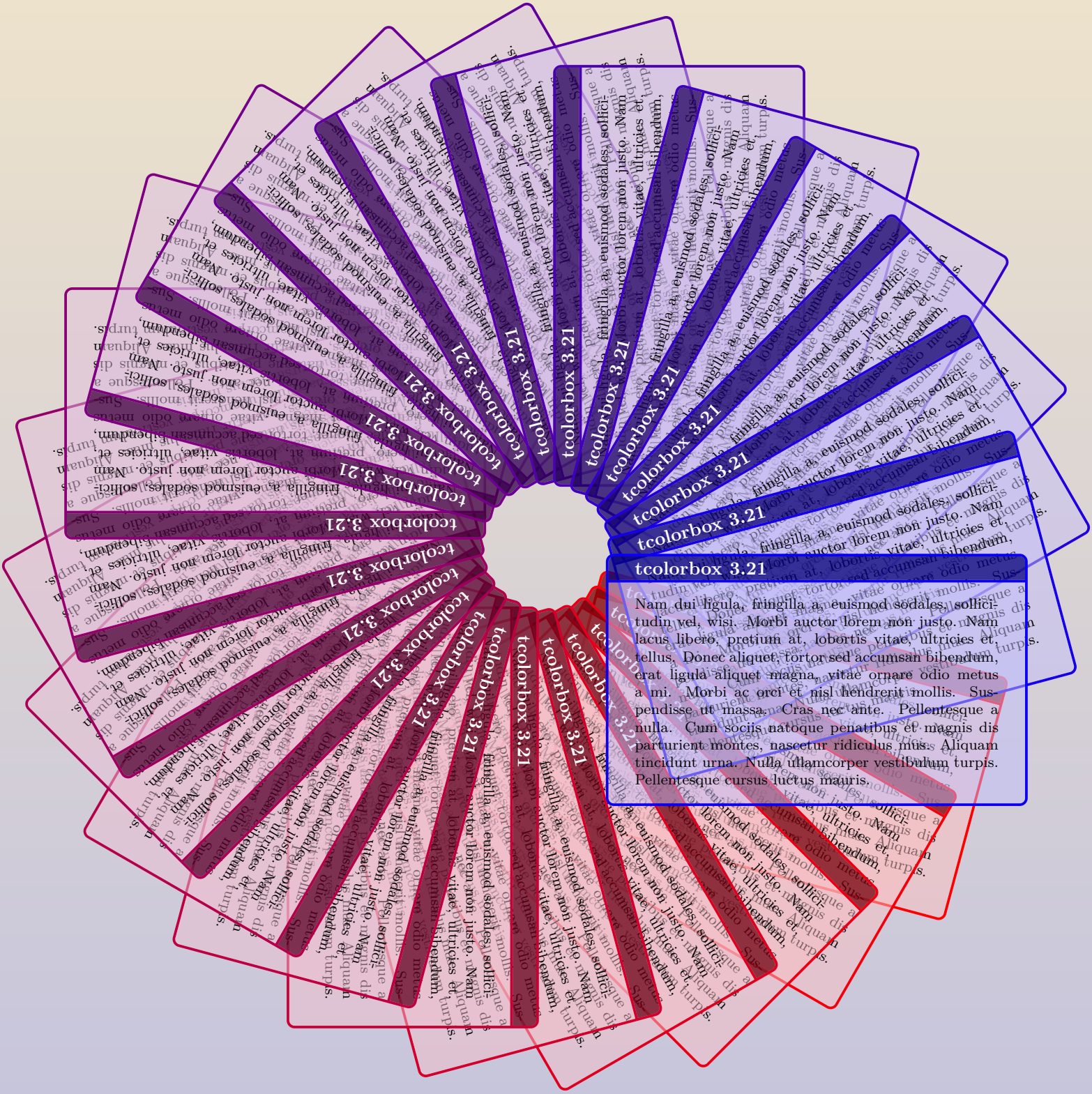
tcolorbox



Manual for version 3.21 (2014/10/10)

Cover code

```
% \usepackage{incgraph}
\begin{inctext}
\begin{tikzpicture}
\definecolorseries{boxcol}{rgb}{last}{red}{blue}
\resetcolorseries[24]{boxcol}
\coordinate (A) at (0,0); \coordinate (B) at (21,29.7);
\path[use as bounding box,top color=Goldenrod!25!white,bottom color=Navy!25!white]
  (A) rectangle coordinate (C) (B);
\foreach \w in {-15,-30,...,-360}
{\node[below right,rotate around={\w:(C)}] at ([xshift=1cm]C) {%
  \begin{tcolorbox}[width=8cm,title=tcolorbox \version,
    fontupper=\scriptsize,fonttitle=\small\bfseries,
    colback=boxcol!25!white,colbacktitle=boxcol!!+!50!black,colframe=boxcol,
    enhanced jigsaw,opacityback=0.5,opacitybacktitle=0.75]
    \lipsum[2]
  \end{tcolorbox}};
}
\node[font=\Huge\bfseries] at ([yshift=12cm]C) {|tcolorbox|};
\node[font=\large\bfseries] at ([yshift=-12cm]C)
  {Manual for version \version\ (\datum)};
\end{tikzpicture}
\end{inctext}
```

If you have trouble printing this document, the reason is quite likely the cover page. Printing the pages starting with page 2 or page 3 should work.

The `tcolorbox` package

Manual for version 3.21 (2014/10/10)

Thomas F. Sturm[1]

Abstract

`tcolorbox` provides an environment for colored and framed text boxes with a heading line. Optionally, such a box can be split in an upper and a lower part. The package `tcolorbox` can be used for the setting of LaTeX examples where one part of the box displays the source code and the other part shows the output. Another common use case is the setting of theorems. The package supports saving and reuse of source code and text parts.

Contents

1 Introduction **6**
- 1.1 Installation . . . 6
- 1.2 Loading the Package . . . 6
- 1.3 Libraries . . . 7

2 Macros for Box Creation **9**

3 Option Keys **14**
- 3.1 Title . . . 14
- 3.2 Subtitle . . . 16
- 3.3 Lower Part . . . 17
- 3.4 Colors and Fonts . . . 20
- 3.5 Geometry . . . 23
- 3.6 Corners . . . 34
- 3.7 Transparency . . . 37
- 3.8 Height Control . . . 39
- 3.9 Box Content Additions . . . 46
- 3.10 Overlays . . . 55
- 3.11 Floating Objects . . . 60
- 3.12 Side by Side . . . 62
- 3.13 Embedding into the Surroundings . . . 65
- 3.14 Bounding Box . . . 67
- 3.15 Layered Boxes and Every Box Settings . . . 73
- 3.16 Capture Mode . . . 76
- 3.17 Text Characteristics . . . 77
- 3.18 Files . . . 78
- 3.19 `\tcbox` Specials . . . 78
- 3.20 Counters, Labels, and References . . . 79
- 3.21 Externalization . . . 81
- 3.22 Miscellaneous . . . 81

[1]Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, Universität der Bundeswehr München, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

4 Initialization Option Keys **82**
4.1 Numbered Boxes . . . 82
4.2 Lists of `tcolorboxes` . . . 87

5 Saving and Loading of Verbatim Texts **88**

6 Technical Overview and Customization **89**
6.1 Skins and Drawing Engines . . . 89
6.2 Code Option Keys . . . 93
6.3 Subskins . . . 96
6.4 Drawing Scheme . . . 97

7 Library LIB skins **101**
7.1 Style Option Keys . . . 101
7.2 Boxed Title Option Keys . . . 107
7.3 Watermark Option Keys . . . 114
7.4 Clip Environments . . . 121
7.5 Border Line Option Keys . . . 126
7.6 Shadow Option Keys . . . 129
7.7 Ti*k*Z Picture Option Keys . . . 138
7.8 Underlay Option Keys . . . 141
7.9 Finish Option Keys . . . 143
7.10 Jigsaw Skin Variants . . . 145
7.11 Draft Mode . . . 147
7.12 Skin Family 'standard' . . . 148
7.13 Skin Family 'enhanced' . . . 150
7.14 Skin Family 'bicolor' . . . 163
7.15 Skin Family 'beamer' . . . 168
7.16 Skin Family 'widget' . . . 173
7.17 Skin Family 'empty' . . . 177
7.18 Skin 'spartan' . . . 186
7.19 Skin 'draft' . . . 187
7.20 Skin Family 'freelance' . . . 189

8 Ti*k*Z Image and Picture Fill Extensions; Auxiliary Macros **190**
8.1 Fill Plain . . . 190
8.2 Fill Stretch . . . 191
8.3 Fill Overzoom . . . 192
8.4 Fill Zoom . . . 193
8.5 Fill Shrink . . . 194
8.6 Fill Tile . . . 195
8.7 Filling Options . . . 196
8.8 Extracting Node Dimensions . . . 198

