

# Introduction to $\text{\LaTeX}$

Writing papers the right way

---

**Xu Cheng and Miguel Xochicale** (🐦@\_mxochicale @xu-cheng 🐙@mxochicale @xu-cheng)

Month Day, Year; HourMinute

Event title

School of Eng

University of X, Earth



This work is licensed under a Creative Commons “Attribution 4.0 International” license.

Get source of this slides and example document from <https://github.com/free-cortex/framework>.



1. Getting Started with  $\text{\LaTeX}$
2. A Basic Document
3. Typesetting Text
4. Advanced Usages

# Getting Started with $\text{\LaTeX}$

---

# Introduction

- $\text{\LaTeX}$  is a document preparation system and document markup language.
- It can be used to typeset articles, books, slides, posters, even graphics.
- **Pros:**
  - It separates presentation/format from contents.
  - Since the source codes are plaintext, it works well with version control system such as git.
  - Highly customizable through various of packages.
- **Cons:**
  - There is no graphic interface to support WYSIWYG style editing.
  - Not suitable to produce unstructured documents.

# Installation

- Windows/Linux

- TeXLive <https://www.tug.org/texlive/>
- Online installer:
  - Windows  
<http://mirror.ctan.org/systems/texlive/tlnet/install-tl-windows.exe>
  - Linux  
<http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz>
- Offline ISO file: <http://mirror.ctan.org/systems/texlive/Images/>

- Mac

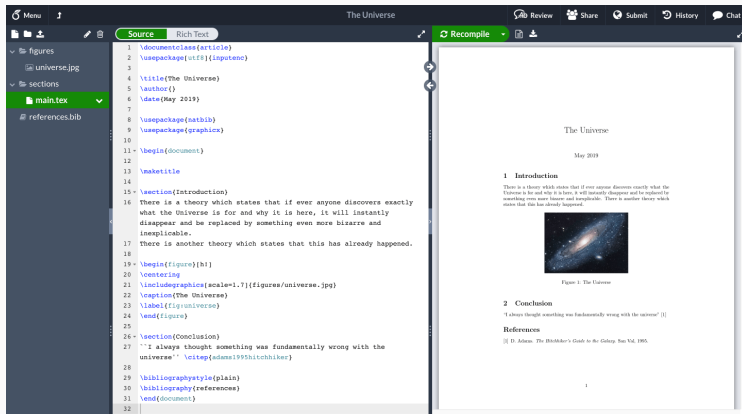
- MacTeX <http://www.tug.org/mactex/>
- Or install through Homebrew (<https://brew.sh>)

```
# Install Homebrew
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
# Install MacTeX
brew cask install mactex
```

- TeXLive/MacTeX release major updates around May each year.  
It is recommended to uninstall the old version and install the new version annually.

- L<sup>A</sup>T<sub>E</sub>X source codes are plaintext. So you can use any editor you like.
- **Visual Studio Code** [Recommend]
  - <https://code.visualstudio.com>
  - LaTeX Workshop <https://github.com/James-Yu/LaTeX-Workshop>
  - Code Spell Checker <https://github.com/streetsidesoftware/vscode-spell-checker>
- **Vim/Neovim**
  - <https://www.vim.org> | <https://neovim.io>
  - Vimtex <https://github.com/lervag/vimtex>
- **Emacs**
  - <https://www.gnu.org/s/emacs>
  - AUCTeX <https://www.gnu.org/software/auctex>
- **TeXstudio**
  - <https://www.texstudio.org>

- **Overleaf** (<https://www.overleaf.com/>) is a online, collaborative LaTeX editor
- Free for personal use
- \$15/month to share project among up to 10 collaborators



# A Basic Document

---



# Hello, $\text{\LaTeX}$ !

- Create `hello.tex` file with following content.

```
% this is hello.tex
\documentclass{article}
\begin{document}
Hello, \LaTeX!
\end{document}
```

- Compile it
  - Click the build button in your  $\text{\LaTeX}$  editor/IDE
  - OR using command line: `latexmk -pdf hello`
- Open `hello.pdf` to preview the result



Compile  $\text{\LaTeX}$  Project in VSCode

## Example of A Complex Document

- Download the source code from <https://github.com/xu-cheng/latex-tutorial/archive/master.zip>
- The example document is located in the `example` folder. It contains:
  - `main.tex` The main tex source
  - `preamble.tex` A subfile to store format definitions
  - `tikz-example.tex` A figure drawn using tikz
  - `ref.bib` A database of references
- Use `latexmk -pdf main` to compile the document
- Access the same example in Overleaf:  
<https://www.overleaf.com/read/qsthqbjphhrz>

