

LaTeX/初学者必读

维基教科书，自由的教学读本

< [LaTeX](#)

這份文件的用意是讓你能更快速的了解[LaTeX \(https://zh.wikibooks.org/wiki/LaTeX\)](https://zh.wikibooks.org/wiki/LaTeX)。

在開始之前，請確認你的電腦已經安裝了LaTeX (安裝LaTeX必要的步驟請參照[Installation](#))。我們會從建立一個全新的LaTeX文件開始，接著會帶著你利用這份文件來輸出一份高品質的文稿，像是postscript格式或是PDF格式。

目录

LaTeX 原始碼

空格或空行

特殊字元

LaTeX 指令

LaTeX 环境

注释

輸入檔案架構

典型的命令行会话

我们的第一份文件

它代表什麼意思？

生成文件

LaTeX 原始碼

LaTeX 是所謂的標記式語言 (markup language), 使用一些標記去做文章的編排。LaTeX 系統把 LaTeX 原始碼, 依其中的標記, 轉成高品質的文件。打個比方說, LaTeX 很像網頁用的 HTML, HTML 用標記來指定網頁的排版, 文字的大小、顏色等等。

LaTeX 原始碼只是一個純文字檔案, 最早使用英文就是用 ASCII 純文字檔, 近代為多國語言的使用 (包括中文), 大多使用 UTF-8 編碼。只使用純英文的檔案, 用 UTF-8 編碼也是完全和以前 ASCII 檔案相容的, 因此建議有可能有中文或其他語言編寫文件的需求, 就使用 UTF-8 編碼。

以下是最简单的LaTeX源代码文件:

```
1 \documentclass{article}
2 \begin{document}
3 Hello World!
4 \end{document}
```

以上源代码经过LaTeX的编译,会输出dvi,pdf等格式的文件.产生如下输出:

Hello World!

空格或空行

空白字元,例如空格或是Tab在Latex中通常都被視為空格，多個空格在latex中也被視為單一空格。若空白在一行的行首時通常不會顯示出來，每一行的結束通常在Latex中都會被視為一個空格。若要表示一個段落，則在上段最後一句和下段第一句間留下空行即可，然而空下多行在latex中會被視為只有空下一行。

下面是例子。

```
1 \documentclass{article}
2 \begin{document}
3   no   matter how much
4   tab or space you input
5   or   even you start a new line
6
7       there will be only one space
8 \end{document}
```

输出的效果如下：

no matter how much tab or space you input or even you start a new line
there will be only one space

特殊字元

以下列舉字元含特殊意義，若你直接輸入下列字元，則下列字元將無法輸出。

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

若要讓這些文字得以輸出，請在這些字元前加反斜線

其他更多的字元可以使用數學公式中的特殊命令或者作為特殊標記來輸出。

- **反斜线** \ 不能通过上面所说的在前面加上另一个反斜线的方式输出, \\ 代表换行, 如果需要输出一个反斜线, 你可以采用\backslash
- \~ 的效果是在下一个字符的上方产生一条**波浪线**, 例如\~n会输出ñ, 如果要输出单个的波浪线, 你可以采用\~{ }.
- 类似地, 你也可以使用\^在之后的一个字母上方加上'^', \^{ }会产生一个在空格上的'^'.
- 如果你想要在文章中加入**链接**, 你可以考虑使用\verb, 关于这个命令, 我们将会[在formatting](#)中进行更详细的讨论.

LaTeX 指令

LaTeX的命令是大小写敏感的, 一般取下列两种格式之一:

- 命令以\开头,命令的主体部分由多个字母组成, 以换行符,空格,tab,数字等非字母字符结尾.

- 命令以\开头,命令的主体部分由一个非字母的字符组成.

对于一些需要参数的命令, 参数一般放在命令主体部分后的大括号{}中. 例如:

```
1 \^{}{n}
```

某些支持可选参数(optional parameters)的命令, 其参数放置在命令主体之后的方括号[]中. 例如:

```
1 \commandname[option1,option2,...]{argument1}{argument2}...
```

LaTeX 环境

环境在LaTeX中具有和命令相同的格式, 但与命令不同的是, 它会对文件的一部分, 而不是某个字母或者某个参数, 起作用.它的语法如下:

```
1 \begin{environmentname}  
2 text to be influenced  
3 \end{environmentname}
```

在\begin 和 \end 之间, 你可以加入其他的命令, 或者声明另一个环境. 一般来说, 环境也可以接受参数,但这个特性并不常用到, 所以我们把它留到后面再讨论.

