

# LaTeX/生成封面和标题

维基教科书，自由的教学读本  
< [LaTeX](#)

很多情况下，用户可能希望创建自己的标题格式，而非沿用LaTeX预设的格式。对于一些如基本文章(basic articles)这样的短文档，`\maketitle` 通常是满足要求的，但是对于书籍和报告这类长文档，经常会涉及到更多的格式要求。虽然可以改变`\maketitle`的输出，但是这一方式，即使对标题进行少量的修改可能都是一个复杂的过程。这时通过scratch创建标题通常是更好的选择, 本部分我们对如何实现这一工作进行说明。

## 目录

- 標準標題頁面
- 提交期刊的標題
- 自訂標題頁面
  - 建立標題
  - 實例
  - 整合標題頁
- 自定义标题包
- 更多标题页面的例子
- 注释和参考

## 標準標題頁面

多数文档类型提供了简单界面，在这一界面中存储了标题显示以及排版的细节。标准的类型只提供了四种 *storing* 命令 (`\title`, `\author` `\thanks` and `\date`)。用户可以存储希望在标题中显示的“任意”信息，包括格式。

实际标题的排版由命令 `\maketitle`来进行排版。 具体的布局由使用的文档类型来决定。

```
\documentclass{article}% use option titlepage to
get the title on a page of its own.
\usepackage{blindtext}
\title{The Triangulation of Titling Data in Non-
Linear Gaussian Fashion via  $\rho$  Series\thanks{No
procrastination}}
\date{2017\textbackslash December}
\author{John Doe\textbackslash Magic Department\thanks{I am no
longer a member of this department}, Richard Miles
University
\and Richard Row, \LaTeX\ Academy}
\begin{document}
\maketitle
\section{Introduction}
\blindtext
\end{document}
```

# The Triangulation of Titling Data in Non-Linear Gaussian Fashion via $\rho$ Series\*

John Doe  
Magic Department, Richard Miles University  
Richard Row, L<sup>A</sup>T<sub>E</sub>X Academy

2017  
December

## 1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus telus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

---

\*No procrastination  
†I am no longer a member of this department

`\thanks` 命令会存储内容，并生成与标题对应的脚注。正如其命令名称所显示的，该命令可以用来对某人致谢。也可以在脚注中打印电子邮件地址或者类似的内容。

作者名字使用`\and`命令分隔，允许作者姓名并列输出。在上例中，由于横向空间不足，作者名字无法在同一行显示。

如果`\date`命令没有定义，Latex会输出当前日期。如果不希望显示日期，使用`\date{}`命令，该命令存储的是空的字符串。

存储标题数据的命令可以在文档开始时使用，因为`\maketitle`并不实际输出，所以应该在`\begin{document}`命令“之后”使用。一般来说，标题占据着文档的首位。

请注意下面对应 KOMA-script 和 memoir 文件类别的例子。两者均给出了（不同）的命令来改变标题的形式。后面会介绍如何完全设计自定义的标题页。

```
\documentclass{scrbook}
\setkomafont{author}{\scshape}
\usepackage{blindtext}
\title{How hard would it be to build a spaceship
from scrap}
\author{Carl Capybara\thanks{I never procrastinate}
and Walter Wombat}
\subtitle{A closer look at the expenses}
\subject{a funny paper}
\begin{document}
\maketitle
\addchap{Introduction}
\blindtext
\end{document}
```

a funny paper

## How hard would it be to build a spaceship from scrap

A closer look at the expenses

CARL CAPYBARA\*      WALTER WOMBAT

December 10, 2017

\*I never procrastinate

```

\documentclass{memoir}% use option titlepage to get
the title on a page of its own.
\usepackage{blindtext}
\title{The influence of colour on the floating
velocity of rubber ducks}
\author{Peter Piranha}
\renewcommand{\maketitlehookb}{\centering You won't
expect the results}
\begin{document}
\maketitle
\chapter{Introduction}
\blindtext
\end{document}