# Comment, Command and Environment

- `%` starts a comment. e.g. `% this is hello.tex`
- `\` starts a command.

```
\command % a command
\command{} % also a command
\command{arg} % a command with an argument
\command{arg1}{arg2} % a command with multiple arguments
\command[opt arg]{arg} % [] is for optional argument
```

- `\begin{} ... \end{}`  denotes an environment

```
\begin{envname}
  inside the environment
\end{envname}
% LaTeX environment can take arguments
\begin{envname}{arg} \end{envname}
\begin{envname}[opt arg]{arg} \end{envname}
```

# Source File Structure

- A document starts with `\documentclass{...}` command to specify the template
- Common templates include:
  - **article**
  - **letter**
  - **acmart** (ACM template)
  - **book**
  - **beamer** (slides)
  - **IEEEtrans** (IEEE template)
  - **report**
  - **standalone** (graphics)
- Template class can accept options, e.g. `\documentclass[a4paper,10pt]{article}`

## Class Options for **article**, **report**, **book**, **letter**

|  |   |
|--|---|
| <code>10pt, 11pt, 12pt</code>          | Set font size.  |
| <code>a4paper, letterpaper, ...</code> | Defines the paper size.   |
| <code>fleqn</code>                     | Typesets displayed formulae left-aligned instead of centred.                        |
| <code>leqno</code>                     | Places the numbering of formulae on the left hand side instead of the right.        |
| <code>titlepage, notitlepage</code>    | Specifies whether a new page should be started after the document title or not.     |
| <code>onecolumn, twocolumn</code>      | Typeset the document in one column or two columns.                                  |
| <code>twoside, oneside</code>          | Specifies whether double or single sided output should be generated.                |
| <code>landscape</code>                 | Changes the layout of the document to print in landscape mode.                      |
| <code>openright, openany</code>        | Makes chapters begin either only on right hand pages or on the next page available. |

# Source File Structure

- The region after `\documentclass` and before `\begin{document}` is called *preamble*.
- You can load packages and define format of the document here, e.g. `\usepackage{amsmath}`
- Package can be loaded with options, e.g. `\usepackage[style=ieee]{biblatex}`
- To find the package document:
  - Run `texdoc <pkg_name>` in command line
  - <http://www.texdoc.net>
- You start the body of the text with `\begin{document}`.
- Finally, `\end{document}` denotes the end of the document.

# Typesetting Text

---

# Syntax

- The main body of  $\text{\LaTeX}$  code is plain text.
- $\text{\LaTeX}$  treats contiguous spaces or a single linebreak as a single space. It starts a new paragraph after empty lines.

It does not matter whether  
you enter one or several  
spaces            after a word.

An empty line starts a new  
paragraph.

It does not matter whether you  
enter one or several spaces after a  
word.

An empty line starts a new para-  
graph.

- `\\` or `\newline` starts a new line without starting a new paragraph.

# Special Characters and Symbols

- Certain characters are reserved, you need to use escape command to typeset them.

```
\# \$ \% \^{} \& \_ \{ \} \~{}  
\textbackslash
```

# \$ % ^ & \_ { } ~ \

- ``text'` and ```text''` typeset ‘single quoted text’ and “double quoted text”
- There are four kinds of dashes
  - hyphen: `-`, e.g. part-time
  - en-dash: `--`, e.g. Pages 1–10
  - em-dash: `---`, e.g. yes—or no?
  - minus sign: `-` inside math environment, e.g.  $-1$
- Use `\ldots` instead of `...` to typeset ellipsis, e.g. a, b, c, ...



