

### 3 Macros for Box Creation 创建盒子的宏命令

```
\begin{tcolorbox}[\langle options \rangle]
  \langle environment content \rangle
\end{tcolorbox}
```

This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts. The appearance of this box is controlled by numerous options. In the most simple case:

这是创建一个带有圆角、强调颜色的文本框的主要环境，并且可以选择是否分为两个部分。这个文本框的外观可以通过众多选项进行控制。在最简单的情况下

源代码为

编译后创建以下的文本框

```
\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

The text content of the box can be divided in an upper and a lower part by the command `\tcblower`. Visually, both parts are separated by a line. For example:

该文本框的内容可以通过命令 `\tcblower` 分为 upper 部分和 lower 部分。在视觉上，upper 和 lower 用一条线隔开。例如：

upper 和 lower 用一条线隔开

```
\begin{tcolorbox}
这是另一个 \textbf{tcolorbox}.
\tcblower
在这里，您会看到盒子的lower部。
\end{tcolorbox}
```

这是另一个 **tcolorbox**.

在这里，您会看到盒子的 lower 部。

The `\langle options \rangle` control the appearance and several functions of the boxes, see ?? on page ?? for the complete list. A quick example is given here:

`\langle options \rangle` 控制盒子的外观和其他功能，请参阅 ?? on page ?? 以获取完整列表。下面给出了一个快速示例：

```
\begin{tcolorbox}[colback=red!5!white,colframe=red!75!black,title=My nice heading]
这是另一个 \textbf{tcolorbox}.
\tcblower
在这里，您会看到盒子的lower部。
\end{tcolorbox}
```

My nice heading

这是另一个 **tcolorbox**.

在这里，您会看到盒子的 lower 部。

#### `\tcblower`

Used inside `tcolorbox` to separate the upper box part from the optional lower box part. The upper and the lower part are treated as separate functional units. If you only want to draw a line, see `\tcblines`<sup>P. ??</sup>.

在 `tcolorbox` 中用于将 upper 与可选的 lower 分离。upper 和 lower 被视为独立的功能

单元。如果你只想画一条线，请参见 `\tcbline`<sup>→P.??</sup>。

我们试试同时有 `tcbline` 和 `tcblower` 的情况

virhuiai

```
\begin{tcolorbox}
第1部分\_1
\tcbline
第1部分\_2
\tcblower
第2部分
\end{tcolorbox}
```

第 1 部分 \_1

第 1 部分 \_2

第 2 部分

### `\tcbset{⟨options⟩}`

Sets options for every following `tcolorbox`<sup>→P.1</sup> inside the current  $\text{\TeX}$  group. By default, this does not apply to nested boxes, see ?? on page ??.

设置当前的  $\text{\TeX}$  组中后续 `tcolorbox` 的选项。默认情况下，这不适用于嵌套的盒子，请参见 ?? on page ??。

For example, the colors of the boxes may be defined for the whole document by this:  
例如，整个文档的盒子的颜色可以通过以下方式定义：

```
\tcbset{colback=red!5!white,colframe=red!75!black}
```

### `\tcbsetforeverylayer{⟨options⟩}`

Sets options for every following `tcolorbox`<sup>→P.1</sup> inside the current  $\text{\TeX}$  group. In contrast to `\tcbset`, this does also apply to nested boxes, see ?? on page ??. Technically, the `⟨options⟩` are appended to the default values for every `tcolorbox` which are applied by `/tcb/reset`<sup>→P.??</sup>.

在当前的  $\text{\TeX}$  组中为每个后续的 `tcolorbox`<sup>→P.1</sup> 设置选项。与 `\tcbset` 不同，它也适用于嵌套的盒子，请参阅 ?? on page ??。从技术上讲，`⟨options⟩` 将附加到每个通过 `/tcb/reset`<sup>→P.??</sup> 应用的 `tcolorbox` 的默认值之后。

`\tcbsetforeverylayer` 设置的选项，不受 `reset` 的影响。

—virhuiai

You should not use this macro, if you are not completely sure that you want to have the `⟨options⟩` also for boxes in boxes (in boxes in boxes ...).

如果您不确定是否希望在嵌套的盒子中（盒子中的盒子中的盒子...）也使用 `⟨options⟩`，请不要使用此宏。

