

Introduction to \LaTeX

Liu Yihao

March 16, 2017

- 1 Getting Started
 - What is \LaTeX
 - Installation of \LaTeX
 - Selection of IDEs
 - Documentation
- 2 The Basic Usages
- 3 Use Text in \LaTeX
- 4 Use Maths in \LaTeX
- 5 Use Graphics and Tables in \LaTeX
- 6 Some Advantage Usages

What is L^AT_EX

From Wikipedia, the free encyclopedia

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 stylise text throughout a document (such as bold and italic), and to add citations and cross-references. A TeX distribution such as TeX Live 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 stylised as L^AT_EX.

Installation of L^AT_EX

Windows

Download TeXLive on the following website

<http://mirror.hust.edu.cn/CTAN/systems/texlive/Images/>

Linux

For example, on Ubuntu (or Debian), Enter the command

`sudo apt-get install texlive-full`

MacOS

Download MacTeX on the following website

<http://tug.org/mactex/mactex-download.html>

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/>

Tex Studio

<http://www.texstudio.org/>

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

Documentation on your computer

If you've installed a full version of TeXLive (as strongly recommended), the \LaTeX documentation about all you want to is in front of you.

Open the command line and input the command

`texdoc docname`

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

`tex` A documentation about `TeX`

`article` A documentation about documentclass `article`

`beamer` A documentation about documentclass `beamer`

`pgf` A documentation about `TikZ` and `PGF` (used to draw graphs)

Just try to `texdoc` about all new things then you will be an expert in \LaTeX .

1 Getting Started

2 The Basic Usages

- Common syntax
- Documentclass
- Document environment
- Packages
- Title, Author and Date
- Sections

3 Use Text in \LaTeX

4 Use Maths in \LaTeX

5 Use Graphics and Tables in \LaTeX

6 Some Advanced Usages

The common syntax of \LaTeX commands

Definition

Command is a word which can be identified by Latex and represents a certain function in output file, or in relation with some specific character or format

All \LaTeX commands have the following syntax

\backslash **command_name** \langle **special_args** \rangle $[$ **optional_args** $]$ $\{$ **required_args** $\}$

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

optional_args Used to define mode of the command, if not specified, \LaTeX will use the default mode

required_args 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 \backslash before each command.

The common syntax of \LaTeX environments

Definition

Environment is an encapsulated part which has a certain format so that it will not be influenced by outer context

All \LaTeX environments have the following syntax

```
\begin{environment_name}<special_args>[optional_args]
```

...

```
\end{environment_name}
```

special_args Similar to commands

optional_args Similar to commands

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

Environment in environment

Of course, the environments can be nested.

Example

```
\begin{environment_name}  
...  
  \begin{environment_name_2}  
    ...  
    \end{environment_name_2}  
  ...  
\end{environment_name}
```

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

book Write a book

beamer Produce a lecture silde like this!

Actually some options can be added, such as

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

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

The article class

The `article` class 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` - 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

The document environment

Definition

An document starts with the `document` environment. A typical (simplest) example is presented below.

Example

```
\documentclass[a4paper]{article}  
\begin{document}  
  ...  
  Hello World!  
  ...  
\end{document}
```

All of your contents should be in the document environment. The document environment **MUST** be **unique** in the whole file.

Magic of packages

Some environments or commands cannot be used directly. In this case, **packages** should be included between **documentclass** and **document** environment.

Command

```
\usepackage[optional_args]{name}
```

There are some very useful packages that you can **ALWAYS** include:

amsmath Define various maths environments

amssymb Define various maths symbols

geometry Adjust the margin, paper size, and etc.

enumerate Generate a list like this!

graphicx Insert image of all types

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

Title, Author and Date

It's very useful to generate a title on the first page of a document, then these commands can be added between `documentclass` and `document` environment.

Command

```
\title{the title}  
\author{the author}  
\date{the date}
```

You can simply use `\date{\today}` to display today's date.

Then in the `document` environment, use the command `\maketitle` to generate a title.

Dividing into sections

Command

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

The default style of sections is like

1 Example Section Name

1.2 Example Subsection Name

1.2.3 Example Subsubsection Name

If a star(*) is added, the sequence number of the section, subsection or subsubsection won't be displayed.