# Font Face & Size

|                           |                   |                               |                  |
|---------------------------|-------------------|-------------------------------|------------------|
| <code>\textrm{...}</code> | roman             | <code>\textsf{...}</code>     | sans serif       |
| <code>\texttt{...}</code> | typewriter        |                               |                  |
| <code>\textmd{...}</code> | medium            | <code>\textbf{...}</code>     | <b>bold face</b> |
| <code>\textup{...}</code> | upright           | <code>\textit{...}</code>     | <i>italic</i>    |
| <code>\textsl{...}</code> | <i>slanted</i>    | <code>\textsc{...}</code>     | SMALL CAPS       |
| <code>\emph{...}</code>   | <i>emphasized</i> | <code>\textnormal{...}</code> | document font    |

Font Face Commands

- Put the text inside the above commands to change the font face.  
e.g. `\textbf{this text will be in bold face}`

# Font Face & Size

`\tiny`

tiny font

`\scriptsize`

very small font

`\footnotesize`

quite small font

`\small`

small font

`\normalsize`

normal font

`\large`

large font

`\Large`

large font

`\LARGE`

very large font

`\huge`

huge

`\Huge`

largest

*Font Size Commands*

- These commands will affect font size in the following text
- Use `{ ... }` to limit its effect range  
e.g. `{\small small size text}`

# Spacing

- Use package *geometry* to change the paper margin

```
\usepackage[top=3cm,bottom=3cm,left=2.5cm,right=2.5cm]{geometry}
```

- To force a new page, use:
  - `\newpage` : create a new page
  - `\clearpage` : create a new page and flush all the floats
  - `\cleardoublepage` : In addition to `\clearpage` , it makes the next page a right-hand page for two-sided printing
- Force a space using `~` (unbreakable) or `\` (breakable)
- Insert horizontal/vertical spaces with `\hspace{1em}` or `\vspace{1ex}`
- Create a line break and insert vertical spaces using `\\ [1ex]`
- Fill space using `\hfill` or `\vfill`

| unit | meaning  |
|------|--|
| pt   | a point is approximately 1/72.27 inch                        |
| mm   | a millimeter   |
| cm   | a centimeter   |
| in   | inch   |
| ex   | roughly the height of an 'x' (lowercase) in the current font |
| em   | roughly the width of an 'M' (uppercase) in the current font  |
| mu   | math unit equal to 1/18 em                                   |

*Length Unit in  $\text{\LaTeX}$*

# Alignment

```
\begin{center}  
  text to be centered  
\end{center}  
  
\begin{flushleft}  
  text to be flushed left  
\end{flushleft}  
  
\begin{flushright}  
  text to be flushed right  
\end{flushright}
```

text to be centered

text to be flushed left

text to be flushed right

# Hyphenation

- $\text{\TeX}$  hyphenates words whenever necessary
- You can custom the hyphenation using `\hyphenation{<word list>}` in the preamble
- For example, `\hyphenation{FORTRAN Hy-phen-a-tion}` instructs:
  - Prevents “FORTRAN”, “Fortran” and “fortran” from being hyphenated
  - Allow “hyphenation” to be hyphenated as well as “Hyphenation”
- Or use `\-` inserts a discretionary hyphen into a word

```
I think this is: su\~per\~cal\~%  
i\~frag\~i\~lis\~tic\~ex\~pi\~%  
al\~i\~do\~cious
```

I think this is: supercalifragilis-  
ticexpialidocious

- `\mbox{...}` causes its argument to be kept together under all circumstances

```
My phone number will change soon.  
It will be \mbox{0116 291 2319}.
```

My phone number will change  
soon. It will be 0116 291 2319.

# Document Structure

- $\text{\LaTeX}$  is built off the idea *structure* over *formatting*
- You can structure the documents using following commands

```
\part{part name} % only available in book
\chapter{chapter name} % available in book and report
\section{section name}
\subsection{subsection name}
\subsubsection{subsubsection name}
```

- The star version commands (e.g. `\section*{}`) suppress the numbering and are not added in the table of contents.
- `\tableofcontents` can be used to create table of contents.
- Use `\appendix` to put rest of content in the appendix.
- For large project, you can put each chapter/section in a separated file. Then use `\input{file_name}` to include them in the root file.

# List Structures

- There are three list structures in  $\text{\LaTeX}$

```
\begin{enumerate}  
  \item Item 1  
  \item Item 2  
\end{enumerate}  
\begin{itemize}  
  \item Item 1  
  \item Item 2  
\end{itemize}  
\begin{description}  
  \item[key1] Item 1  
  \item[key2] Item 2  
\end{description}
```

1. Item 1

2. Item 2

• Item 1

• Item 2

**key1** Item 1

**key2** Item 2



# List Structures

- You can use them in nested fashion

```
\begin{enumerate}
  \item Level 1
    \begin{enumerate}
      \item Level 2
    \end{enumerate}
  \item Level 1
    \begin{itemize}
      \item Level 2
    \end{itemize}
\end{enumerate}
```