9 Libraries LIB listings, LIB listingsutf8, and LIB minted **199**
9.1 Loading the Libraries . . . 199
9.1.1 Loading LIB listings . . . 199
9.1.2 Loading LIB listingsutf8 . . . 199
9.1.3 Loading LIB minted . . . 200
9.2 Common Macros of the Libraries . . . 200
9.3 Option Keys of the LIB listings Library . . . 206

Contents

9.4 Option Keys of the LIB listingsutf8 Library . . . 208
9.5 Option Keys of the LIB minted Library . . . 209
9.6 Common Option Keys of all Libraries . . . 211
9.7 Creation of LaTeX Tutorials . . . 218
9.8 Creation of LaTeX Exercises . . . 225
9.9 List of Exercises . . . 228
9.10 Solutions for the given LaTeX Exercises . . . 229

10 Library LIB theorems 231
10.1 Macros of the Library . . . 231
10.2 Option Keys of the Library . . . 235
10.3 Examples for Definitions and Theorems . . . 248

11 Library LIB breakable 253
11.1 Technical Overview . . . 253
11.2 Limitations and Known Bugs . . . 254
11.3 Main Option Keys . . . 255
11.4 Option Keys for the Break Appearance . . . 259
11.5 Breakable boxes and the `multicol` package . . . 261
11.6 Break Sequence for the Skins . . . 264
11.7 Break by Hand (Faked Break) . . . 272

12 Library LIB fitting 273
12.1 Macros of the Library . . . 273
12.2 Option Keys of the Library . . . 275

13 Library LIB hooks 283
13.1 Concept of Hooks . . . 283
13.2 Box Content Additions . . . 284
13.3 Embedding into the Surroundings . . . 285
13.4 Overlays . . . 286
13.5 Watermarks . . . 288
13.6 Underlays . . . 290
13.7 Finishes . . . 291
13.8 Skin Code . . . 291

14 Library LIB xparse 293
14.1 Option Keys . . . 293
14.2 Producing `tcolorbox` Environments and Commands . . . 295
14.3 Producing `tcbox` Commands . . . 298
14.4 Producing `tcblisting` Environments . . . 301
14.5 Producing `tcbinputlisting` Commands . . . 303
14.6 Producing `tboxfit` Commands . . . 304

15 Library LIB documentation 306
15.1 Macros of the Library . . . 306
15.2 Option Keys of the Library . . . 315
15.3 Predefined Colors of the Library . . . 320

A Picture Credits 321

References 322

Index 324

Contents

1 Introduction

The package originates from the first edition of my book «*LaTeX – Einführung in das Textsatzsystem*» [19] in about 2006. For the LaTeX examples and tutorials given there, I wanted to have accentuated and colored boxes to display source code and compiled text in combination. Since, in my opinion, this type of boxes is also quite useful to highlight definitions and theorems, I applied them for my lecture notes in mathematics [16–18] as well. With this package, you are invited to apply these boxes for similar projects.

Starting with version 2.00, for all internal calculations ε-TeX [2] expressions are used in replacement of the package `calc` [21]. The breaking news for version 2.00 is the support for breakable boxes. This new feature allows new applications of the package without affecting the core package too much if you do not need boxes to break automatically. With version 2.20, the often requested 'side by side' mode for listings has been added. With version 3.00, boxed titles are introduced together with improved customization options for overlays, underlays, finishes, and own code extensions.

Since the first public release in 2011, I received a lot of feedback from all over the world. I want to thank all who wrote me for supporting this package by sending bug reports and ideas for new or better features.

1.1 Installation

Typically, `tcolorbox` will be installed as part of a major LaTeX distribution and there is nothing special to do for a user.

If you intend to make a local installation *by hand*, see the `README` file of the `tcolorbox` package for some hints. The short story is: you have to install not only `tcolorbox.sty`, but also all `*.code.tex` files in the local `texmf` tree.

1.2 Loading the Package

The base package `tcolorbox` loads the packages `pgf` [20], `verbatim` [15], `etoolbox` [7], and `environ` [13]. `tcolorbox` itself is loaded in the usual manner in the preamble:

```
\usepackage{tcolorbox}
```

The package takes option keys in the key-value syntax. Alternatively, you may use these keys later in the preamble with `\tcbuselibrary` → P. 7 (see there). For example, the key to typeset listings is:

```
\usepackage[listings]{tcolorbox}
```

1.3 Libraries

The base package `tcolorbox` is extendable by program libraries. This is done by using option keys while loading the package or inside the preamble by applying the following macro with the same set of keys.

`\tcbuselibrary{`⟨*key list*⟩`}`

Loads the libraries given by the ⟨*key list*⟩.

```
\tcbuselibrary{listings,theorems}
```

The following keys are used inside `\tcbuselibrary` respectively `\usepackage` without the key tree path `/tcb/library/`.

`/tcb/library/skins` (LIB skins)

Loads the package `tikz` [20] and provides additional styles (skins) for the appearance of the colored boxes; see Section 7 from page 101.

`/tcb/library/listings` (LIB listings)

Loads the package `listings` [6] and provides additional macros for typesetting listings which are described in Section 9 from page 199.

`/tcb/library/listingsutf8` (LIB listingsutf8)

Loads the packages `listings` [6] and `listingsutf8` [10] for UTF-8 support. This is a variant of the library LIB listings and is described in Section 9 from page 199.

`/tcb/library/minted` (LIB minted)

Loads the package `minted` [14] to typeset listings with the `Pygments` [12] tool, also see Section 9 on page 199.

`/tcb/library/theorems` (LIB theorems)

Provides additional macros for typesetting theorems which are described in Section 10 from page 231.

`/tcb/library/breakable` (LIB breakable)

Provides support for automatic box breaking from one page to another; see Section 11 from page 253.

`/tcb/library/fitting` (LIB fitting)

Provides support for font size adaption of the box content to the box dimensions; see Section 12 from page 273.

`/tcb/library/hooks` (LIB hooks)

Extends several option keys to 'hookable' keys; see Section 13 from page 283.

`/tcb/library/xparse` (LIB xparse)

Provides document command production with `xparse` for `tcolorbox`; see Section 14 from page 293.

`/tcb/library/documentation` (LIB documentation)

Provides additional macros for typesetting LaTeX documentations which are described in Section 15 from page 306.

`/tcb/library/many` (style, no value)

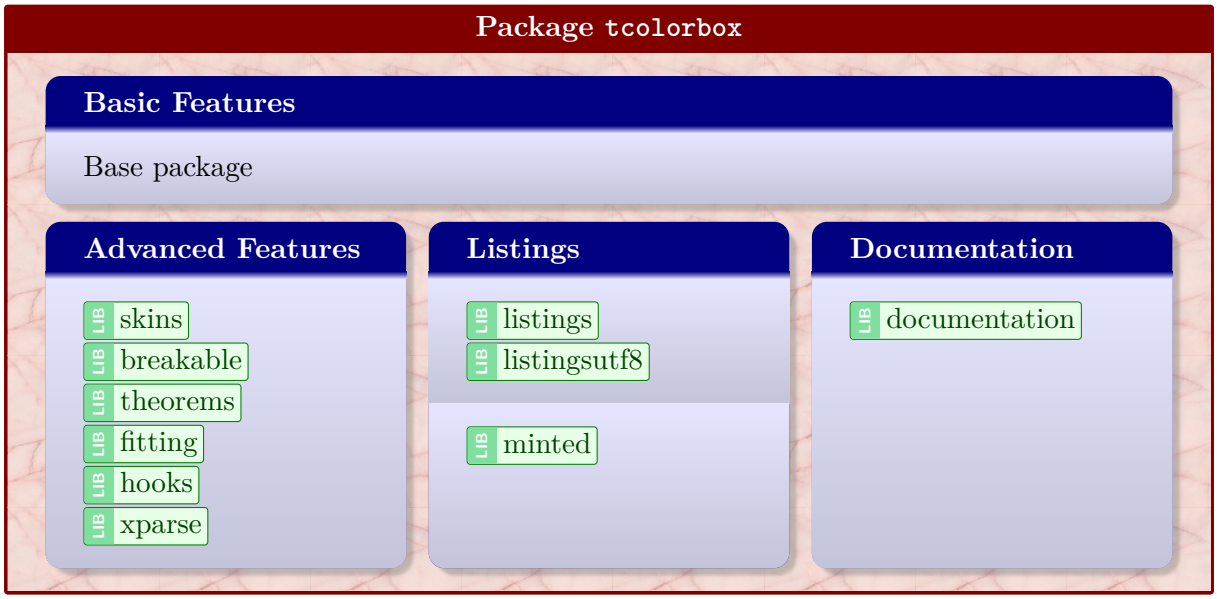
Loads the libraries LIB skins, LIB breakable, LIB hooks, LIB theorems, LIB fitting, and LIB xparse. Use this shortcut, if you want to use all features of `tcolorbox` with exception of typesetting listings and using the specialized LIB documentation library.

`/tcb/library/most` (style, no value)

Loads all libraries except LIB minted and LIB documentation. Use this shortcut, if you want to use all features of `tcolorbox` with exception of using the `minted` package and using the specialized LIB documentation library.