```

## The influence of colour on the floating velocity of rubber ducks

You won't expect the results

Peter Piranha

December 10, 2017

同样的，类文档相比较可能的命令而言，可以呈现更为丰富的细节。

## 提交期刊的标题

期刊对应一类特定的板式。为了保证这一板式，期刊经常会提供定义其板式的模版。标题的样式(如电子邮箱，机构名称，关键词)严重的依赖模版，而且不同的期刊之间差异可能很大。如果期刊提供了模版，那么需要按照模版的要求进行排版。如果期刊没有提供对应的模版，那么需要作者使用上面提到的那些更为基础的Latex标题的一些概念。

## 自订标题页面

### 建立标题

Normally, the benefit of using LaTeX instead of traditional word processing programs is that LaTeX frees you to concentrate on content by handling margins, justification, and other typesetting concerns. On the other hand, if you want to write your own title format, it is exactly the opposite: you have to take care of everything--this time LaTeX will do nothing to help you. It can be challenging to create your own title format since LaTeX was not designed to be graphically interactive in the adjustment of layout. The process is similar to working with raw HTML with the added step that each time you want to see how your changes look, you have to re-compile the source. While this may seem like a major inconvenience, the benefit is that once the customized title format has been written, it serves as a template for all other documents that would use the title format you have just made. In other words, once you have a layout you like, you can use it for any other documents where you would like the same layout without any additional fiddling with layout.

First step: since you'll be working only on the first page of your document and you'll have to compile very often, you don't have to compile the whole document each time, you only need to take a look at the first page. That is why we'll first create a dummy document for preparing the title and then we'll simply include it within the existing big document we are writing. Call the dummy document `test_title.tex` and put the following code in it:

```

\documentclass[12pt,a4paper]{report}
\usepackage{graphicx}
\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}
\begin{document}
\input{./title.tex}
\end{document}

```

It is meant to be compiled with `pdflatex` to create a PDF in output. It is a very basic document, but take care that it has the same settings of the document you are writing, so the output won't change when you include the title in your document. In this case (see the first line) the font size is set to 12pt and the paper size is an A4. The package `graphicx` is included to insert an image in the title. Then a command is defined called `\HRule`; it will just insert a horizontal line whose length is like the size of the paper and whose thickness is 0.5 mm. If you want you can change its settings in the definition. Finally the document starts and it simply includes the `title.tex` file, that must be placed in the same directory of our dummy file `test_title.tex`.

Now create the `title.tex` and write in it:

```
\begin{titlepage}

\end{titlepage}
```

all the things you want to put in the title must be inside the `titlepage` environment. Now if you compile `test_title.tex` you will see a preview of your title in the `test_title.pdf` file. Here is what you need to know to write your title:

### Alignment

if you want to center some text just use `\begin{center} ... \end{center}`. If you want to align it differently you can use the environment `flushright` for right-alignment and `flushleft` for left alignment.

### Images

the command for including images is the following (the example is for a small logo, but you can introduce any image of any size): `\includegraphics[width=0.15\textwidth]{./logo}`. There is no `\begin{figure}` as you usually do because you don't want it to be floating, you just want it there where you placed it. When handling it, remember that it is considered like a big box by the TeX engine.

### Text size

If you want to change the size of some text just place it within brackets, *{like this}*, and you can use the following commands (in order of size): `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\small`, `\footnotesize`, `\tiny`. So for example:

```
{\large this text is slightly bigger than normal}, this one is not
```

`\normalsize` is used to create text at the default size for the document.

### New lines

you can force the start of a new line by `\\`. If you want to add more vertical space you don't need to use several new-line commands, just insert some vertical space. For example, this way `\\[1cm]` you start a new line after having left 1 cm of empty space.

### 日期

you can insert the date of the current day with the command `\today`. If you do not wish to insert any date, keep it blank e.g. `\date{}`

### Filling the page

the command `\vfill` keeps on adding empty spaces until the page is full. If you put it in the page, you are sure that all the following text will be placed at the bottom of the page.