Notice: Sections can be sorted into commands, not environments, so it doesn't have `begin` and `end` clauses. However, the whole contents between two sections is belonged to one section

- 1 Getting Started
- 2 The Basic Usages
- 3 Use Text in \LaTeX**
 - UTF-8 encoding
 - Special symbols and accents
 - Spaces, lines and pages
 - Fonts
 - Enumerate
 - Lines on text
- 4 Use Maths in \LaTeX
- 5 Use Graphics and Tables in \LaTeX
- 6 Some Advantage Usages

Use UTF-8 encoding in L^AT_EX

UTF-8 encoding is widely used in modern computer applications, so it's useful to include the `inputenc` package and use UTF-8 encoding.

Command

```
\usepackage[utf-8]{inputenc}
```

Example

café

However, different operating systems and compiling engines have different support on UTF-8 encoding, some UTF-8 codes that work on your computer may not work on others, so it is recommended to use commands (will be introduced later) instead of directly copy and paste the UTF-8 codes from the Internet.

(This part is not important)

If you want to use a language other than English, another package `babel` is needed.

Command

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

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

Example

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

Then L^AT_EX will print $\alpha\beta\zeta\delta\epsilon\varphi\gamma AB^{\Delta}E\Phi\Gamma$

Of course, you can use some commands these greek letters, such as

`\alpha`, `\beta` and etc, which is more convenient when you only need to print few of them, and it doesn't need any package listed above.

Special symbols

Some special symbols can't be directly used since they are reserved by L^AT_EX

| | | | | | | | | | | | |
|-----------------|---|------------------|----|-----------------|---|-----------------------------|---|-----------------|---|-----------------|---|
| <code>\#</code> | # | <code>\\$</code> | \$ | <code>\%</code> | % | <code>\&</code> | & | <code>\~</code> | ~ | <code>\`</code> | ` |
| <code>\{</code> | { | <code>\}</code> | } | <code>\-</code> | - | <code>\textbackslash</code> | \ | | | | |

Many L^AT_EX 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

| | | | | | | | | | |
|---------------------|----|--------------------|---|--------------------|---|---------------------|---|--------------------|---|
| <code>\`{o}</code> | ò | <code>\'{o}</code> | ó | <code>\^{o}</code> | ô | <code>\\"{o}</code> | ö | <code>\~{o}</code> | õ |
| <code>\={o}</code> | ō | <code>\.{o}</code> | ô | <code>\u{o}</code> | ǔ | <code>\v{o}</code> | ǖ | <code>\H{o}</code> | ő |
| <code>\t{oo}</code> | ôo | <code>\r{o}</code> | õ | <code>\c{o}</code> | ç | <code>\d{o}</code> | ð | <code>\b{o}</code> | ɒ |



Something interesting

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

- `\TeX` - T_EX
- `\LaTeX` - L^AT_EX
- `\LaTeXe` - L^AT_EX 2_ε

Spaces may be confusing

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

-  - the basic space in L^AT_EX (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 L^AT_EX about it using a tie (`~`), such as Prof.`~`Hamade (Prof. Hamade).
- `\,` - makes a 1/6 em space, commonly used before units (notice the space before em on this page)
- `\;` - makes a 2/7 em space
- `\quad` - makes a 1 em space
- `\qquad` - makes a 2 em space
- `` - makes actually the space of `text`, but `text` will be invisible.

Separate contents into lines and pages

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

- `\newline` - begin a new line
- `\\` - begin a new line
- `\\[offset]` - begin a new line with an offset
- `\linebreak` - begin a new line with the words discrete
- `\newpage` - begin a new page
- `%` - begin a line comment

Basic commands about fonts

First, let's start with some commands that transform font types

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

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

```
{\bf Sample Text}
```


Sometimes we don't want to transform the font types, instead, we can only change the font type of some specified text, then the following commands are used (you can similarly use all font types on the previous page)

- `\textbf` - **Sample Text**
- `\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 `\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}`)

Enumerate and Item

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.], ...

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, the position of `style` 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.

