LaTeX Intro Manual

jan_chen

September 2023

Contents

1 Introduction to LaTeX

LaTeX, which is pronounced *Lah-tech* or *Lay-tech*, is a **document preparation system for high-quality typesetting**. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. Using LaTeX can make your work look more professional and organized. Besides, it is also an useful tool to take notes that intensively involve formulae and symbols.

2 How to LaTeX

2.1 Begin with documentclass

The first command `\documentclass[option]{class}` tells the compiler which kind of document class you'll choose. Specify the document class in `{}`, such as **article**, **book**, **slides** or **ctexart** if you want to write in Chinese. However, **article** is the most commonly used class with some structural command:

```
\part{}, \chapter{}, \section{}, \subsection{}, \subsection{} ...
```

Usually you don't need to care about the **option** parameter, which specifies the default font and aligning patterns. If you need to modify it, go Google yourself :P.

2.2 Include some useful syntax or tools in usepackage

The second command `\usepackage{}` tells the compiler which **macro package** you want to include in your article. There are many useful functions defined in those packages, if you want use them, you need to include the package. Packages you almost always have to include:

- The **asmmath**: defines many mathematical functions and environments.
  Official documentation here

- The **graphicx**: defines the methods to insert images into your article.
  Official documentation: here

- The **geometry**: defines the methods to modify the paper layout of your article.
  Official documentation: here

- The **array** or **booktabs**: defines the methods to make tables.
  I think you have known where to find the official documentation :D.

- The **listings**: defines the methods to insert codes into your article.
- The **xcolor**: if you want to add some colors in your article.
- The **ctex**: if you want type Chinese.

Some packages require you to specify the overall settings before beginning document, don't forget that.

2.3 One last thing before starting to write

You should specify the **title**, **author** and **date** before starting your document. You can simply use `\title{}`, `\author{}` and `\date{}` or start a title page by calling `\begin{titlepage} & \end{titlepage}`, but we'll not discuss about it today.

2.4 Now we get start, formally

Call `\begin{document}` to begin with your article (and remember to call `\end{document}` when you finish it).
Then make your title by calling `\maketitle`. After that, we truly start.

2.4.1 Structural command

To make your article organized, you should make your article a clear structure. Make good use of `\section{}`, `\subsection{}` and `\subsubsection{}`, where you can name the section title in the `{}`.

As default, each section will have its own number. To cancel the section number, use `\section*{}` instead (this is a common way shared by many other environments).

Besides, there are no indent before the first paragraph, if you want indent the first paragraph, use the **indentfirst** package.

To set the paragraph patterns, use the `\setlength{}{}` command (this command should be used before starting the document).

- `\setlength{\parindent}{1em}` will set the indent size to be 1em
- `\setlength{\parskip}{1em}` will set the paragraph skip to be 1em
- `\setlength{\baselineskip}{1em}` will set the line skip to be 1em (if this doesn't work, try `\linespread{2.0}`, which will set the line skip two times the normal size)

(you may want to find out the exact size of 1em by yourself and other scales)

There are several ways to start a new paragraph. You can use `\par` to do it or leave a blank line in your code to separate each paragraph. To simply start a new line, add `\\` at the end of the last line.

If you want to insert a catalogue into your work, you can use `\tableofcontents` to automatically generate one. The effect is like the *Contents* part of the current document. The catalogue will include all sections as default.

2.4.2 Add math to your work

All math formulae and symbols can only be displayed in **math environment**. The simplest way to create math environment is using `$`. `$equation$` will insert inline math equation and `$$equations$$` will insert equations between lines. If you want to generate equations with equation numbers, you can use `\begin{equation}` (to cancel it, add *).

However, in the **equation** environment, you can not start a new line. If you want to insert multiple lines of equations, try `\begin{align}`, where you are allowed to use `\\` to start new lines. Besides, **align** environment also enables you to literally 'align' your equations. To achieve this, add `&` in your equations to specify to which symbol you want latex to align them.

Most of the math functions are in the form of `\func{}{}`, where you can put numbers or symbols in the `{}`. E.g. `\frac{a}{b}` will generate $\frac{a}{b}$. LaTeX has embedded plenty of math functions and symbols, we'll not name them in this manual but you can still Google them yourself.

Other math environments:

- `\begin{gather}`: the equations will be center aligned

- `\begin{split}`: give one number to several lines of equations (need to be embedded within equation environment)

- `\begin{cases}`: use a `{` to list all the cases

- `\begin{math}`: more general, like `\begin{align}` but no numbers

The textual rule in math environments is quite different from others. If you want to insert some textual explanation into your equations, please use `\text{}` (this command is included in **amsmath** package).

Math environments don't support blank space either. If you want to insert spaces, use `\quad` or `\qquad` (there are other types of blank which we don't mention here).

2.4.3 Insert pictures

Before inserting pictures, make sure you have included the **graphicx** package.