LaTeX中的一切都可以用环境和命令的形式表示.

注释

LaTeX源码的注释以%开头, LaTeX将会忽略掉百分号之后一整行的内容以及最后的换行符还有下一行文本前的所有空白.

- 注释可以用于在LaTeX源代码文件中加入一些备忘的事项,而注释中的内容将不会出现在输出的文件中.

```
1 \documentclass{article}  
2 \begin{document}  
3 This is the words to be output %This will be ignored by LaTeX  
4 %You won't see me in the file output.  
5 \end{document}
```

- 请注意%还可以用来分隔很长的单词, 或者说长的,中间不允许加入空格等空白字符的一串字母. 例如:

```
1 \documentclass{article}  
2 \begin{document}  
3 This is an % stupid  
4 % Better: instructive <----  
5 example: Supercal%  
6         ifragilist%  
7 icexpialidocious  
8 |render= This is an example: Supercalifragilisticexpialidocious  
9 \end{document}
```

尽管在源代码中Supercalifragilisticexpialidocious被分成了三行, 但实际上, 在输出的文件中, 它还是以中间没有任何空白字符的完整单词的形式输出.

LaTeX的核心语法并没有预置可以将多行文字一并注释的命令. 如需解决方案请参照 [multi-line comments](#).

輸入檔案架構

当LaTeX编译某个源代码文件时, 它会假定源代码文件遵从特定的结构. 例如, 每一个源代码文件都要以`\documentclass{...}`开始.

这行命令声明了你想要编写的文件的类型(例如`article, book...`), 在这之后, 你可以加入一些配置文档风格的命令, 或者你也可以通过`\usepackage{...}`加载一些宏包来给LaTeX增加一些新的功能.

当所有的这些配置工作都已经完成后, 你可以开始编辑文档的正文部分了. 通过输入`\begin{document}`来表示进入文档正文. 接下来你可以输入文档的正文和一些必要的命令. 当你已经完成文档的编辑之后, 你需要输入`\end{document}`来结束文档的编辑.

命令, 告诉LaTeX每天调用它。 LaTeX将忽略此命令后面的任何内容。 `\pm`和`\mp`之间的区域称为**前导码**。

典型的命令行会话

LaTeX本身没有GUI (图形用户界面), 因为它只是一个程序, 可以处理你的输入文件, 并生成DVI或PDF文件. 某些LaTeX安装具有图形前端, 你可以在其中单击LaTeX来编译输入文件. 在其他系统上可能涉及一些打字, 所以这里是如何诱导LaTeX在基于文本的系统上编译输入文件. 请注意: 此说明假定你已在计算机上安装了可用的LaTeX.

1. 编辑/创建LaTeX输入文件. 该文件必须是纯ASCII文本. 在Unix上, 所有编辑器都会创建这种格式. 在Windows上, 你可能需要确保以ASCII或纯文本格式保存文件. 在为文件选择名称时, 请确保其带有`.tex` 扩展名.
2. 在输入文件上运行LaTeX. 如果成功, 你将获得`.dvi`文件. 可能需要多次运行LaTeX才能获得目录和所有内部引用. 当你的输入文件有错误时, LaTeX会告诉你它并停止处理你的输入文件.

键入`ctrl-D`以返回命令行。

```
latex foo.tex
```

现在你可以查看DVI文件. 在Unix上使用X11, 你可以输入`xdvi foo.dvi`, 在Windows上, 你可以使用名为`yap` 的程序 (另一个预览程序). (现在`evince`和`okular`——许多Linux发行版的标准文档查看器, 都能够查看DVI文件。)

你可以使用 `pdflatex`运行类似的过程, 以从原始的`.tex` 生成PDF文档. 与上面类似, 输入命令:

```
pdflatex foo.tex
```

现在你可以查看PDF文件, `foo.pdf`.