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

- 1 Getting Started
- 2 The Basic Usages
- 3 Use Text in \LaTeX
- 4 Use Maths in \LaTeX**
 - Basic equations
 - Aligns
 - Inline
- 5 Use Graphics and Tables in \LaTeX
- 6 Some Advantage Usages
- 7 References

The equation environment

An `equation` environment contains a set of maths equations

Command

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

Example

$$\operatorname{curl} F = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) \hat{n}_x + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) \hat{n}_y + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \hat{n}_z \quad (1)$$

If a star(*) is added, the sequence number of the equation won't be displayed. Note that the environment name in the `\begin` and `\end` statements must be the same(both or neither have a * here).

The L^AT_EX script of the equation above is quite long, but not so difficult as you think so, while how I display the script to you is far more confusing, and you may check it in the tex file of the lecture slides

```
curl\ 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
```

In the script, only a space after `\` will be printed as a space, `\partial` prints the symbol ∂ , `\frac{...}{...}` makes a fraction, `\left(` and `\right)` makes brackets (of course they can be nested and must be in couple, but you can use two kinds of brackets on the both side, i.e., `\left[` and `\right\rbrace`, in which you must use `\rbrace` or `\}` to print a right brace `}`

How about equations with multiple lines?

The `aligned` environment can be used.

Example

$$\begin{cases} x + y = 1 \\ x - y = 1 \end{cases} \implies \begin{cases} x = 1 \\ y = 0 \end{cases} \quad (2)$$

```
\left\lbrack\begin{aligned}
  x+y&=1\\x-y&=1
\end{aligned}\right.\Longrightarrow
\left\lbrack\begin{aligned}
  x&=1\\y&=0
\end{aligned}\right.
```

We can use a dot(.) when we want to insert nothing in one of the brackets.

Something more about equation environment

What if the space between equation and the main body paragraph is considered larger than expectation? Is there any way to modify the line spacing?

In default style of equation is like

Example

your body paragraph is supposed to be typed here

$$a \times b = c \tag{3}$$

your body paragraph is supposed to be typed here

But if we add

`\setlength\abovedisplayskip{pt}` or
`\setlength\belowdisplayskip{pt}`, we have

Example

your body paragraph is supposed to be typed here

$$a \times b = c \tag{4}$$

your body paragraph is supposed to be typed here

```
{\setlength\abovedisplayskip{0pt}
\setlength\belowdisplayskip{0pt}
\begin{equation}
a \times b = c
\end{equation}}
```

The margin between the body paragraphs and the equation will be lessened as is in the example.

The align/aligned environment

An **align** environment is used outside a maths environment like **equation**

Command

```
\begin{align(*)}
...
\end{align(*)}
```

An **aligned** environment is used inside a maths environment like **equation**, it is known as an **inline** environment.

Command

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

Other properties of them are very similar.

The `align`/`aligned` environment is a basic align and multiline environment.

Example

$$a + b \Leftrightarrow b + a \tag{5}$$

$$(a + b) + c \Leftrightarrow a + (b + c) \tag{6}$$

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

In order to make a new line, you can easily use `\\` where you'd like (but not in certain maths environments such as `equation`). `&` is used to align the equations, you can use multiple `&`s and the `&`s on every line will be aligned respectively.

A simple method of entering math environment

Usually, we can use `$$...$$` to display a maths equation instead of `\begin{equation*}...\end{equation*}`, which almost have same effect.

However, there is another style of math environment, inline style, which will display the maths equation on the same line of the text before it. It is used like `$...$`

Example

This is a simple equation

$$x^2 + y^2 = 1$$

This is a simple inline equation $x^2 + y^2 = 1$

The concentration of $[\text{H}_3\text{O}^+]$

This is a simple equation `$$x^2+y^2=1$$`

This is a simple inline equation `$x^2+y^2=1$ \\`

The concentration of `[H$_3$O$^+$]`

The difference between inline and normal

Actually, the display style of inline and normal equations have some differences.