1. Level 1
  - 1.1 Level 2
2. Level 1
  - Level 2

# List Structures

- Use package *enumitem* to custom the list format

```
\usepackage{enumitem}
\setlist{noitemsep,partopsep=0pt,topsep=.8ex}
\setlist[enumerate,1]{label=\arabic*.,ref=\arabic*}
\newlist{inlineenum}{enumerate*}{1}
\setlist[inlineenum]{label=(\roman*),ref=(\roman*)}

\begin{itemize}[label=-]
  \item Item
\end{itemize}
```

- Common mathematical packages

```
\usepackage{amsmath}  
\usepackage{amssymb}  
\usepackage{amsfonts}  
\usepackage{mathrsfs}  
\usepackage{latexsym}
```

- List of mathematical symbols  
<https://www.caam.rice.edu/~heinken/latex/symbols.pdf>
- “Short Math Guide for  $\text{\LaTeX}$ ” (access by `texdoc short-math-guide`) for comprehensive guide

# Math Mode & Environment

- There are two math mode

- Inline math mode: `\sum_k^n k` or `\(\sum_k^n k\)` to typeset  $\sum_k^n k$
- Display math mode: `$$\sum_k^n k$$` or `\[\sum_k^n k\]` to typeset

$$\sum_k^n k$$

- Use `equation` environment to number the equation in display mode

```
\begin{equation}
E = mc^2
\end{equation}
```

$$E = mc^2 \quad (1)$$

- Use `\tag` to change the equation label

```
\begin{equation}
1 + 1 = 3 \tag{dumb}
\end{equation}
```

$$1 + 1 = 3 \quad (\text{dumb})$$

# Math Mode & Environment

- Use `align` environment to align multiple equations

```
\begin{align}
B' &= -\nabla \times E, \\
E' &= \nabla \times B - 4\pi j,
\end{align}
```

$$B' = -\nabla \times E, \quad (2)$$

$$E' = \nabla \times B - 4\pi j, \quad (3)$$

- Use `\nonumber` to disable the number for some lines

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

$$\begin{aligned} a &= b + c \\ &= d + e \end{aligned} \quad (4)$$

# Math Mode & Environment

- `align*` environment disable the number entirely

```
\begin{align*}B' &= -\nabla \times E, \\ E' &= \nabla \times B - 4\pi j, \\ \end{align*}
```

$$\begin{aligned}B' &= -\nabla \times E, \\ E' &= \nabla \times B - 4\pi j,\end{aligned}$$

- `gather` / `gather*` display a set of consecutive equations, centered and with no alignment

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

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

# Math Symbols

- The following symbols that can be used directly in math environment

+ - = ! / ( ) [ ] < > | ' : \*

+ - = ! / ( ) [ ] < > | ' : \*

- Greek letters

`\alpha`, `\beta`, `\gamma`, `\pi`, `\phi`, `\varphi`

$\alpha, \beta, \gamma, \pi, \phi, \varphi$

- Operators

`\cos(2\theta) = \cos^2\theta - \sin^2\theta`  
`\lim\limits_{x \rightarrow \infty} \exp(-x) = 0`  
`a \bmod b`  
`x \equiv a \pmod{b}`  
`\log\{N\}`

$\cos(2\theta) = \cos^2 \theta - \sin^2 \theta \quad \lim_{x \rightarrow \infty} \exp(-x) = 0$

# Math — Custom Operators

- You can define your own operators

```
\operatorname{arg\,max}_a f(a) =  
\operatorname*{arg\,max}_b f(b)
```

$$\operatorname{arg\,max}_a f(a) = \operatorname{arg\,max}_b f(b)$$

- If it is frequently used,

```
% declared in preamble  
\DeclareMathOperator*{\argmax}{arg\,max} % or \DeclareMathOperator{\argmax}{arg\,max}  
  
% then used in the document  
\[ \argmax_c f(c) \]
```