```

\begin{tcolorbox}[title=virhuiai家的盒子]
默认盒子的内容
\end{tcolorbox}

\tcbset{colback=green!10!white}
\tcbsetforeverylayer{%
colframe=red!75!black}

\begin{tcolorbox}%
[title=外层的盒子]
\cs{tcbset}设置有效\par
\cs{tcbsetforeverylayer}设置有效
\begin{tcolorbox}[title=嵌套的盒子]
\cs{tcbset}设置【无效】\
\cs{tcbsetforeverylayer}有效
\end{tcolorbox}
\end{tcolorbox}
\bigskip

\begin{tcolorbox}[reset,%
title={reset的盒子}]
|\tcbsetforeverylayer|%
的设置保持!
\end{tcolorbox}

\begin{tcolorbox}[title=默认盒子]
绿色的背景色
\end{tcolorbox}

```

virhuiai 家的盒子

默认盒子的内容

外层的盒子

\tcbset 设置有效  
\tcbsetforeverylayer 设置有效

嵌套的盒子

\tcbset 设置【无效】  
\tcbsetforeverylayer 有效

reset 的盒子

\tcbsetforeverylayer 的设置保持!

默认盒子

绿色的背景色

**\tcbbox**[*<options>*]{*<box content>*}

Creates a colored box which is fitted to the width of the given *<box content>*. In principle, most *<options>* for a **tcolorbox**<sup>→P.1</sup> can be used for \tcbbox with some restrictions. A \tcbbox cannot have a lower part and cannot be broken.

创建一个彩色的盒子，它适应于给定的 *<box content>* 的宽度。原则上，大多数 **tcolorbox**<sup>→P.1</sup> 的 *<选项>* 都可以用于 \tcbbox，但有一些限制。 \tcbbox 不能有 lower 部分，也不可分割。

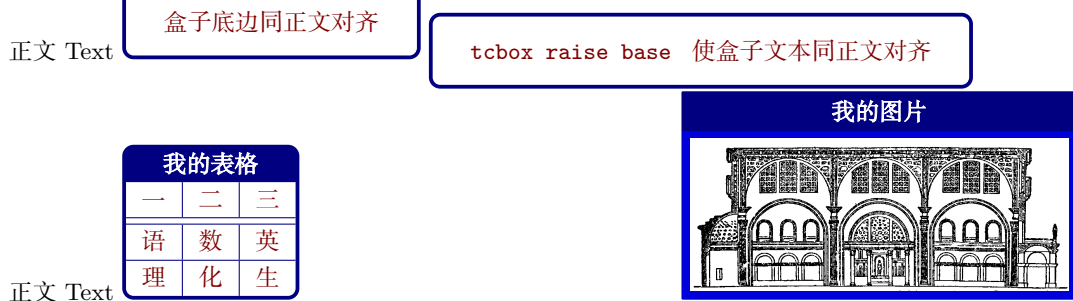
```

\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
fonttitle=\bfseries,nobeforeafter,center title}

正文Text \tcbbox{盒子底边同正文对齐} %
\tcbbox[tcbbox raise base]{\tt tcbox raise base } 使盒子文本同正文对齐}%\hfill

正文Text
\tcbbox[left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=0mm,
toptitle=0.5mm,bottomtitle=0.5mm,title=我的表格]{%
\arrayrulecolor{blue!50!black}\renewcommand{\arraystretch}{1.2}%
\begin{tabular}{r|c|l}
一 & 二 & 三 \\\hline\hline
语 & 数 & 英 \\\hline
理 & 化 & 生
\end{tabular}}\hfill
%
\tcbbox[colback=blue!85!black,
left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=1mm,arc=0mm,boxrule=0.5pt,
title=我的图片]{%
\includegraphics[width=5cm]{Basilica_5.png}}

```



### 例. 3.1: 一个最简单的使用 tcolorbox 的中文例子

—virhuiai

```

\PassOptionsToPackage{no-math}{fontspec}
\PassOptionsToPackage{AutoFakeBold=true,AutoFakeSlant=true}{xeCJK}
\documentclass{book}
\usepackage[heading=true,scheme=chinese,space=auto]{ctex}

\usepackage[all]{tcolorbox}
\begin{document}
\thispagestyle{empty}
\tcbset{nobeforeafter}
Text \tcbbox{你好} \tcbbox[tcbbox raise
base]{世界}\hfill
\end{document}

```



```
% \usepackage{tikz}
\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
fonttitle=\bfseries,center title}