`/tcb/library/all` (style, no value)

Loads all libraries. Use this shortcut only, if you indend to use the LIB documentation library.



2 Macros for Box Creation

```
\begin{tcolorbox}[⟨options⟩]
  ⟨environment content⟩
\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 the source code

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

creates the following compiled text box:

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:

```
\begin{tcolorbox}
This is another \textbf{tcolorbox}.
\tcblower
Here, you see the lower part of the box.
\end{tcolorbox}
```

This code gives the following box:

This is another **tcolorbox**.

Here, you see the lower part of the box.

The ⟨*options*⟩ control the appearance and several functions of the boxes, see Section 3 on page 14 for the complete list. A quick example is given here:

```
\begin{tcolorbox}[colback=red!5!white,colframe=red!75!black,title=My nice heading]
This is another \textbf{tcolorbox}.
\tcblower
Here, you see the lower part of the box.
\end{tcolorbox}
```

My nice heading

This is another **tcolorbox**.

Here, you see the lower part of the box.

`\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 `\tcbline` → P. 153.

`\tcbset{`⟨*options*⟩`}`

Sets options for every following `tcolorbox` → P. 9 inside the current TEX group. By default, this does not apply to nested boxes, see Section 3.15 on page 73.

For example, the colors of the boxes may be defined for the whole document by this:

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

`\tcbsetforeverylayer{`⟨*options*⟩`}`

Sets options for every following `tcolorbox` → P. 9 inside the current TEX group. In contrast to `\tcbset`, this does also apply to nested boxes, see Section 3.15 on page 73. Technically, the ⟨*options*⟩ are appended to the default values for every tcolorbox which are applied by `/tcb/reset` → P. 81.

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 . . .).

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

\begin{tcolorbox}[title=All options for this box]
  This is a tcolorbox.\par\medskip
  \begin{tcolorbox}[title=Nested box]
    Note that this nested box has a red frame but no green background.
  \end{tcolorbox}
\end{tcolorbox}
\bigskip

\begin{tcolorbox}[reset]
  Options given with |\tcbsetforeverylayer| survive a |reset|.
\end{tcolorbox}
```

All options for this box

This is a tcolorbox.

Nested box

Note that this nested box has a red frame but no green background.

Options given with `\tcbsetforeverylayer` survive a `reset`.

`\tcbox[⟨options⟩]{⟨box content⟩}`

Creates a colored box which is fitted to the width of the given ⟨*box content*⟩. In principal, most ⟨*options*⟩ for a `tcolorbox` → P. 9 can be used for `\tcbox` with some restrictions. A `\tcbox` cannot have a lower part and cannot be broken.

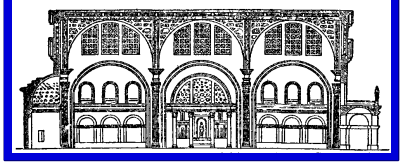
```
\tcbset{colframe=blue!50!black,colback=white,colupper=red!50!black,
        fonttitle=\bfseries,nobeforeafter,center title}

Text \tcbox[tcbox raise base]{Hello World}\hfill
%
\tcbox[left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=0mm,
  toptitle=0.5mm,bottomtitle=0.5mm,title=My table]{%
  \arrayrulecolor{blue!50!black}\renewcommand{\arraystretch}{1.2}%
  \begin{tabular}{r|c|l}
  One    & Two    & Three \\\hline\hline
  Men    & Mice   & Lions \\\hline
  Upper & Middle & Lower
  \end{tabular}}\hfill
%
\tcbox[colback=blue!85!black,
  left=0mm,right=0mm,top=0mm,bottom=0mm,boxsep=1mm,arc=0mm,boxrule=0.5pt,
  title=My picture]{%
  \includegraphics[width=5cm]{Basilica_5.png}}
```

Text Hello World

My table

One	Two	Three
Men	Mice	Lions
Upper	Middle	Lower

My picture



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

% Fixed width box
\begin{tcolorbox}Hello\\World!\end{tcolorbox}

% Fitted width box (like hbox or makebox)
\tcbox{Hello\\World!}

% Fitted width box (using a \tikzname\  node)
\tcbox[tikznode]{Hello\\World!}
```

Hello
World!

HelloWorld!

Hello
World!

See Section 14.2 on page 295 and Section 14.3 on page 298 for more elaborate methods to create new environments and commands.

`\newtcolorbox[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new environment ⟨*name*⟩ based on `tcolorbox`[→ P. 9]. Basically, `\newtcolorbox` operates like `\newenvironment`. 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`[→ P. 19] is set to the given ⟨*name*⟩ automatically. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

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

\begin{mybox}
This is my own box.
\end{mybox}
```

This is my own box.

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

\begin{mybox}{Hello there}
This is my own box with a mandatory title.
\end{mybox}
```

Hello there

This is my own box with a mandatory title.

```
\newtcolorbox{mybox}[2][]{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}[colback=yellow]{Hello there}
This is my own box with a mandatory title
and options.
\end{mybox}
```

Hello there

This is my own box with a mandatory title and options.

Definition in the preamble:

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

```
\begin{pabox}[colback=yellow]{Hello there}
This is my own box with a mandatory
numbered title and options.
\end{pabox}
```

Examp. 2.1: Hello there

This is my own box with a mandatory numbered title and options.

`\renewtcolorbox[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Operates like `\newtcolorbox`, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

`\newtcbox[⟨init options⟩]{\⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new macro `\⟨name⟩` based on `\tcbox` → P. 11. Basically, `\newtcbox` operates like `\newcommand`. The new macro `\⟨name⟩` optionally takes ⟨*number*⟩+1 arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcbox`. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

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

\mybox{This is my own box.}
```

This is my own box.

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

\mybox{Hello there}{This is my own box.}
```

Hello there

This is my own box.

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

\mybox[colback=yellow]{Hello there}%
  {This is my own box.}
```

Hello there

This is my own box.

Definition in the preamble:

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

```
\pbbox[colback=yellow]{Hello there}%
  {This is my own box.}
```

(2.2) Hello there

This is my own box.

```
\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}},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`, but based on `\renewcommand` instead of `\newcommand`. An existing macro is redefined.

3 Option Keys

For the ⟨*options*⟩ in `tcolorbox` → P. 9 respectively `\tcbset` → P. 10 the following `pgf` keys can be applied. The key tree path `/tcb/` is not to be used inside these macros. It is easy to add your own style keys using the syntax for `pgf` keys, see [19, 20] or the examples starting from page 218.

3.1 Title

`/tcb/title`=⟨*text*⟩ (no default, initially empty)

Creates a heading line with ⟨*text*⟩ as content.

```
\begin{tcolorbox}[title=My heading line]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My heading line

This is a **tcolorbox**.

`/tcb/notitle` (no value, initially set)

Removes the title line if set before.

`/tcb/adjusted title`=⟨*text*⟩ (style, no default, initially unset)

Creates a heading line with ⟨*text*⟩ as content. The minimal height of this line is adjusted to fit the text given by `/tcb/adjust text`. This option makes sense for single line headings if boxes are set side by side with equal height. Note that it is very easy to trick this adjustment.

```
\tcbset{colback=White,arc=0mm,width=(\linewidth-4pt)/4,
equal height group=AT,before=,after=\hfill,fonttitle=\bfseries}

The following titles are not adjusted:\\
\foreach \n in {xxx,ggg,AAA,\"Agypten}
{\begin{tcolorbox}[title=\n,colframe=red!75!black]
  Some content.\end{tcolorbox}}
Now, we try again with adjusted titles:\\
\foreach \n in {xxx,ggg,AAA,\"Agypten}
{\begin{tcolorbox}[adjusted title=\n,colframe=blue!75!black]
  Some content.\end{tcolorbox}}
```

The following titles are not adjusted:

xxx	ggg	AAA	Ägypten
Some content.	Some content.	Some content.	Some content.

Now, we try again with adjusted titles:

xxx	ggg	AAA	Ägypten
Some content.	Some content.	Some content.	Some content.

`/tcb/adjust text`=⟨*text*⟩ (no default, initially `Äpgjy`)

This sets the reference text for `/tcb/adjusted title`. If your texts never exceed 'Äpgjy' in depth and height you don't need to care about this option.

`/tcb/detach title` (no value)

Detaches the title from its normal position. The text of the title is stored into `\tcbtitletext` and the formatted title is available by `\tcbtitle`. The main application is to move the title from its usual place to another one.

```
\newtcolorbox{mybox}[2][]{colbacktitle=red!10!white,
  colback=blue!10!white,coltitle=red!70!black,
  title={#2},fonttitle=\bfseries,#1}

\begin{mybox}{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[detach title,before upper={\tcbtitle\quad}]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[detach title,after upper={\par\hfill\tcbtitle}]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
```

My title

This is a **tcolorbox**.

My title This is a **tcolorbox**.

This is a **tcolorbox**.

My title

`/tcb/attach title` (no value)

Attaches the title to its normal position. This option is used to reverse `/tcb/detach title`.

