

Introduction to \LaTeX

Liu Yihao

December 31, 2016

What is \LaTeX

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 \LaTeX .

Installation of \LaTeX

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.

All begins with documentclass

Definition

In a \LaTeX file, the **first** line must be

```
\documentclass[...]{...}
```

For example, you can use the following:

- ❶ **\documentclass**{article} - Write a report or an science article
- ❷ **\documentclass**{book} - Write a book
- ❸ **\documentclass**{beamer} - Produce a lecture silde like this!

Actually some parameters can be added, such as

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

It means that the font-size is 11pt and the document is two-sided on an A4 paper. More features can be found in the \LaTeX Document on your own.

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]`

Example

`texdoc tex` - A documentation about [TeX](#)

`texdoc article` - A documentation about documentclass [article](#)

`texdoc beamer` - A documentation about documentclass [beamer](#)

`texdoc 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 .

Magic of packages

Definition

After the `documentclass`, packages can be included

```
\usepackage[...]{...}
```

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

- 1 `\usepackage{amsmath}` - Define various maths environments
- 2 `\usepackage{amssymb}` - Define various maths symbols
- 3 `\usepackage{geometry}` - Adjust the margin, paper size, and etc.
- 4 `\usepackage{enumerate}` - Generate a list like this!
- 5 `\usepackage{graphicx}` - Insert image of all types

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

Environments in environments

Definition

In \LaTeX , most of the environments are in the following format

```
\begin{environmentName}  
...  
\end{environmentName}
```

Of course, the environments can be nested.

Example

```
\begin{environmentName}  
...  
  \begin{anotherEnvironmentName}  
    ...  
    \end{anotherEnvironmentName}  
  ...  
\end{environmentName}
```


The document environment

Definition

An document starts with the `document` environment, after including the packages and making some settings with commands

```
\documentclass[...]{...}
```

```
\usepackage[...]{...}
```

```
\usepackage[...]{...}
```

```
\somecommand{...}
```

```
\begin{document}
```

...

```
\end{document}
```

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

Dividing into sections

Definition

A \LaTeX file can be divided into sections

$\backslash\text{section}(*)\{\dots\}$

$\backslash\text{subsection}(*)\{\dots\}$

$\backslash\text{subsubsection}(*)\{\dots\}$

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.

The equation environment

Definition

An **equation** environment contains a set of maths equations

```
\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.

The \LaTeX 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\lbrace\begin{aligned}
  x+y&=1\\x-y&=1
\end{aligned}\right.\Longrightarrow
\left\lbrace\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.

The align/aligned environment

Definition

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

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

...

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

Definition

An **aligned** environment is used inside a maths environment like **equation**

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

```
\begin{aligned}
```

...

```
\end{aligned}
```

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

Other properties of them are very similar.

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

Example

$$a + b \Leftrightarrow b + a \tag{3}$$

$$(a + b) + c \Leftrightarrow a + (b + c) \tag{4}$$

```
\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.

