this symbol

Table 4: Binary Operators.

$+$	+	$-$	-		
$\pm$	\pm	$\mp$	\mp	$\triangleleft$	\triangleleft
$\cdot$	\cdot	$\nabla\cdot$	\div	$\triangleright$	\triangleright
$\times$	\times	$\setminus$	\setminus	$\star$	\star
$\cup$	\cup	$\cap$	\cap	$*$	\ast
$\sqcup$	\sqcup	$\sqcap$	\sqcap	$\circ$	\circ
$\vee$	\vee , \lor	$\wedge$	\wedge , \land	$\bullet$	\bullet
$\oplus$	\oplus	$\ominus$	\ominus	$\diamond$	\diamond
$\odot$	\odot	$\oslash$	\oslash	$\uplus$	\uplus
$\otimes$	\otimes	$\bigcirc$	\bigcirc	$\amalg$	\amalg
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\dagger$	\dagger
$\lhd$	\lhd ^{a}	$\rhd$	\rhd ^{a}	$\ddagger$	\ddagger
$\unlhd$	\unlhd ^{a}	$\unrhd$	\unrhd ^{a}	$\wr$	\wr

Table 5: BIG Operators.

$\sum$	\sum	$\bigcup$	\bigcup	$\bigvee$	\bigvee
$\prod$	\prod	$\bigcap$	\bigcap	$\bigwedge$	\bigwedge
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\biguplus$	\biguplus
$\int$	\int	$\oint$	\oint	$\bigodot$	\bigodot
$\bigoplus$	\bigoplus	$\bigotimes$	\bigotimes		

Table 6: Arrows as Accents.

$\overrightarrow{AB}$	\overrightarrow{AB}	$\underrightarrow{AB}$	\underrightarrow{AB}
$\overleftarrow{AB}$	\overleftarrow{AB}	$\underleftarrow{AB}$	\underleftarrow{AB}
$\overleftrightarrow{AB}$	\overleftrightarrow{AB}	$\underleftrightarrow{AB}$	\underleftrightarrow{AB}

Table 7: Arrows.

$\leftarrow$	\leftarrow or \gets	$\longleftarrow$	\longleftarrow
$\rightarrow$	\rightarrow or \to	$\longrightarrow$	\longrightarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown
$\rightleftharpoons$	\rightleftharpoons	$\iff$	\iff (bigger spaces)
$\uparrow$	\uparrow	$\downarrow$	\downarrow
$\updownarrow$	\updownarrow	$\Uparrow$	\Uparrow
$\Downarrow$	\Downarrow	$\Updownarrow$	\Updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow
$\swarrow$	\swarrow	$\nwarrow$	\nwarrow
$\rightsquigarrow$	\leadsto [a]		

[a] Use the latexsym package to access this symbol

Table 8: Delimiters.

(	(	)	)	↑	\uparrow
[	[or \lbrack	]	] or \rbrack	↓	\downarrow
{	\{ or \lbrace	}	\} or \rbrace	↕	\updownarrow
⟨	\langle	⟩	\rangle	⇑	\Uparrow
∣	| or \vert	∥	\\| or \Vert	⇓	\Downarrow
/	/	\	\backslash	⇕	\Updownarrow
⌊	\lfloor	⌋	\rfloor		
⌉	\rceil	⌈	\lceil		

Table 9: Large Delimiters.

⎧	\lgroup	⎫	\rgroup	⎰	\lmoustache
⎪	\arrowvert	‖	\Arrowvert	⎪	\bracevert
⎩	\rmoustache				

Table 10: Miscellaneous Symbols.

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$ [a]	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$ [a]	\Box [a]	$\Diamond$ [a]	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a] Use the latexsym package to access this symbol

Table 11: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

Table 12: $\mathcal{AMS}$ Delimiters.

⌜	\ulcorner	⌝	\urcorner	⌞	\llcorner	⌟	\lrcorner
|	\lvert	|	\rvert	‖	\lVert	‖	\rVert

Table 13: $\mathcal{AMS}$ Greek and Hebrew.

$\digamma$	\digamma	$\varkappa$	\varkappa	ℶ	\beth	ℷ	\gimel	ℸ	\daleth

Table 14: Math Alphabets.