`/tcb/attach title to upper=`⟨*text*⟩ (style, default empty, initially unset)

Attaches the title to the begin of the upper part of the box content. The optional ⟨*text*⟩ is set between the formatted title and the box content.

```
\newtcolorbox{mybox}[2][]{colbacktitle=red!10!white,
  colback=blue!10!white,coltitle=red!70!black,
  title={#2},fonttitle=\bfseries,#1}

\begin{mybox}[attach title to upper={\ ---\ }]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
\begin{mybox}[attach title to upper,after title={:\ }]{My title}
  This is a \textbf{tcolorbox}.
\end{mybox}
```

My title — This is a **tcolorbox**.

My title: This is a **tcolorbox**.

More title options are documented in Section 3.9 on page 46 and Section 7.2 on page 107.

3.2 Subtitle

Inside the box content, one or more subtitles can be added. In general, a subtitle is a further tcolorbox → P. 9 which inherits some color and geometry options from the enclosing box. It may be customized just like any other tcolorbox → P. 9.

N 2014-10-10 **\tcbsubtitle**[⟨*options*⟩]{⟨*text*⟩}

Used inside a tcolorbox → P. 9 to add a subtitle box with the given ⟨*text*⟩. This is an independent tcolorbox → P. 9 which is formatted by several inherited properties of the enclosing box, by further settings from /tcb/subtitle style, and by the given ⟨*options*⟩.

```
\begin{tcolorbox}[title=My title,
    colback=red!5!white,
    colframe=red!75!black,
    fonttitle=\bfseries]
  This is a \textbf{tcolorbox}.
\tcbsubtitle[before skip=\baselineskip]%
  {My subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

```
\begin{tcolorbox}[title=My title,
    colback=red!5!white,
    colframe=red!75!black,
    colbacktitle=yellow!50!red,
    coltitle=red!25!black,
    fonttitle=\bfseries]
  This is a \textbf{tcolorbox}.
\tcbsubtitle[before skip=\baselineskip]%
  {My subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

N 2014-10-10 **/tcb/subtitle style**=⟨*options*⟩ (no default, initially empty)

Adds tcolorbox ⟨*options*⟩ to the settings for \tcbsubtitle.

```
\begin{tcolorbox}[title=My title,
  colback=red!5!white,
  colframe=red!75!black,
  colbacktitle=yellow!50!red,
  coltitle=red!25!black,
  fonttitle=\bfseries,
  subtitle style={boxrule=0.4pt,
    colback=yellow!50!red!25!white} ]
  This is a \textbf{tcolorbox}.
\tcbsubtitle{My subtitle}
  Further text.
\tcbsubtitle{Second subtitle}
  Further text.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My subtitle

Further text.

Second subtitle

Further text.

3.3 Lower Part

`/tcb/lowerbox`=⟨*mode*⟩ (no default, initially `visible`)

Controls the treatment of the lower part of the box. Feasible values for ⟨*mode*⟩ are:

- `visible`: usual type setting of the lower part,
- `invisible`: empty space instead of the lower part contents,
- `ignored`: the lower part is not used (here).

The last two values are usually applied in connection with `savelowerto`.

```
\begin{tcolorbox}[lowerbox=invisible,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part (but invisible).
\end{tcolorbox}

\begin{tcolorbox}[lowerbox=ignored,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part (but ignored).
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/savelowerto`=⟨*file name*⟩ (no default, initially empty)

Saves the content of the lower part in a file for an optional later usage.

```
\begin{tcolorbox}[lowerbox=invisible,savelowerto=\jobname_bspsave.tex,colback=white]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part which may be quite complex:
$\displaystyle f(x)=\frac{1+x^2}{1-x^2}$.
\end{tcolorbox}

Now, we load the saved text:\\
\input{\jobname_bspsave.tex}
```

This is a **tcolorbox**.

Now, we load the saved text:

This is the lower part which may be quite complex: $f(x) = \dfrac{1+x^2}{1-x^2}$.

`/tcb/lower separated=true|false` (default `true`, initially `true`)

If set to `true`, the lower part is visually separated from the upper part. It depends on the chosen skin how the visualization of the separation is done.

```
% \tcbuselibrary{skins}
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,nobeforeafter,
  width=(\linewidth-4mm)/2,fontlower=\itshape}

\begin{tcolorbox}[title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}\hfill
\begin{tcolorbox}[title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
\par\bigskip\par
\begin{tcolorbox}[sidebyside,title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}\hfill
\begin{tcolorbox}[sidebyside,title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
\par\bigskip\par
\begin{tcolorbox}[beamer,title=Lower separated]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}\hfill
\begin{tcolorbox}[beamer,title=Lower not separated,lower separated=false]
This is the upper part.
\tcblower
This is the lower part.
\end{tcolorbox}
```

Lower separated

This is the upper part.

This is the lower part.

Lower not separated

This is the upper part.

This is the lower part.

Lower separated

This is the upper part. | *This is the lower part.*

Lower not separated

This is the upper part. | *This is the lower part.*

Lower separated

This is the upper part.

This is the lower part.

Lower not separated

This is the upper part.

This is the lower part.

`/tcb/savedelimiter=`⟨*name*⟩ (no default, initially `tcolorbox`)

Used in connection with new environment definitions which extend `tcolorbox` and use or allow the option `savelowerto`. To catch the end of the new box environment ⟨*name*⟩ has to be the name of this environment. Additionally, the environment definition has to use `\tcolorbox` instead of `\begin{tcolorbox}` and `\endtcolorbox` instead of `\end{tcolorbox}`.

```
\newenvironment{mybox}[1]{%
  \tcolorbox[savedelimiter=mybox,
             savelowerto=\jobname_bspsave2.tex,lowerbox=ignored,
             colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
             title=#1]}%
  {\endtcolorbox}

\begin{mybox}{My Example}
Upper part.
\tcblower
Saved lower part!
\end{mybox}

Now, the saved part is used:
\begin{tcolorbox}[colback=green!5]
\input{\jobname_bspsave2.tex}
\end{tcolorbox}
```

My Example

Upper part.

Now, the saved part is used:

Saved lower part!

The `savedelimiter` is used implicitely with `\newtcolorbox` → P. 12 which allows a more convenient usage:

```
\newtcolorbox{mybox}[1]{%
             savelowerto=\jobname_bspsave2.tex,lowerbox=ignored,
             colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
             title=#1}%

\begin{mybox}{My Example}
Upper part.
\tcblower
Saved lower part!
\end{mybox}

Now, the saved part is used:
\begin{tcolorbox}[colback=green!5]
\input{\jobname_bspsave2.tex}
\end{tcolorbox}
```

My Example

Upper part.

Now, the saved part is used:

Saved lower part!

3.4 Colors and Fonts

`/tcb/colframe=`⟨*color*⟩ (no default, initially `black!75!white`)

Sets the frame ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[colframe=red!50!white]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/colback=`⟨*color*⟩ (no default, initially `black!5!white`)

Sets the background ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[colback=red!50!white]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/title filled=true|false` (default `true`, initially `false`)

Switches the drawing of the title background according to the given value. This option is set to `true` automatically by `/tcb/colbacktitle`, `/tcb/opacitybacktitle`→ P. 37, and `/tcb/title style`→ P. 104, and `/tcb/title code`→ P. 95.

```
\begin{tcolorbox}[title=My title,title filled]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[title=My title,
  title filled=false]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

My title

This is a **tcolorbox**.

`/tcb/colbacktitle=`⟨*color*⟩ (no default, initially `black!50!white`)

Sets the background ⟨*color*⟩ of the title area of the box.

```
\begin{tcolorbox}[colbacktitle=red!50!white,
  title=My title,coltitle=black,
  fonttitle=\bfseries]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/colupper=⟨*color*⟩ (no default, initially `black`)

Sets the text ⟨*color*⟩ of the upper part.

```
\begin{tcolorbox}[colupper=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/collower=⟨*color*⟩ (no default, initially `black`)

Sets the text ⟨*color*⟩ of the lower part.

```
\begin{tcolorbox}[collower=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/coltext=⟨*color*⟩ (style, no default, initially `black`)

Sets the text ⟨*color*⟩ of the box. This is an abbreviation for setting `colupper` and `collower` to the same value.

```
\begin{tcolorbox}[coltext=red!75!black]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/coltitle=⟨*color*⟩ (no default, initially `white`)

Sets the title text ⟨*color*⟩ of the box.

```
\begin{tcolorbox}[coltitle=red!75!black,
  colbacktitle=black!10!white,title=Test]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Test

This is a **tcolorbox**.

/tcb/fontupper=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the upper part (e. g. font settings).

```
\begin{tcolorbox}[fontupper=Hello!~\sffamily]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Hello! This is a **tcolorbox**.

/tcb/fontlower=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the lower part (e. g. font settings).