我们的第一份文件

现在我们可以创建我们的第一个文档。我们将会用尽可能少的代码来进行最简单的输出：典型的Hello,World!程序。

- 打开你喜爱的编辑器。如果你使用 `vim`, `emacs` 或 `Notepad++`, LaTeX代码将会自动高亮标记。
- 输入以下LaTeX代码

```
1 \documentclass{article}
2 \begin{document}
3 Hello World!
4 \end{document}
```

- 保存为`hello.tex`.

它代表什麼意思？

<code>% hello.tex - Our first LaTeX example!</code>	The first line is a <i>comment</i> . This is because it begins with the percent symbol (%); when LaTeX sees this, it simply ignores the rest of the line. Comments are useful for people to annotate parts of the source file. For example, you could put information about the author and the date, or whatever you wish.
<code>\documentclass{article}</code>	This line is a command and tells LaTeX to use the <code>article</code> document class. A document class file defines the formatting, which in this case is a generic article format. The handy thing is that if you want to change the appearance of your document, substitute <code>article</code> for another class file that exists.
<code>\begin{document}</code>	This line is the beginning of the environment called <code>document</code> ; it alerts LaTeX that content of the document is about to commence. Anything above this command is known generally to belong in the <i>preamble</i> .
<code>Hello World!</code>	This was the only actual line containing real content - the text that we wanted displayed on the page.
<code>\end{document}</code>	The <code>document</code> environment ends here. It tells LaTeX that the document source is complete, anything after this line will be ignored.

As we have said before, each of the LaTeX commands begin with a backslash (`\`). This is LaTeX's way of knowing that whenever it sees a backslash, to expect some commands. Comments are not classed as a command, since all they tell LaTeX is to ignore the line. Comments never affect the output of the document.

生成文件

It is clearly not going to be the most exciting document you have ever seen, but we want to see it nonetheless. I am assuming that you are at a command prompt, already in the directory where `hello.tex` is stored.

1. Type the command: `latex hello` (the `.tex` extension is not required, although you can include it if you wish)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

```
Output written on hello.dvi (1 page, 232 bytes).
Transcript written on hello.log.
```

This means that your source file has been processed and the resulting document is called *hello.dvi*, which takes up 1 page and 232 bytes of space. This way you created the DVI file, but with the same source file you can create a PDF document. The steps are exactly the same as before, but you have to replace the command `latex` with `pdflatex`:

1. Type the command: `pdflatex hello` (as before, the `.tex` extension is not required)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

```
Output written on hello.pdf (1 page, 5548 bytes).
Transcript written on hello.log.
```

you can notice that the PDF document is bigger than the DVI, even if it contains exactly the same information. The main differences between the DVI and PDF formats are:

- **DVI** 需要更少的磁盘空间，创建起来更快。但它不包括文档中的字体，因此如果你想在另一台计算机上正确查看文档，则必须安装所有必需的字体。它不支持任何交互性，如超链接或动画图像。DVI查看器也不常见，因此你可以考虑在排版时使用它来预览文档。
- **PDF**需要更多的磁盘空间并且创建速度较慢，但它包含文档中所有必需的字体，因此你不会遇到任何可移植性问题。它支持内部和外部超链接。它还支持高级印刷功能：悬挂标点符号，字体扩展和边距字距调整，从而为TeX引擎提供更大的灵活性和更好的输出效果。如今它是共享和发布文档的事实标准，因此你可以考虑将其用于文档的最终版本。

About now, you saw you can create both DVI and PDF document from the same source. This is true, but it gets a bit more complicated if you want to introduce images or links. This will be explained in detail in the next chapters, but for now assume you can compile in both DVI and PDF without any problem.

Note, in this instance, due to the simplicity of the file, you only need to run the LaTeX command once. However, if you begin to create complex documents, including bibliographies and cross-references, etc, LaTeX needs to be executed multiple times to resolve the references. But this will be discussed in the future when it comes up.

取自“<https://zh.wikibooks.org/w/index.php?title=LaTeX/初学者必读&oldid=125032>”

此页面最后编辑于2019年8月14日 (星期三) 06:02。

本站的全部文字在創用CC 姓名標示-相同方式分享 3.0 協議之條款下提供，附加條款亦可能應用（請參閱使用條款）。Wikibooks®和維基教科書標誌是維基媒體基金會的註冊商標；維基™是維基媒體基金會的商標。維基媒體基金會是在美國佛羅里達州登記的501(c)(3)免稅、非營利的慈善機構。