Example

Expression

`\left(\frac{1}{\frac{1}{2}}\right)`
`^{\frac{1}{2}}`

inline

$$\left(\frac{1}{\frac{1}{2}}\right)^{\frac{1}{2}}$$

normal

$$\left(\frac{1}{\frac{1}{2}}\right)^{\frac{1}{2}}$$

`\lim_{n\rightarrow\infty} a_n = +\infty`

$$\lim_{n\rightarrow\infty} a_n = +\infty$$

$$\lim_{n\rightarrow\infty} a_n = +\infty$$

`\sum_{k=1}^{10} k = 55`

$$\sum_{k=1}^{10} k = 55$$

$$\sum_{k=1}^{10} k = 55$$

However, most of the differences can be fixed by some other commands

Example

Expression

`\left(\dfrac{1}{\frac{1}{2}}\right)`
`^{\frac{1}{2}}`

inline

$$\left(\frac{1}{\frac{1}{2}}\right)^{\frac{1}{2}}$$

normal

$$\left(\frac{1}{\frac{1}{2}}\right)^{\frac{1}{2}}$$

`\lim\limits_{n\rightarrow\infty} a_n = +\infty`

$$\lim_{n \rightarrow \infty} a_n = +\infty$$

$$\lim_{n \rightarrow \infty} a_n = +\infty$$

`\sum\limits_{k=1}^{10} k = 55`

$$\sum_{k=1}^{10} k = 55$$

$$\sum_{k=1}^{10} k = 55$$

Here the command `\limits` can be used in much more situations to fix the position of the bounds. The command `\dfrac` is used to print a fraction in normal size.

- 1 Getting Started
- 2 The Basic Usages
- 3 Use Text in \LaTeX
- 4 Use Maths in \LaTeX
- 5 Use Graphics and Tables in \LaTeX**
 - Include graphs
 - Draw tables
 - Draw graphs
- 6 Some Advantage Usages
- 7 References

Include graphs

It's very useful to include graphs in L^AT_EX, especially in report and paper writing. Here is a common template of including a single floating graph.

Command

```
\usepackage{graphicx}  
\begin{figure}[position]  
  \centering  
  \includegraphics[options]{file}  
  \caption{caption}  
  \label{label}  
\end{figure}
```

- **position** - we usually use **htbp** here
- **options** - the width, height and other options about the graph
- **caption** - the caption displayed above/under the graph
- **label** - used for references in a document (will be introduced later)

Usually you need to optimize the size and some other properties of the graph, most of them can be set in `options`. (Only some useful options are listed here)

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

Example

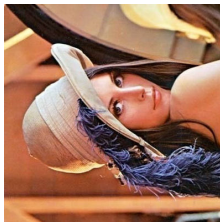


Figure: Lena

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.8\linewidth,
angle=90]{lena.jpg}
  \caption{Lena}
  \label{fig-lena}
\end{figure}
```

Include multiple graphs

Sometimes you need to include a series of graphs, then the [subfigure](#) package

Draw tables

Example

| Title 1 | Title 2 | Title 3 |
|---------|---------|---------|
| 1 | 2 | 3 |

The example above goes like this:

```
\begin{tabular}{|l|c|r|} %l represents aligning left;
                    c represents centering;
                    r represents aligning right
                    | means the vertical frame of a column
\hline % hline means to draw a horizontal line for all columns
Title 1&Title 2&Title 3 %& is used to divide contents of different columns
\hline
1 & 2 & 3
\hline
\end{tabular}
```

Something more about tabular

`\multirow`
`\multicolumn`
`\cline`

Table environment

Definition

A `table` environment is used to arrange the place of a tabular

```
\begin{table(*)}[htbp]
```

...

```
\end{table(*)}
```

[h] means inserting the tabular to the current place.

[t] means inserting the tabular to the top of the page.

[b] means inserting the tabular to the bottom of the page.

[p] means inserting the tabular to another new page, which is common in dealing with big table.

Insert a graph

- 1 Getting Started
- 2 The Basic Usages
- 3 Use Text in \LaTeX
- 4 Use Maths in \LaTeX
- 5 Use Graphics and Tables in \LaTeX
- 6 Some Advantage Usages**
 - New and renew
- 7 References

- 1 Getting Started
- 2 The Basic Usages
- 3 Use Text in \LaTeX
- 4 Use Maths in \LaTeX
- 5 Use Graphics and Tables in \LaTeX
- 6 Some Advantage Usages
- 7 References**
 - Symbol table
 - Package List
 - Contributors

Reference resources

- \LaTeX 入门, Haiyang Liu, Publishing House of Electronics Industry, 2013.6, ISBN 978-7-121-20208-7
- 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.