```
\begin{tcolorbox}[fontlower=\sffamily\bfseries]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/fonttitle=⟨*text*⟩ (no default, initially empty)

Sets ⟨*text*⟩ before the content of the title text (e. g. font settings).

```
\begin{tcolorbox}[fonttitle=\sffamily\bfseries\large,title=Hello]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Hello

This is a **tcolorbox**.

More color options are provided by using skins documented in Section 7 from page 101.

3.5 Geometry

/tcb/width=⟨*length*⟩ (no default, initially `\linewidth`)

Sets the total width of the colored box to ⟨*length*⟩. See also /tcb/height → P. 39.

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

\begin{tcolorbox}[width=\linewidth/2]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/toprule=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the top rule to ⟨*length*⟩.

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

\begin{tcolorbox}[toprule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/bottomrule=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the bottom rule to ⟨*length*⟩.

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

\begin{tcolorbox}[bottomrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

/tcb/leftrule=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the left rule to ⟨*length*⟩.

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

\begin{tcolorbox}[leftrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/rightrule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the right rule to ⟨*length*⟩.

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

\begin{tcolorbox}[rightrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/titlerule`=⟨*length*⟩ (no default, initially `0.5mm`)

Sets the line width of the rule below the title to ⟨*length*⟩.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,
  colbacktitle=red!90!black}

\begin{tcolorbox}[titlerule=3mm,title=This is the title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is the title

This is a **tcolorbox**.

`/tcb/boxrule`=⟨*length*⟩ (style, no default, initially `0.5mm`)

Sets all rules of the frame to ⟨*length*⟩, i. e. `/tcb/toprule` → P. 23, `/tcb/bottomrule` → P. 23, `/tcb/leftrule` → P. 23, `/tcb/rightrule`, and `/tcb/titlerule`.

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

\begin{tcolorbox}[boxrule=3mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

More options for drawing a `/tcb/borderline` → P. 126 are provided by using skins documented in Section 7 from page 101.

`/tcb/arc`=⟨*length*⟩ (no default, initially `1mm`)

Sets the inner radius of the four frame arcs to ⟨*length*⟩.

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

\begin{tcolorbox}[arc=0mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[arc=4mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/outer arc`=⟨*length*⟩ (no default, initially unset)

Sets the outer radius of the four frame arcs to ⟨*length*⟩.

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

\begin{tcolorbox}[arc=4mm,outer arc=1mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/auto outer arc` (no value, initially set)

Sets the outer radius of the four frame arcs automatically in dependency of the inner radius given by `/tcb/arc`.

`/tcb/boxsep`=⟨*length*⟩ (no default, initially `1mm`)

Sets a common padding of ⟨*length*⟩ between the text content and the frame of the box. This value is added to the key values of `left`, `right`, `top`, `bottom`, and `middle` at the appropriate places.

```
\tcbset{colback=red!5!white,colframe=red!75!black,width=(\linewidth-4mm)/2,
  before=,after=\hfill}

\begin{tcolorbox}[boxsep=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[boxsep=5mm,draft]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

frame: w=195.33255pt, h=48.97505pt

upper: w=141.2724pt, h=6.296pt

interior: w=192.4873pt, h=46.1298pt

`/tcb/left`=⟨*length*⟩ (style, no default, initially `4mm`)

Sets the left space between all text parts and frame (additional to `boxsep`). This is an abbreviation for setting `lefttitle`, `leftupper`, and `leftlower` to the same value.

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

\begin{tcolorbox}[left=0mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/lefttitle`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between title text and frame (additional to `boxsep`).

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

\begin{tcolorbox}[lefttitle=3cm,title=My Title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My Title

This is a **tcolorbox**.

`/tcb/leftupper`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between upper text and frame (additional to `boxsep`).

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

\begin{tcolorbox}[leftupper=3cm,title=My Title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My Title

This is a **tcolorbox**.

`/tcb/leftlower`=⟨*length*⟩ (no default, initially `4mm`)

Sets the left space between lower text and frame (additional to `boxsep`).

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

\begin{tcolorbox}[leftlower=3cm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/right`=⟨*length*⟩ (style, no default, initially `4mm`)

Sets the right space between all text parts and frame (additional to `boxsep`). This is an abbreviation for setting `righttitle`, `rightupper`, and `rightlower` to the same value.

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

\begin{tcolorbox}[width=5cm,right=2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/righttitle`=⟨*length*⟩ (no default, initially `4mm`)

Sets the right space between title text and frame (additional to `boxsep`).

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

\begin{tcolorbox}[width=5cm,righttitle=2cm,title=My very long title text]
This is a \textbf{tcolorbox} with standard upper box dimensions.
\end{tcolorbox}
```

My very long title text

This is a **tcolorbox** with standard upper box dimensions.

`/tcb/rightupper`=⟨*length*⟩ (no default, initially `4mm`)

Sets the right space between upper text and frame (additional to `boxsep`).

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

\begin{tcolorbox}[width=5cm,rightupper=2cm,title=My very long title text]
This is a \textbf{tcolorbox} with compressed upper box dimensions.
\end{tcolorbox}
```

My very long title text

This is a **tcolorbox** with compressed upper box dimensions.

/tcb/rightlower=⟨*length*⟩ (no default, initially 4mm)

Sets the right space between lower text and frame (additional to boxsep).

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

\begin{tcolorbox}[width=5cm,rightlower=2cm]
This is a \textbf{tcolorbox} with standard upper box dimensions.
\tcblower
This is the lower part with large space at right.
\end{tcolorbox}
```

This is a **tcolorbox** with standard upper box dimensions.

This is the lower part with large space at right.

/tcb/top=⟨*length*⟩ (no default, initially 2mm)

Sets the top space between text and frame (additional to boxsep).

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

\begin{tcolorbox}[top=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

/tcb/toptitle=⟨*length*⟩ (no default, initially 0mm)

Sets the top space between title and frame (additional to boxsep).

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

\begin{tcolorbox}[toptitle=3mm,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

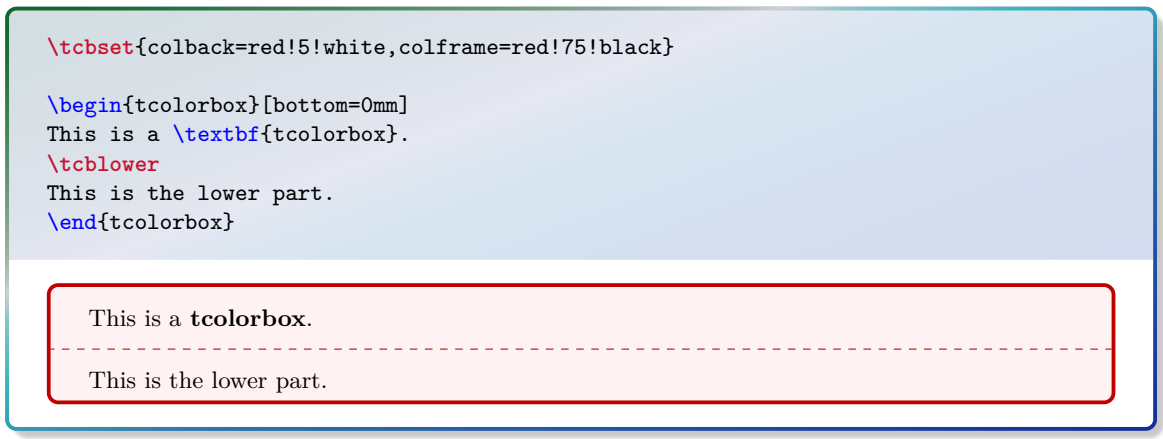
My title

This is a **tcolorbox**.

/tcb/bottom=⟨*length*⟩ (no default, initially 2mm)

Sets the bottom space between text and frame (additional to boxsep).

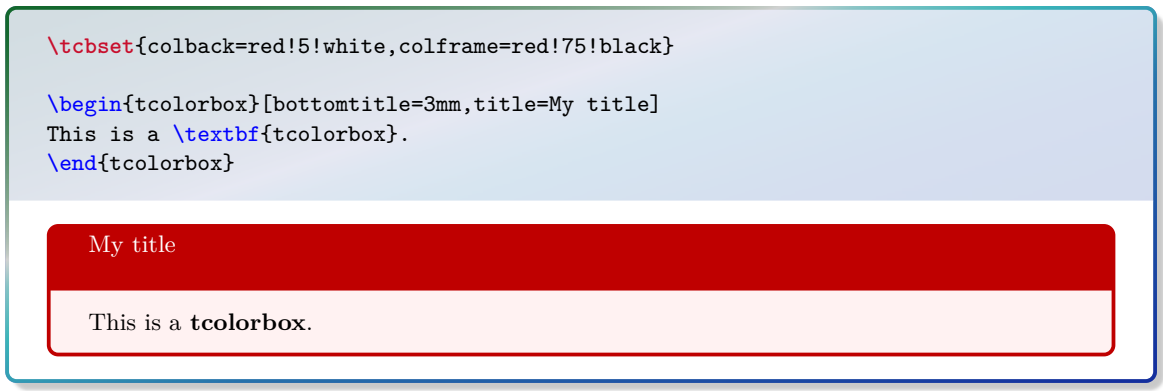
```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[bottom=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tcb/bottomtitle=⟨*length*⟩ (no default, initially 0mm)

Sets the bottom space between title and frame (additional to boxsep).