Example	Command	Required package
ABCDEabcde1234	`\mathrm{ABCDE abcde 1234}`	
$ABCDEabcde1234$	`\mathit{ABCDE abcde 1234}`	
$ABCDEabcde1234$	`\mathnormal{ABCDE abcde 1234}`	
$\mathcal{ABCDE}\dashv\lfloor\rfloor\lceil\rceil\infty\in\ni\triangle$	`\mathcal{ABCDE abcde 1234}`	
$\mathscr{ABCDE}$	`\mathscr{ABCDE abcde 1234}`	mathrsfs
$\mathfrak{ABCDEabcde1234}$	`\mathfrak{ABCDE abcde 1234}`	amsfonts or amssymb
$\mathbb{ABCDE}\nexists\nexists\nexists\nexists$	`\mathbb{ABCDE abcde 1234}`	amsfonts or amssymb

Table 15: $\mathcal{AMS}$ Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot		
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\doublecup	$\Cap$	\doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\Barwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\intercal$	\intercal	$\circledast$	\circledast	$\rightthreetimes$	\rightthreetimes
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge	$\leftthreetimes$	\leftthreetimes

Table 16: $\mathcal{AMS}$ Binary Relations.

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox

Table 17: $\mathcal{AMS}$ Binary Relations. (... continue)

$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\shortparallel$	\shortparallel	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\blacktriangleleft$	\blacktriangleleft	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\vartriangleright$	\vartriangleright	$\because$	\because	$\varpropto$	\varpropto
$\blacktriangleright$	\blacktriangleright	$\Subset$	\Subset	$\between$	\between
$\trianglerighteq$	\trianglerighteq	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\shortmid$	\shortmid	$\smallsmile$	\smallsmile
$\trianglelefteq$	\trianglelefteq	$\therefore$	\therefore	$\sqsubset$	\sqsubset

Table 18: $\mathcal{AMS}$ Arrows.

⇠	\dashleftarrow	⇢	\dashrightarrow
⇇	\leftleftarrows	⇉	\rightrightarrows
⇆	\leftrightarrows	⇄	\rightleftarrows
⇚	\Lleftarrow	⇛	\Rrightarrow
↞	\twoheadleftarrow	↠	\twoheadrightarrow
↢	\leftarrowtail	↣	\rightarrowtail
⇋	\leftrightharpoons	⇌	\rightleftharpoons
↰	\Lsh	↱	\Rsh
↫	\looparrowleft	↬	\looparrowright
↶	\curvearrowleft	↷	\curvearrowright
↺	\circlearrowleft	↻	\circlearrowright
⊸	\multimap	⇈	\upuparrows
⇊	\downdownarrows	↿	\upharpoonleft
↾	\upharpoonright	⇂	\downharpoonright
⇝	\rightsquigarrow	↭	\leftrightsquigarrow

Table 19: $\mathcal{AMS}$ Negated Binary Relations and Arrows.

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Table 20: $\mathcal{AMS}$ Miscellaneous.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	Ⓢ	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle		
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\sphericalangle$	\sphericalangle	$\mho$	\mho

15 Appendix
- Symbol table
- Package List
- Reference

Package List

These are the packages you are most likely to use in daily LaTeX writing.

- geometry
- amsmath
- amssymb
- amsfonts
- multicol
- multirow
- tabu
- graphicx
- subfigure
- hyperref
- ulem
- ctex
- enumerate
- latexsym
- tikz
- listings

Appendix
Symbol table
Package List
Reference

15 Appendix

- Symbol table
- Package List
- Reference

Bibliography

- LaTeX 入门, 刘海洋, 电子工业出版社, 2013.6, ISBN 978-7-121-20208-7
- The Not So Short Introduction to LaTeX 2_ε, Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl, Version 5.05, July 18, 2015 (http://www.ctan.org/tex-archive/info/lshort/english/)
- Introduction to LaTeX, David Reid (https://wenku.baidu.com/view/f08fbdf24693daef5ef73d23.html)

Contributors

This LaTeX beamer slide is contributed to

- Liu Yihao (https://github.com/tc-imba)
- Zhou Yanjun (https://github.com/AuroraZK)
- Zhang Yifei (https://github.com/zhangyifei-chelsea)

For LaTeX lectures of the JI Technology Department.
For all students in JI as a reference in report/homework writing.

This is a long-term maintained project on GitHub, if you have any suggestions, make an issue on it, PRs are welcomed as well.