# Math — Power, Indices, Fraction, Root

- Powers and indices are equivalent to superscripts and subscripts in normal text mode. The caret (^) character is used to raise something, and the underscore (\_) is for lowering. If more than one expression is raised or lowered, they should be grouped using curly braces ({ } and { } ).

```
k_{n+1} = n^2 + k_n^2 - k_{n-1}
n^{22}
f(n) = n^5 + 4n^2 + 2 |_{n=17}
\sum_{i=1}^n i
\lim_{x \to \infty} \frac{1}{x}
```

$$k_{n+1} = n^2 + k_n^2 - k_{n-1} n^{22} f(n) = n^5 + 4n^2 + 2|_{n=1}$$

- Fraction and root

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
\sqrt{2}
\sqrt[n]{1+x+x^2+x^3+\dots+x^n}
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k} \sqrt[n]{1+x+x^2+x^3+\dots+x^n}$$

# Math — Delimiters

- Brackets, braces and delimiters

```
( a ), [ b ], \{ c \}, | d |, \| e \|,  
\langle f \rangle, \lfloor g \rfloor,  
\lceil h \rceil, \ulcorner i \urcorner
```

$$(a), [b], \{c\}, |d|, \|e\|, \langle f \rangle, [g], [h], \lceil i \rceil$$

- Automatic sizing

```
\left(\frac{x^2}{y^3}\right)  
P\left(A=2\middle|\frac{A^2}{B}>4\right)  
\left\{\frac{x^2}{y^3}\right\}
```

$$\left(\frac{x^2}{y^3}\right) P\left(A=2\middle|\frac{A^2}{B}>4\right)\left\{\frac{x^2}{y^3}\right\}$$

- Manual sizing

```
( \big( \Big( \bigg( \Bigg(
```

$$(((( ($$

- Matrices

```
\begin{matrix}  
a & b & c \\  
d & e & f \\  
g & h & i  
\end{matrix}
```

```
\begin{pmatrix}  
a & b & c \\  
d & e & f \\  
g & h & i  
\end{pmatrix}
```

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$
$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

- Other matrix environment with different delimiter: `bmatrix`, `Bmatrix`, `vmatrix`, and `Vmatrix`

# Math — Array

- Array

```
\begin{array}{c|c}  
1 & 2 \\  
\hline  
3 & 4  
\end{array}
```

$$\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$

```
f(x) = \left\{ \begin{array}{ll} x & \text{if } x > 0, \\\ 0 & \text{otherwise.} \end{array} \right.
```

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

- Cases

```
f(x) = \begin{cases} x & \text{if } x > 0, \\\ 0 & \text{otherwise.} \end{cases}
```

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

|                               |                             |
|-------------------------------|-----------------------------|
| <code>\mathnormal{...}</code> | <i>ABCDEF abcdef 123456</i> |
| <code>\mathrm{...}</code>     | ABCDEF abcdef 123456        |
| <code>\mathit{...}</code>     | <i>ABCDEF abcdef 123456</i> |
| <code>\mathbf{...}</code>     | <b>ABCDEF abcdef 123456</b> |
| <code>\mathsf{...}</code>     | ABCDEF abcdef 123456        |
| <code>\mathtt{...}</code>     | ABCDEF abcdef 123456        |
| <code>\mathfrak{...}</code>   | <i>ℳℒ℔℔℔ abcdef 123456</i>  |
| <code>\mathcal{...}</code>    | <i>ABCDEF</i>               |
| <code>\mathbb{...}</code>     | ABCDEF                      |

*Math Fonts*

| $\text{\LaTeX}$ code | Description   |
|----------------------|---|
| <code>\qquad</code>  | twice of <code>\quad</code> (= 36 mu)                     |
| <code>\quad</code>   | space equal to the current font size (= 18 mu)            |
| <code>\,</code>      | 3/18 of <code>\quad</code> (= 3 mu)                       |
| <code>\:</code>      | 4/18 of <code>\quad</code> (= 4 mu)                       |
| <code>\;</code>      | 5/18 of <code>\quad</code> (= 5 mu)                       |
| <code>\!</code>      | -3/18 of <code>\quad</code> (= -3 mu)                     |
| <code>\</code>       | space after backslash, equivalent of space in normal text |

*Spacing in Math*

| $\text{\LaTeX}$ code      | Output | Description  |
|---------------------------|--------|--|
| <code>\dots</code>        | ...    | generic dots. It automatically manages whitespaces according to the context, it's a higher level command.              |
| <code>\ldots</code>       | ...    | the output is similar to the previous one, but there is no automatic whitespace management; it works at a lower level. |
| <code>\cdots</code>       | ...    | These dots are centered relative to the height of a letter.  |
| <code>\vdots</code>       | ⋮      | vertical dots  |
| <code>\ddots</code>       | ⋱      | diagonal dots  |
| <code>\hdotsfor{n}</code> |        | to be used in matrices, it creates a row of dots spanning $n$ columns.   |