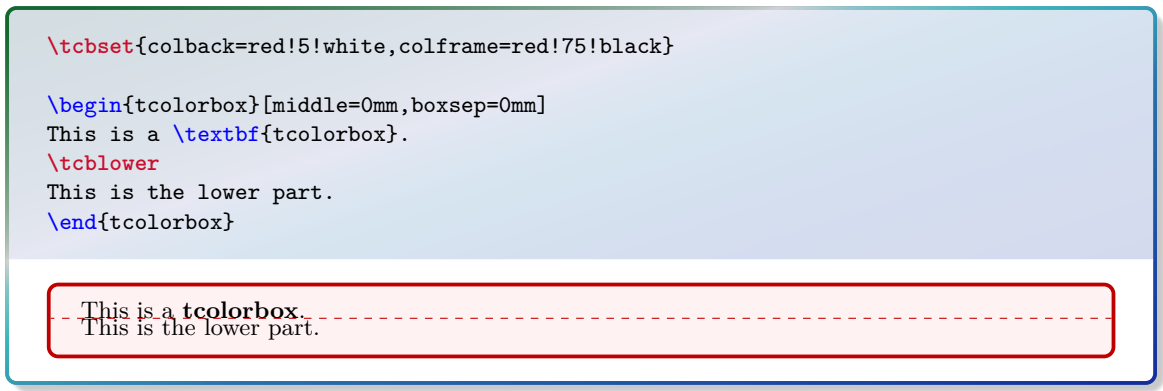
```
\tcbset{colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}[bottomtitle=3mm,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/middle=⟨*length*⟩ (no default, initially 2mm)

Sets the space between upper and lower text to the separation line (additional to boxsep).

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

\begin{tcolorbox}[middle=0mm,boxsep=0mm]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

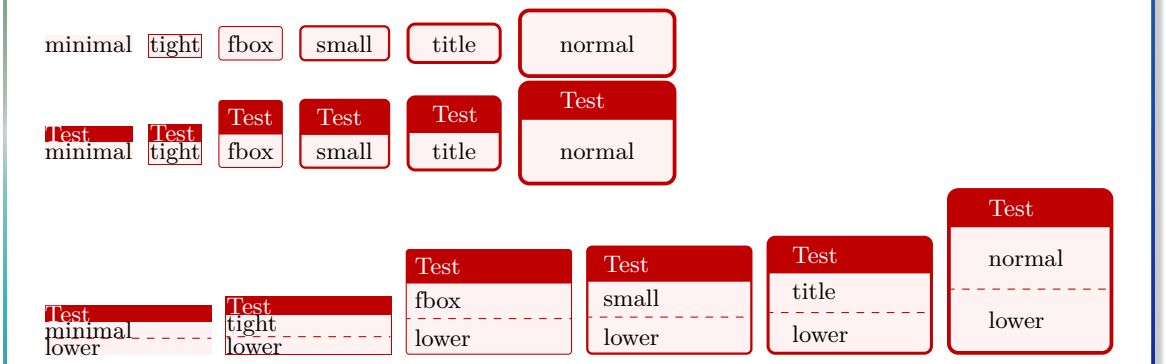


`/tcb/size=`⟨*name*⟩ (no default, initially `normal`)

Sets all geometry keys with exception of `/tcb/width` → P. 23 to predefined length values. For ⟨*name*⟩, the following values are feasible:

- `normal`: normal sized boxes e.g. of width `\linewidth`.
- `title`: title line sized boxes.
- `small`: small boxes e.g. for keyword highlighting.
- `fbox`: identical to the standard `\fbox`.
- `tight`: no padding space at all.
- `minimal`: no padding space, no box rules.

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

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \tcbox[size=\s,on line]{\s} }

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \tcbox[size=\s,on line,title=Test]{\s} }

\foreach \s in {minimal,tight,fbox,small,title,normal} {
  \begin{tcolorbox}[size=\s,on line,title=Test,width=2.2cm]
    \s \tcblower lower\end{tcolorbox} }
```



Predefined values

	normal	title	small	fbox	tight	minimal
boxrule	0.5mm	0.4mm	0.3mm	0.4pt	0.4pt	0.0pt
boxsep	1.0mm	1.0mm	1.0mm	3.0pt	0.0pt	0.0pt
left	4.0mm	2.0mm	1.0mm	0.0pt	0.0pt	0.0pt
right	4.0mm	2.0mm	1.0mm	0.0pt	0.0pt	0.0pt
top	2.0mm	0.25mm	0.0mm	0.0pt	0.0pt	0.0pt
bottom	2.0mm	0.25mm	0.0mm	0.0pt	0.0pt	0.0pt
toptitle	0.0mm	0.0mm	0.0mm	0.0pt	0.0pt	0.0pt
bottomtitle	0.0mm	0.0mm	0.0mm	0.0pt	0.0pt	0.0pt
middle	2.0mm	0.75mm	0.5mm	1.0pt	0.2pt	0.0pt
arc	1.0mm	0.75mm	0.5mm	1.0pt	0.0pt	0.0pt
outer arc	auto	auto	auto	auto	0.0pt	0.0pt

`/tcb/oversize=`⟨*length*⟩ (style, default `0pt`)

Sets the text width of the upper part to the current line width plus an optional ⟨*length*⟩. This is achieved by changing the keys `/tcb/width` → P. 23 `/tcb/enlarge left by` → P. 68, and `/tcb/enlarge right by` → P. 68 appropriately. The resulting box is overlapping into the left and right margin of the page. Note that this style option has to be given *after* all other geometry keys!

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

\textit{Normal text for comparison:}\\
\lipsum[2]

\begin{tcolorbox}[oversize,title=Oversized box]
\lipsum[2]
\end{tcolorbox}

\begin{tcolorbox}[title=Normal box]
\lipsum[2]
\end{tcolorbox}
```

Normal text for comparison:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Oversized box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Normal box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/toggle left and right=`⟨*toggle preset*⟩ (default `evenpage`, initially `none`)

According to the ⟨*toggle preset*⟩, the left and the right settings of the `tcolorbox` are switched or not. Feasible values are:

- `none`: no switching.
- `forced`: the values of the left and right rules and spaces are switched.
- `evenpage`: if the page is an even page, the values of the left and right rules and spaces are switched. It is recommended that one use this setting in conjunction with `/tcb/check odd page` → P. 80.

```
% \usepackage{changepage} for 'check odd page'
% \usepackage{lipsum}
% \usetikzlibrary{patterns}
% \tcbuselibrary{skins,breakable}
\begin{tcolorbox}[skin=enhancedmiddle,breakable,
  check odd page,toggle left and right,
  boxrule=0mm,top=0mm,bottom=0mm,left=1mm,right=1mm,
  rightrule=1cm,colupper=blue!25!black,
  interior style={fill overzoom image=lichtspiel.jpg,fill image opacity=0.25},
  frame style={pattern=crosshatch dots light steel blue},
  overlay={%
    \ifoddpage\coordinate (X) at ([xshift=-5mm]frame.east);
         \else\coordinate (X) at ([xshift=5mm]frame.west);\fi
    \fill[shading=ball,ball color=blue!50!white,opacity=0.5] (X) circle (4mm);}]
\lipsum[1-6]
\end{tcolorbox}
```

This example switches a `1cm` thick rule from the left to the right side depending on the page number. Thereby, the rule is always on the outer side of the double-sided paper. Additionally, a ball is drawn on the outer side with help of an overlay.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc

eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

3.6 Corners

The four corners of any `tcolorbox` can be set individually as `/tcb/sharp corners` or as `/tcb/rounded corners` → P. 35. These settings are also reflected in the behavior of `/tcb/borderline` → P. 126 and `/tcb/shadow` → P. 129 as one would expect.

By default, all four corners are *rounded*. So, only the `/tcb/sharp corners` option will be necessary for most use cases. The `/tcb/rounded corners` → P. 35 option can be used to revert a `/tcb/sharp corners` setting.

`/tcb/sharp corners`=⟨*position*⟩ (default `all`, initially unset)

The ⟨*position*⟩ denotes one or more of the four box corners to be set as *sharp* corners. The not assigned corners will retain their mode. Feasible values for ⟨*position*⟩ are:

- `northwest`
- `northeast`
- `southwest`
- `southeast`
- `north`
- `south`
- `east`
- `west`
- `downhill`
- `uphill`
- `all`

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharp corners=northwest ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharp corners ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/rounded corners=`⟨*position*⟩ (default `all`, initially `all`)

The `/tcb/rounded corners` can be used to revert a `/tcb/sharp corners` → P. 34 setting. The ⟨*position*⟩ denotes one or more of the four box corners to be set as *rounded* corners. The not assigned corners will retain their mode. Feasible values for ⟨*position*⟩ are[2]:

- `northwest`
- `northeast`
- `southwest`
- `southeast`
- `north`
- `south`
- `east`
- `west`
- `downhill`
- `uphill`
- `all`

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,sharp corners,
  rounded corners=northwest ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/sharpish corners` (style, no value)

Shortcut for setting `/tcb/arc` → P. 25 and `/tcb/outer arc` → P. 25 to `0pt`. With this setting, rounded corners will appear als quasi-sharp, but e. g. the shadow will be somewhat rounder than the shadow of really sharp corners.

Corners are still of type *rounded* with this option, but appear *sharp*. To switch back to rounded corners, one has to adapt `/tcb/arc` → P. 25 and `/tcb/outer arc` → P. 25.