Here is an example:

```
\begin{ figure }[htbp]
    \centering
    \ includegraphics [scale = 0.5]{image.png}
    \caption{Image}
    \ label {1}
\end{ figure }[htbp]
```

Firstly, use `\begin{figure}[]` to create a floating figure environment. You can specify the position of the figure in the `[]`. The common input is `[htbp]` where

- **h** means placing the figure at the current position, if the current page doesn't have enough space, this option won't work

- **t** means placing the figure at the top of the page

- **b** means placing the figure at the bottom of the page

- **p** means placing the figure at any page that allows

if **h** is not satisfied, then **t**, then **b** and at last **p**. (Actually, latex will always follow this order)

However, if you definitely want your figure to be inserted at the current place, include **float** package and use `[H]` instead of `[htbp]`

The `\centering` command will center align the image.

The `\includegraphics[]{}` is where you tell the compiler which image to choose. Remember, you have to **save the image file under the folder of the latex project**. Input the **relative path** of the file in `{}` and other options in `[]` such as the size or the scale of the image.

The `\caption{}` command will give your image a title and `\label{}` will label your image so that you can cite the image later using `\ref{}`

2.4.4 Create tables

First you should know that there are many latex table generators in the Internet, try to use them will save you a lot of time. The one I recommend is the *Tables Generator*, click here to know more. It is a powerful table generator, which can also generate **MarkDown** tables :D, it is free, of course.

Here is the code of a table generate by TG, you can copy and try to generate a table by yourself:

```
\begin{table}[]
\centering
\begin{tabular}{| lll |}
\hline
     & column1 & column2 \\ \hline
row1 & a       & b       \\ \hline
row2 & c       & d       \\ \hline
\end{tabular}
\caption{MyTable}
\label{tab:my—table}
\end{table}
```

As you can see, some options are similar to inserting pictures. The

```
\begin{tabular}{|l l l|}
```

command tells the compiler to generate a table with 3 columns, each of which is aligned left. If you want them to be center aligned, use

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

The `\hline` command insert a horizontal line separate each row (you can call this command multiple times). The `&` tells the compiler which element belongs to which column.

Above is the most common way to create tables (and powerful enough), still, there are more options where you can find on TG website or in the internet, we won't elaborate them here.

2.4.5 Notations and citations

Remember you have label the pictures and tables before? Now it's time to cite them.

Use `\ref{}` to cite the pictures and tables wherever you want. The label name should be consistent with the ref name. If you write a huge article and it's hard to find where the source is, use `\pageref{}` to cite the page number of the source. Note that `\label{}` can not only be applied to figures and tables, it can also label equations.

If you want to cite hyperlink in your article, use **hyperref** package, which contains many options for citing links (for details, refer to official documents or search the internet). The commonly used one is `\href{URL}{text}` where you can create a hyperlink hidden behind the text. Or you can use `\url{URL}` for a single link.

To insert footnote, simply use `\footnote{}`. To set margin note, use `\marginpar{}`.

If you want to cite other's work, you need to create a reference list. There are two ways for you to do this.

- Not use BibTeX: use `\begin{\thebibliography{}}` to create a bibliography environment. First, specify the maximum number of reference in the `{}`. Then, use `\bibitem[]{}`to add a reference. Specify the ref name in the {} and the reference information after that command (Note that this method doesn't provide a standard reference pattern, you should take care of it yourself).

- Use BibTeX[1]: Firstly, you should create a .bib file, say *ref.bib*, which will contain the information of the bibs. In the .bib file record the bib like this:

```
@article {name1, %the name of this ref
author = {Authors, connected with 'and'
if there are many},
title = {Title},
journal = {Journal name},
volume = {volume n},
number = {page number},
year = {published year},
abstract = {not necessary}
}
```

```
@book{name2,
author = "",
year = "",
title = "",
publisher = ""
}
```

Then save the .bib file under the same folder of the latex project.

At last, insert two commands before ending the document:

  - `\bibliographystyle{plain}`: specify the reference format, multiple choices, search by yourself.
  - `\bibliography{ref}`: insert the bib, here the **ref** corresponds to the name of the .bib file.

Now you can use `\cite{}` in your article to cite other's works.

3 Before The End

LaTeX is really a powerful document preparation system. There is a great deal of other beautiful functions waiting for you to explore. This manual only contains the most basic syntax of LaTeX but I hope it could help you to introduce to it and arouse your interest in high-quality digital writing, though there are still many flaws in this document since I'm neither a professional LaTeX user `XD. Thanks!`

References

[1] https://www.ctan.org/pkg/amsmath

[2] https://www.ctan.org/pkg/graphicx

[3] https://www.ctan.org/pkg/geometry

[4] https://www.tablesgenerator.com/

[1] BibTeX is a bibliography manage tool, a BibTeX-style file ends with .bib which can be seen as a bibliography library