*Dots in Math*

| $\text{\LaTeX}$ code         | Output              | Description                                |
|------------------------------|---------------------|--|
| <code>A_1,A_2,\dotsc,</code> | $A_1,A_2,\dots,$    | for “dots with commas”                     |
| <code>A_1+\dotsb+A_N</code>  | $A_1 + \dots + A_N$ | for “dots with binary operators/relations” |
| <code>A_1 \dotsm A_N</code>  | $A_1 \cdots A_N$    | for “multiplication dots”                  |
| <code>\int_a^b \dotsi</code> | $\int_a^b \cdots$   | for “dots with integrals”                  |
| <code>A_1\dotso A_N</code>   | $A_1 \dots A_N$     | for “other dots” (none of the above)       |

## *Semantic Dots in Math*

- It is recommended to use these semantically oriented commands.



# Figure and Table

- To create a float block to place figure or table

```
% for figure
\begin{figure} ... \end{figure}
% for table
\begin{table} ... \end{table}
% star version put it across multiple columns
\begin{figure*} ... \end{figure*}
\begin{table*} ... \end{table*}
```

- Positioning can be denoted as an optional argument

```
\begin{figure}[placement specifier] ... \end{figure}
```

# Figure and Table

| Specifier | Description   |
|-----------|---|
| 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.<br>Require <code>\usepackage{float}</code> .                  |

## *Placement Specifier for Floats*

- You can use single or multiple specifiers.  $\text{\LaTeX}$  will attempt to apply the rules in descending priority. e.g. `\begin{figure}[tbhp] ... \end{figure}` .
- Use `\FloatBarrier` to prevent floats from being moved over it. (Require `\usepackage{placeins}` )

# Figure

- `\usepackage{graphicx}` is commonly required to insert the figure.
- Use `\includegraphics[opt]{figure-path}` to add the figure  
`opt` can be `width=.5\linewidth`, `height=10cm`, or `scale=0.5`, etc.
- `\textwidth` means the width of current page, `\linewidth` means the width of the current column.
- Image can be in .png/.jpg/.pdf/.eps format. It is recommended to use .pdf or .eps vector formats.
- Common tools to draw the figure:
  - inkscape <https://inkscape.org>
  - tikz `texdoc tikz`, <https://github.com/xiaohanyu/awesome-tikz>
  - gnuplot <http://www.gnuplot.info>
  - matplotlib <https://matplotlib.org> or seaborn <https://seaborn.pydata.org>
  - <https://www.flaticon.com> offers many vector graph assets.

# Caption

- Use `\caption{}` to add the caption, `\caption*{}` to suppress the numbering.
- Package *caption* provides the command `\captionof{<type>}{<caption text>}` that lets you typeset a caption without a floating environment.
- It also allows to custom the caption format.

```
\usepackage{caption}  
\captionsetup{format=plain, font=small, labelfont=bf}
```

# Figure

- An example of full figure block

```
\begin{figure}[t] % put the figure at the top of the page
  \centering
  \includegraphics[width=.8\linewidth]{path-to-the-figure-file}
  \caption{The caption of this figure}
\end{figure}
```

# TikZ Figure

- You can create a tikz figure in a standalone file.

```
\documentclass[tikz]{standalone}
\usetikzlibrary{positioning}
\begin{document}
\begin{tikzpicture}
  \node[draw] (start) { Start };
  \node[draw, right=2cm of start] (end) { End };
  \draw[-latex] (start) -- (end);
\end{tikzpicture}
\end{document}
```

- The standalone file can be compiled directly or included in the document.

```
% need to pass additional `-shell-escape` argument to the compiler
\usepackage[mode=buildnew]{standalone}

\begin{figure}[t]
  \centering
  \includestandalone[width=0.8\linewidth]{./figure} % without the `.tex` extension
  \caption{TikZ Figure in Article}
\end{figure}
```