```
\begin{tcolorbox}[colback=red!5!white,
  colframe=red!75!black,
  sharpish corners ]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

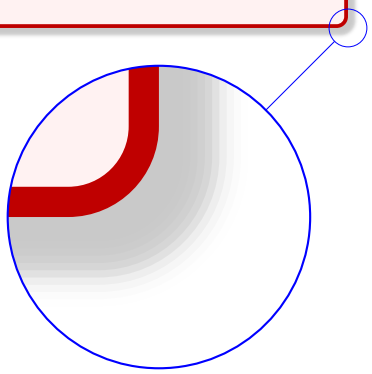
[2] The graphical examples assume that the boxes where set to have sharp corners before.

The following examples will show the differences between /tcb/rounded corners→ P. 35, /tcb/sharpish corners→ P. 35, and /tcb/sharp corners→ P. 34. The later two give the same core box, but /tcb/borderline→ P. 126 and /tcb/shadow→ P. 129 settings are slightly different. The following examples use /tcb/drop fuzzy shadow→ P. 131.

My title

This is a **tcolorbox**.

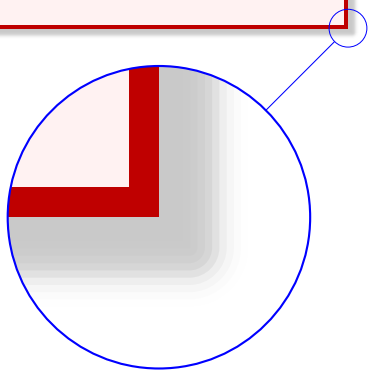
rounded corners



My title

This is a **tcolorbox**.

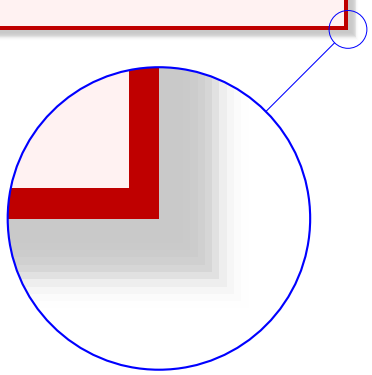
sharpish corners



My title

This is a **tcolorbox**.

sharp corners



3.7 Transparency

Transparency effects are likely to be used in conjunction with *jigsaw* skin variants, see Section 7.10 on page 145.

`/tcb/opacityframe`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the frame opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[opacityframe=0.25]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/opacityback`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the background opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,
  opacityframe=0.5, opacityback=0.5]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

`/tcb/opacitybacktitle`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the title background opacity of the box to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,
  opacityframe=0.5, opacitybacktitle=0.5,
  title filled, title=This is a title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

`/tcb/opacityfill`=⟨*fraction*⟩ (style, no default, initially `1.0`)

Sets the fill opacity for frame, interior and optionally the title background to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[standard jigsaw,
  opacityfill=0.7, title=This is a title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a title

This is a **tcolorbox**.

/tcb/opacityupper=⟨*fraction*⟩ (no default, initially 1.0)

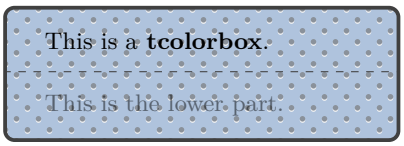
Sets the text opacity of the upper box part to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacityupper=0.5,
  interior style={pattern=crosshatch dots ⤸
      ⤹light steel blue}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

/tcb/opacitylower=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the lower box part to the given ⟨*fraction*⟩.

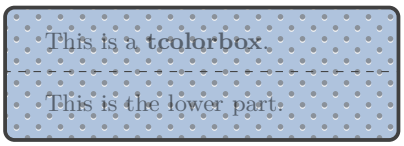
```
\begin{tcolorbox}[enhanced,opacitylower=0.5,
  interior style={pattern=crosshatch dots ⤸
      ⤹light steel blue}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tcb/opacitytext=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the upper and the lower box part to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacitytext=0.5,
  interior style={pattern=crosshatch dots ⤸
      ⤹light steel blue}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

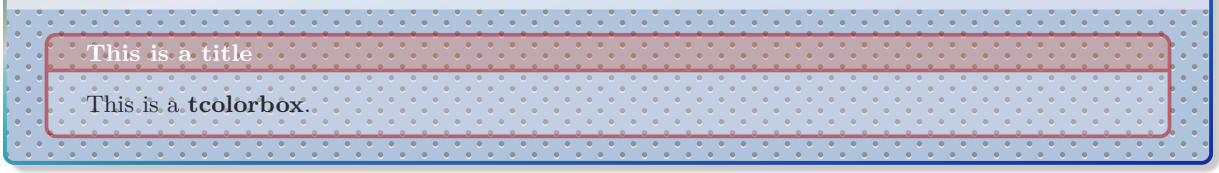


/tcb/opacitytitle=⟨*fraction*⟩ (no default, initially 1.0)

Sets the text opacity of the box title to the given ⟨*fraction*⟩.

```
\begin{tcolorbox}[enhanced,opacitytitle=0.7,
  fonttitle=\bfseries,title=This is a title,
  title style={pattern=crosshatch dots light ⤸
      ⤹steel blue}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

```
\begin{tcolorbox}[enhanced jigsaw,fonttitle=\bfseries,title=This is a title,
  opacityframe=0.5,opacityback=0.25,opacitybacktitle=0.25,opacitytext=0.8,
  colback=red!5!white,colframe=red!75!black,colbacktitle=yellow!20!red]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



3.8 Height Control

In a typical usage scenario, the height of a `tcolorbox` is computed automatically to fit the content. Nevertheless, the height can be set to a fixed value or to fit commonly for several boxes, e. g. if boxes are set side by side.

The height control keys are only applicable to unbreakable boxes. If a box is set to be `/tcb/breakable` → P. 255, the height is always computed according to the *natural height*.

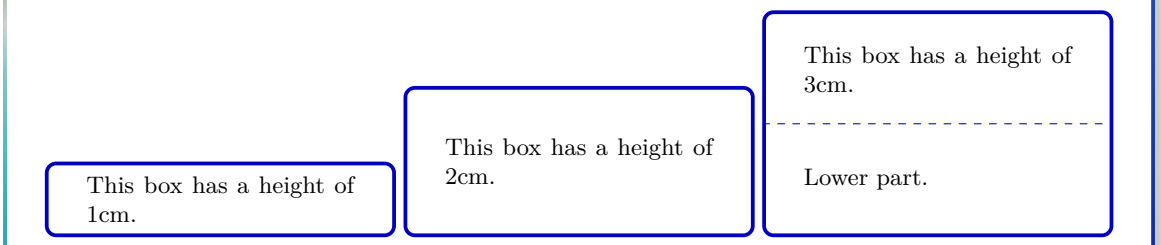
`/tcb/natural height` (no value, initially set)

Sets the total height of the colored box to its natural height depending on the box content.

`/tcb/height`=⟨*length*⟩ (no default)

Sets the total height of the colored box to ⟨*length*⟩ independent of the box content. ⟨*length*⟩ is the minimum height of the box, if `/tcb/height plus` is larger than zero.

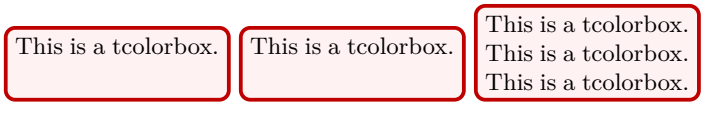
```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white}

\begin{tcolorbox}[height=1cm,valign=center]
  This box has a height of 1cm.
\end{tcolorbox}
\begin{tcolorbox}[height=2cm,valign=center]
  This box has a height of 2cm.
\end{tcolorbox}
\begin{tcolorbox}[height=3cm,split=0.5,valign=center,valign lower=center]
  This box has a height of 3cm.
  \tcblower
  Lower part.
\end{tcolorbox}
```



`/tcb/height plus`=⟨*length*⟩ (no default, initially `0pt`)

The box may extend a given fixed `/tcb/height` up to the given ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,nobeforeafter}

\begin{tcolorbox}[height=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[height=1cm,height plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[height=1cm,height plus=1cm]
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/height from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box height to a dimension between ⟨*min*⟩ and ⟨*max*⟩.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=4.5cm,nobeforeafter,
  height from=2cm to 8cm}

\begin{mybox}
This is a tcolorbox.
\end{mybox}
\begin{mybox}
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{mybox}
\begin{mybox}
\lipsum[2]
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

N 2014-09-22 `/tcb/height fill=true|false` (default `true`, initially `false`)

If set to `true`, the height of the `tcolorbox` is set to the rest of the available vertical space of the current page. Note that the `tcolorbox` is always set as its own paragraph using this option. This option overrules `/tcb/natural height` → P. 39, `/tcb/height` → P. 39, etc.

Note that the library LIB breakable has to be loaded to use this key!

This height control key is only applicable to unbreakable boxes but it uses code from the library LIB breakable. The counterpart for breakable boxes is `/tcb/height fixed for` → P. 260.