% 固定宽度的盒子
\begin{tcolorbox}tcolorbox中\\可以使用换行命令!\end{tcolorbox}

% 宽度自适应盒子 (类似 hbox 、 makebox)
\tcbbox{tcbbox中\\换行命令\newline 无效的情况!}

% 宽度自适应盒子 (使用 \tikzname\ node)
\tcbbox[tikznode]{你好\\tikznode可以换行!}
```

tcolorbox 中  
可以使用换行命令!

tcbbox 中换行命令无效的情况!

你好  
tikznode 可以换行!

! See ?? on page ?? and ?? on page ?? for more elaborate methods to create new environments and commands.

参见 ?? on page ?? 和 ?? on page ?? 了解更多创建新环境和命令的方法。

`\newtcolorbox[<init options>]{<name>}[<number>][<default>]{<options>}`

Creates a new environment *<name>* based on `tcolorbox`<sup>P.1</sup>. This means, the new environment *<name>* optionally takes *<number>* arguments, where *<default>* is the default value for the optional first argument. The *<options>* are given to the underlying `tcolorbox`. Note that `/tcb/savedelimiter`<sup>P.??</sup> is set to the given *<name>* automatically. The *<init options>* allow setting up automatic numbering, see Section ?? from page ??.

基于 `tcolorbox`<sup>P.1</sup> 创建一个新的环境 *<name>*。这意味着, 新环境 *<name>* 可以选择性地使用 *<number>* 个参数, 其中 *<default>* 是可选第一个参数的默认值。*<options>* 是提供给底层 `tcolorbox` 的选项。请注意, `/tcb/savedelimiter`<sup>P.??</sup> 会自动设置为给定的 *<name>*。*<init options>* 可用于设置自动编号, 请参见第??页的第??节。

在 L<sup>A</sup>T<sub>E</sub>X 中, 命令的参数通常按照从左到右的顺序给出。因此, 如果一个命令有多个参数, 并且其中某个参数有一个默认值, 通常将默认值放在参数列表的第一个位置。

这种设计选择的一个原因是为了方便使用。在大多数情况下, 用户只需要提供必要的参数, 而不需要显式地提供默认值。将默认值放在第一个参数的位置上, 使得在使用命令时可以更加简洁和直观, 因为用户只需要提供额外的参数, 而无需关心默认值。

另外, 将默认值放在第一个参数的位置也有助于确保命令的一致性和可读性。当多个参数具有默认值时, 将它们放在参数列表的相同位置可以使命令的使用方式更加统一和一致。这样, 用户可以更容易地理解和记忆命令的参数结构。

—virhuiai

## 只指定了两色颜色属性的例子

```
\newcolorbox{mybox}{colback=red!5!white,colframe=red!75!black}
\begin{mybox}
这是我的盒子。
\end{mybox}
```

这是我的盒子。

## 定义一个带有必选参数的环境

```
\newcolorbox{mybox}[1]{colback=red!5!white,
colframe=red!75!black,fonttitle=\bfseries,
title={#1}}
\begin{mybox}{你好呀}
这是我定义的盒子，带有必选的标题参数。%
\footnote{dispExample中定义的盒子，%
在环境外失效。} \hfill ---virhuiai}
\end{mybox}
```

你好呀

这是我定义的盒子，带有必选的标题参数。<sup>a</sup>

<sup>a</sup>dispExample 中定义的盒子，在环境外失效。——virhuiai

## 定义一个带有可选参数的环境

```
\newcolorbox{mybox}[2][colback=green]{colback=red!5!white,
colframe=red!75!black,fonttitle=\bfseries,
colbacktitle=red!85!black,enhanced,
attach boxed title to top center={yshift=-2mm},
title={#2},#1}

\begin{mybox}{Hello there}
这是我定义的盒子，带有必选的标题参数和%
可选的配置参数。
\end{mybox}

\begin{mybox}[colback=yellow]{Hello there}
这是我定义的盒子，带有必选的标题参数和%
可选的配置参数。
\end{mybox}
```

Hello there

这是我定义的盒子，带有必选的标题参数和可选的配置参数。

Hello there

这是我定义的盒子，带有必选的标题参数和可选的配置参数。

## 导言中的定义：

```
\newcolorbox[auto counter,number within=section]{pabox}[2][{%
colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
title=例~\thetcbcounter: #2,#1}
```

## 译者对上面这个盒子的分析

上面由一条命令生成：

```
\inputpreamblelisting{A}
```

这条命令定义为：

```
\newcommand{\inputpreamblelisting}[1]{%
\tcbinputlisting{title= 导言中的定义:,
base example,coltitle=black,fonttitle=\itshape,titlerule=0pt,
colbacktitle=Navy!15!ExampleBack,
top=0mm,
%before=\par\smallskip,%
before skip balanced=4pt plus 2pt minus 1pt,
after skip balanced=5pt plus 2pt minus 1pt,
listing style=mydocumentation,
listing only,listing file={\jobname_preamble_#1.tex}}%
```

```
}
```

可以看出代码内容来自 `\jobname_preamble_A.tex`<sup>1</sup>:

```
\begin{tcbverbatimwrite}{\jobname_preamble_A.tex}
\newtcolorbox[auto counter,number within=section]{pabox}[2][\%
colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
title=Examp.-\thetcbcounter: #2,#1]
\end{tcbverbatimwrite}
```

```
\begin{pabox}[colback=yellow]{你好哇! }
这是我的盒子，带一个可选参数%
和一个必填的带有编号的标题。
\end{pabox}
```

**例 3.1: 你好哇!**

这是我的盒子，带一个可选参数和一个必填的带有编号的标题。