# Formatting Tables

- The *tabular* environment defines the table
- Use package *booktabs* to create professional table

```
\centering\small
\begin{tabular}{llr}
\toprule
\multicolumn{2}{c}{Item} & \\
\cmidrule(r){1-2}
Animal & Description & Price (\$) \\
\midrule
Gnat & per gram & 13.65 \\
      & each & 0.01 \\
Gnu & stuffed & 92.50 \\
Emu & stuffed & 33.33 \\
Armadillo & frozen & 8.99 \\
\bottomrule
\end{tabular}
```

| Item      |             |            |
|-----------|-------------|------------|
| Animal    | Description | Price (\$) |
| Gnat      | per gram    | 13.65      |
|           | each        | 0.01       |
| Gnu       | stuffed     | 92.50      |
| Emu       | stuffed     | 33.33      |
| Armadillo | frozen      | 8.99       |

- More guidance: <https://en.wikibooks.org/wiki/LaTeX/Tables>
- *excel2latex* can be used to generate  $\text{\LaTeX}$  code from excel table

# Subfloats

- Use package *subcaption* to create subfigures or subtables

```
\begin{figure}
  \centering
  \begin{subfigure}[b]{0.5\textwidth}
    \includegraphics[width=\textwidth]{gull}
    \caption{A gull}
  \end{subfigure}
  ~%add desired spacing between images, e.g. ~, \quad, \hfill, \\ etc.
  \begin{subfigure}[b]{0.5\textwidth}
    \includegraphics[width=\textwidth]{tiger}
    \caption{A tiger}
  \end{subfigure}
  \caption{Pictures of animals}
\end{figure}
```



# References

- You can use `\label{<label name>}` to make a label

```
\section{Section Title}\label{sec:label-a}  
\begin{figure}  
  ...  
  \caption{figure caption}\label{fig:label-b}  
\end{figure}  
\begin{equation}  
  E=mc^2 \label{eqn:label-c}  
\end{equation}
```

- Use `\ref{<label name>}` to reference them
- Use package *hyperref* to generate pdf hyperlink and create url  
e.g. `\url{https://google.com}`
- Use package *cleveref* for auto infer reference types  
e.g. `\cref{fig:label}` is equivalence to `Fig.~\ref{fig:label}`
- Use `\footnote{...}` to insert footnote

# Theorems

- There are many packages to offer theorem environments.
- Here, we use `\usepackage{amsthm,thmtools}`
- Declare the theorem environments (document `texdoc thmtools`)

```
\declaretheorem[style=plain]{axiom}  
\declaretheorem[style=definition]{definition}  
\declaretheorem[style=definition]{example}  
\declaretheorem[style=plain]{lemma}  
\declaretheorem[style=plain]{theorem}  
\declaretheorem[style=remark]{remark}
```

- Use it in the document

```
\begin{theorem}[Euclid]  
  For every prime  $p$ , there is a prime  $p' > p$ .  
  In particular, there are infinitely many primes.  
\end{theorem}
```

- `\usepackage{thm-restate}` to repeat the same theorem multiple times

# Algorithms

- There are two common packages to typeset algorithm:
  - *algorithm2e*
  - *algorithmicx*
- Example using algorithm2e:

```
\begin{algorithm}[H]
  \caption{How to write algorithms}
  \KwData{this text}
  \KwResult{learn to write algorithm}
  initialization\;
  \While{not at end of this document}{
    read current\;
    \eIf{understand}{
      go to next section\;
      current section becomes this one\;
    }{
      go back to the beginning\;
    }
  }
\end{algorithm}
```

---

**Algorithm 1:** How to write algorithms

---

**Data:** this text

**Result:** learn to write algorithm  
initialization;

**while** *not at end of this document* **do**

|  |   |
|--|---|
|  | read current;                           |
|  | <b>if</b> <i>understand</i> <b>then</b> |
|  | go to next section;                     |
|  | current section becomes this one;       |
|  | <b>else</b>                             |
|  | go back to the beginning;               |

---

# Source Code Highlight

- Using package *listings* to highlight the source code.

```
\begin{lstlisting}[language=Python]
def fib():
    a, b = 0, 1
    while 1:
        yield a
        a, b = b, a + b
\end{lstlisting}
```

```
def fib():
    a, b = 0, 1
    while 1:
        yield a
        a, b = b, a + b
```

- Alternatively, use `\lstinputlisting[opt]{file path}` to read code from another file.
- Package *minted* offers more features and better highlights. But it requires:
  - Install Pygments <http://pygments.org>
  - Pass additional argument `-shell-escape` to the compiler