## 實例

All these tips might have made you confused. Then, here is a practical example. Get the `test_title.tex` described above and here is an example of a `title.tex`. On the right you can see the output after you compile `test_title.tex` in PDF:

```
\begin{titlepage}
\begin{center}

% Upper part of the page
\includegraphics[width=0.15\textwidth]{./logo}\\[1cm]
\textsc{\LARGE University of Beer}\\[1.5cm]
\textsc{\Large Final year project}\\[0.5cm]

% Title
\hrule \\[0.4cm]
{ \huge \bfseries Lager brewing techniques}\\[0.4cm]
\hrule \\[1.5cm]

% Author and supervisor
\begin{minipage}{0.4\textwidth}
\begin{flushleft} \large
\emph{Author:}\\
John \textsc{Smith}
\end{flushleft}
\end{minipage}
\begin{minipage}{0.4\textwidth}
\begin{flushright} \large
\emph{Supervisor:} \\
Dr.-Mark \textsc{Brown}
\end{flushright}
\end{minipage}

\vfill

% Bottom of the page
{\large \today}

\end{center}
\end{titlepage}
```



UNIVERSITY OF BEER

FINAL YEAR PROJECT

---

## Lager brewing techniques

---

*Author:*  
John SMITH

*Supervisor:*  
Dr. Mark BROWN

September 12, 2006

The picture is from a file called `logo.png` that is in the same directory of both `title.tex` and `test_title.tex`. Since I wanted to insert both the author and supervisor names properly aligned I used a trick: I created two small minipages, one on left and one on the right. Their width is a bit less than half of page width (as you can see, they are exactly 40% of the text width). Within the minipages I have used different alignments. Using `\vfill` I could write the date exactly at the bottom of the page.

As you can see, the code looks "dirtier" than standard LaTeX source because you have to take care of the output as well. If you start changing font's output it will get more confused, but you can do it: it's only for the title and your complicated code will be isolated from all the rest within its own file `title.tex`.

### 整合標題頁

Assuming that your title page is now contained in a file named `title.tex`, it must be placed in the same directory as the main document. In order to integrate it, the `\input` command must be used by placing `\input{./title.tex}` at the top of the document. Don't forget to add the commands `\usepackage{graphicx}` and `\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}` in the preamble section as well.

For example, the top section of your document would look like:

```
...
\usepackage{graphicx}
\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}
\begin{document}
\input{./title.tex}
\tableofcontents
...
```

# 自定义标题包

宏包`titling`<sup>[1]</sup>提供了一种方式，来控制 `\maketitle` 和 `\thanks` 命令的排版。它在仅对标准的输出做微小更改时很有用。

意大利用户也许还想要使用`frontespizio`<sup>[2]</sup>宏包。它定义了在意大利使用的卷首图片。

宏包`authblk`<sup>[3]</sup>提供了一种新的排版作者的方式，这在向没有提供模板的期刊提交文稿时特别有用。

## 更多标题页面的例子

The `titlepages` package presents many different styles for title pages.

TeX.SE (<http://tex.stackexchange.com/questions/85904/showcase-of-beautiful-title-page-done-in-tex>) has a collection of titlepages.

Another small collection can be found on [Github \(https://github.com/johannesbottcher/titlepageExamples/\)](https://github.com/johannesbottcher/titlepageExamples/).

## 注释和参考

1. [Titling package webpage on CTAN \(http://www.ctan.org/pkg/titling\)](http://www.ctan.org/pkg/titling)
2. [Frontespizio package webpage on CTAN \(http://www.ctan.org/pkg/frontespizio\)](http://www.ctan.org/pkg/frontespizio)
3. [authblk package webpage on CTAN \(http://www.ctan.org/pkg/authblk\)](http://www.ctan.org/pkg/authblk)

---

取自“<https://zh.wikibooks.org/w/index.php?title=LaTeX/生成封面和标题&oldid=119584>”

---

此页面最后编辑于2019年3月13日 (星期三) 14:49。

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。