```
\renewtcolorbox[<init options>]{<name>}[<number>][<default>]{<options>}
```

An existing environment is redefined.

操作类似 `\newtcolorbox`<sup>→ P.5</sup>，一个已存在的环境被重新定义。

```
\newtcbbox[<init options>]{<name>}[<number>][<default>]{<options>}
```

Creates a new macro `\<name>` based on `\tcbbox`<sup>→ P.3</sup>. The new macro `\<name>` optionally takes `<number>` arguments, where `<default>` is the default value for the optional first argument. The `<options>` are given to the underlying `tcbbox`. The `<init options>` allow setting up automatic numbering, see Section ?? from page ??.

基于 `\tcbbox`<sup>→ P.3</sup> 创建一个新的宏 `\<name>`。新的宏 `\<name>` 可选地接受 `<number>` 个参数，其中 `<default>` 是可选第一个参数的默认值。`<options>` 应用于底层的 `tcbbox`。`<init options>` 可用于设置自动编号，参见第??页的第??节。

```
\newtcbbox{\mybox}{colback=red!5!white,
colframe=red!75!black}

\mybox{这是我的例子。}
```

这是我的例子。

```
\newtcbbox{\mybox}[1]{colback=red!5!white,
colframe=red!75!black,fonttitle=\bfseries,
title={#1}}

\mybox{必选的标题}{这是我的盒子。}
```

**必选的标题**

这是我的盒子。

```
\newtcbbox{\mybox}[2][colback=green]%
{colback=red!5!white,
colframe=red!75!black,fonttitle=\bfseries,
title={#2},#1}

\mybox{Hello there}%
{首参可选，次参必填!}

\mybox[colback=yellow]{Hello there}%
{首参可选，次参必填!}
```

**Hello there**

首参可选，次参必填!

**Hello there**

首参可选，次参必填!

<sup>1</sup>`\jobname` 的值就是当前被编译的文件名 `tcolorbox`。

### 导言中的定义:

```
% counter from previous example
\newtcbox[use counter from=pabox]{\pabox}[2][\%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=(\thetcbcounter) #2,#1}
```

```
\pabox[colback=yellow]{Hello there}%
{标题中使用pabox的计数器。}
```

(3.2) Hello there

标题中使用 pabox 的计数器。

```
\newtcbox{\mybox}[1][red]{on line,
arc=0pt,outer arc=0pt,
colback=#1!10!white,colframe=#1!50!black,
boxsep=0pt,
left=1pt,right=1pt,top=2pt,bottom=2pt,
boxrule=0pt,bottomrule=1pt,toprule=1pt%
}
\newtcbox{\xmybox}[1][red]{on line,
arc=7pt,
colback=#1!10!white,colframe=#1!50!black,
before upper={\rule[-3pt]{0pt}{10pt}},%简单实现增高
boxrule=1pt,
boxsep=0pt,
left=6pt,right=6pt,top=2pt,bottom=2pt}
```

The \mybox[green]{quick} brown \mybox{fox} \mybox[blue]{jumps} over the  
 \mybox[green]{lazy} \mybox{dog}. \par  
 The \xmybox[green]{quick} brown \xmybox{fox} \xmybox[blue]{jumps} over the  
 \xmybox[green]{lazy} \xmybox{dog}.

The quick brown fox jumps over the lazy dog.  
 The quick brown fox jumps over the lazy dog.

**\renewtcbox**[*<init options>*]{*<name>*}[*<number>*][*<default>*]{*<options>*}

Operates like **\newtcbox**<sup>P.7</sup>, but based on **\renewcommand** instead of **\newcommand**. An existing macro is redefined.

类似于 **\newtcbox**<sup>P.7</sup>, 基于 **\renewcommand** 重新定义一个已经存在的宏。

N 2014-10-20

**\tcolorboxenvironment**{*<name>*}{*<options>*}

An existing environment *<name>* is redefined to be boxed inside a **tcolorbox** with the given *<options>*.

现有的环境 *<name>* 被重新定义为包含在给定 *<options>* 的 **tcolorbox** 中。



```
% tcbuselibrary{skins}
\newenvironment{myitemize}{%
\begin{itemize}}{\end{itemize}}
```

```
%blanker选项表示在环境内部的上下两端不添加任何额外的空白
\tcOLORBOXenvironment{myitemize}{blanker,
%before skip和after skip选项分别表示在环境前后添加6pt的垂直空白。
before skip=6pt,after skip=6pt,
% 西边（左）的线
borderline west={3mm}{0pt}{red}
}
```

一些文本。

```
\begin{myitemize}
\item 甲
\item 乙
\item 丙
\end{myitemize}
```

更多的文本。

一些文本。

- 甲
- 乙
- 丙

更多的文本。

See further examples in ?? on page ??.

请参见?? on page ??中的其他例子。