# Bibliography

- `.bib` file acts as a database of references, and only includes in the bibliography those references you cite in your paper

```
@article{nameofentry,  
  author = {John Doe and Jane Doe},  
  title  = {The title of the work},  
  journal = {The name of the journal},  
  year   = 1993,  
  pages  = {201--213},  
  month  = 7,  
  volume = 4  
}
```

```
@inproceedings{nameofentry,  
  author = {Doe, John and Doe, Jane},  
  title  = {The title of the work},  
  booktitle = {The name of the proceedings},  
  year   = 2019,  
  pages  = {100--110},  
  month  = 1,  
  address = {Conference location},  
}
```

- More examples can be found in
  - <http://web.mit.edu/rsi/www/pdfs/bibtex-format.pdf>
  - <https://www.verbosus.com/bibtex-style-examples.html>

# Bibliography

- Use `\cite{nameofentry}` to cite the referenced paper in the main text
- There are two solutions to typeset bibliography
  - BibTeX: old and widely support

```
cite some paper~\cite{paperentry}.  
\bibliographystyle{IEEEtrans}  
\bibliography{path to bib file}
```

- BibLaTeX: new and have more features, document: `texdoc biblatex`

```
\usepackage[style=ieee,giveninits=true,doi=false]{biblatex}  
\addbibresource{path to bib file}  
\begin{document}  
cite some paper~\cite{paperentry}.  
\printbibliography  
\end{document}
```

## Advanced Usages

---

# More Packages

- Color: *color*, *xcolor*

```
\usepackage{color}  
\usepackage[table,dvipsnames]{xcolor}
```

- Draw Boxes: *tcolorbox*
- Draw Graphics: *tikz*, *overpic*
- Slides: *beamer*
- Poster: *tikzposter*
- Miscellaneous: *microtype*, *footmisc*, *balance*



# Define Commands and Environment

- Define command using: `\newcommand{\name}[num]{definition}`

```
\newcommand{\highlight}[1]{%  
  {\color{red} #1}%  
}  
\highlight{Text in red}
```

Text in red

- Define environment using: `\newenvironment{name}[num]{before}{after}`

```
\newenvironment{response}{%  
  \begingroup  
  \textbf{Response}: \itshape  
}{%  
  \endgroup  
}  
\begin{response}  
  Some response.  
\end{response}
```

Response: *Some response.*

- More information: <https://en.wikibooks.org/wiki/LaTeX/Macros>

# LaTeX Engines

- There are several LaTeX engines
  - **pdf<sub>l</sub>atex**: most commonly used
  - **xelatex** and **lua<sub>l</sub>atex**: new, offer more features
    - better font support, typeset other language than English, etc
- To compile LaTeX manually, you usually need run multiple commands

```
pdflatex root_file  
bibttex root_file # or `biber root_file` if using biblatex  
pdflatex root_file  
pdflatex root_file
```

- Or use **latexmk** to automatically run commands for you

```
latexmk -pdf root_file # use pdflatex  
latexmk -xelatex root_file # use xelatex  
latexmk -lualatex root_file # use lualatex
```

- Some LaTeX editors (such as VSCode with LaTeX Workshop, Vim with vimtex) use **latexmk** under the hook

# Other Command Line Tools

- **latexmk**

- In addition to build project, it can also be used to clean up auxiliary files

```
latexmk -c
```

- It is highly customizable. You can create `.latexmkrc` file to configure `latexmk` . document: `texdoc latexmk`

```
# use pdflatex by default, so you don't need to pass `~pdf` argument  
$pdf_mode = 1;  
# -synctex=1 allow easy jumps between latex source and pdf file  
# -file-line-error make latex report file and line number when there is an error  
$pdflatex = 'pdflatex -synctex=1 -file-line-error %O %S';
```

- **chktex**: Lint the  $\text{\LaTeX}$  source code for common problem. document: `texdoc chktex`
- **latexindent**: Format the  $\text{\LaTeX}$  source code. document: `texdoc latexindent`
- **latexdiff**: Marking up difference between  $\text{\LaTeX}$  files. document: `texdoc latexdiff`

## Further Readings

- $\text{\LaTeX}$  Wikibooks: <https://en.wikibooks.org/wiki/LaTeX>
- The Not So Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$ : `texdoc lshort`
- Short Math Guide for  $\text{\LaTeX}$ : `texdoc short-math-guide`
- The TeX FAQ List: <https://texfaq.org>
- LaTeX Stack Exchange: <https://tex.stackexchange.com>
- Always remember to use Google when you encounter problems

Thanks  
Questions?