```
% \usepackage{lipsum}
% \tcbuselibrary{breakable}
\begin{tcolorbox}[height fill,
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Box which fills the rest of the page]
\lipsum[1]
\end{tcolorbox}
```

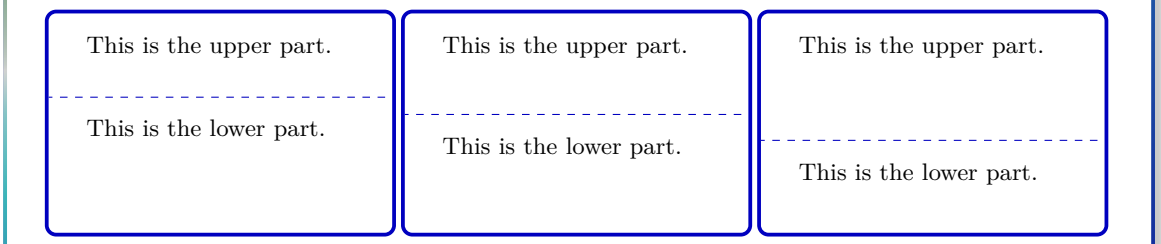
Box which fills the rest of the page

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/space=`⟨*fraction*⟩ (no default, initially 0)

If the height of a `tcolorbox` is not the natural height, the space difference between the forced and the natural size is distributed between the upper and the lower part of the box. This space could also be negative. ⟨*fraction*⟩ with a value between 0 and 1 is the amount of space which is added to the upper part, the rest is added to the lower part. If there is no lower part, then all of the space is added to the upper part always.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=3cm}

\foreach \f in {0.2,0.4,0.7}
{\begin{tcolorbox}[space=\f]
    This is the upper part.
    \tcblower
    This is the lower part.
\end{tcolorbox}}
```



`/tcb/space to upper` (style)

This is an abbreviation for `space=1`, i. e. all extra space is added to the upper part.

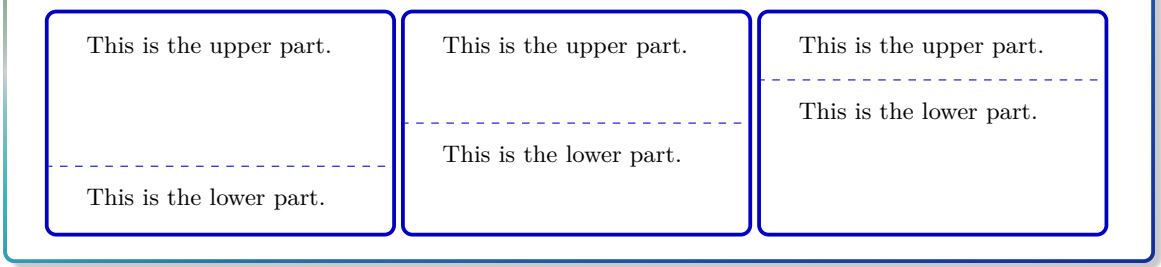
`/tcb/space to lower` (style, initially set)

This is an abbreviation for `space=0`, i. e. all extra space is added to the lower part (if there is any).

`/tcb/space to both` (style)

This is an abbreviation for `space=0.5`, i. e. the extra space equally distributed between the upper and the lower part.

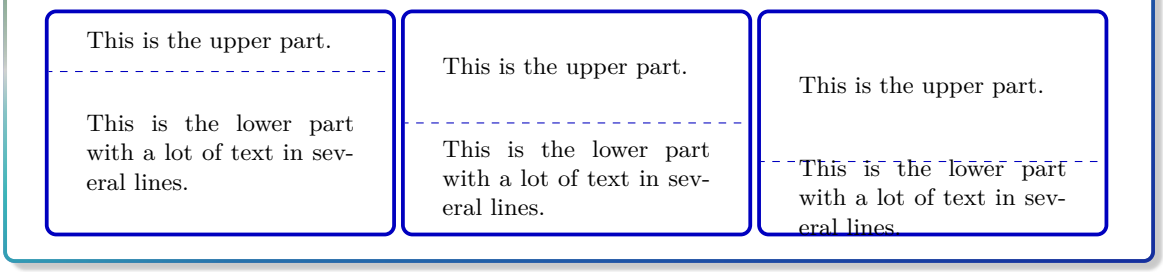
```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=3cm}

\foreach \myspace in {space to upper,space to both,space to lower}
{\begin{tcolorbox}[\myspace]
    This is the upper part.
    \tcblower
    This is the lower part.
\end{tcolorbox}}
```



`/tcb/split`=⟨*fraction*⟩ (no default)

If the height of a `tcolorbox` is not the natural height, the ⟨*fraction*⟩ with a value between 0 and 1 determines the positioning of the segmentation between the upper and the lower part. Here, 0 stands for top and 1 for bottom. Note that the box is split regardless of the actual dimensions of the text parts!

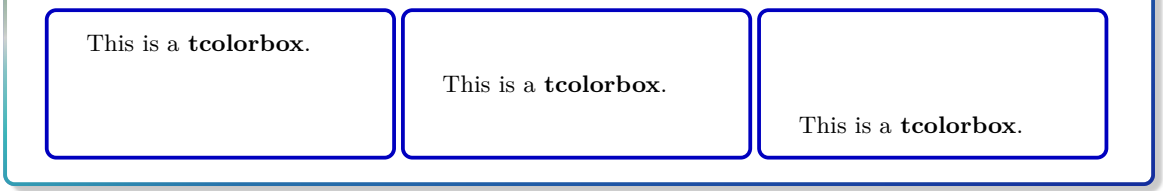
```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,height=3cm,
colback=white,colframe=blue!75!black,valign=center,valign lower=center}

\foreach \f in {0.1,0.5,0.8}
{\begin{tcolorbox}[split=\f]
This is the upper part.
\tcblower
This is the lower part with a lot of text in several lines.
\end{tcolorbox}}
```



`/tcb/valign`=⟨*alignment*⟩ (no default, initially `top`)

If the height of a `tcolorbox` is not the natural height, `valign` determines the vertical ⟨*alignment*⟩ of the upper part. Feasible values are `top`, `center`, and `bottom`. For a box with natural height, these values are meaningless.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,
colframe=blue!75!black,colback=white,height=2cm}

\foreach \myalign in {top,center,bottom}
{\begin{tcolorbox}[valign=\myalign]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}}
```



`/tcb/valign lower`=⟨*alignment*⟩ (no default, initially `top`)

This key has the same meaning for the lower part as `valign` for the upper part, i. e. it determines the vertical ⟨*alignment*⟩ of the lower part with feasible values `top`, `center`, and `bottom`.

`/tcb/equal height group=`⟨*id*⟩ (no default)

Boxes which are members of an `equal height group` will all get the same height, i. e. the maximum of all their natural heights. The ⟨*id*⟩ serves to distinguish between different height groups. This ⟨*id*⟩ should contain only characters which are feasible for TEX macro names, typically alphabetic characters but no numerals and spaces. Note that you have to compile twice to see changes and that height groups are global definitions.

```
\tcbset{width=(\linewidth-2mm)/3,before=,after=\hfill,arc=0mm,
colframe=blue!75!black,colback=white,fonttitle=\bfseries}

\begin{tcolorbox}[equal height group=A,adjusted title={One}]
  My smallest box.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=A,adjusted title={Two}]
  This box is also small.
  \tcblower
  But with a lower part.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=A,adjusted title={Three}]
  This box contains a lot of text just to fill the space
  with word flowing and flowing and flowing until the box
  is filled with all of it.
\end{tcolorbox}\linebreak
%
\tcbset{width=(\linewidth-1mm)/2,before=,after=\hfill,arc=0mm,
colframe=red!75!black,colback=white}
%
\begin{tcolorbox}[equal height group=B]
  Now, we use another equal height group.
\end{tcolorbox}%
\begin{tcolorbox}[equal height group=B]
  \begin{equation*}
    \int\limits_{0}^{1} x^2 = \frac13.
  \end{equation*}
\end{tcolorbox}
```

One

My smallest box.

Two

This box is also small.

But with a lower part.

Three

This box contains a lot of text just to fill the space with word flowing and flowing and flowing until the box is filled with all of it.

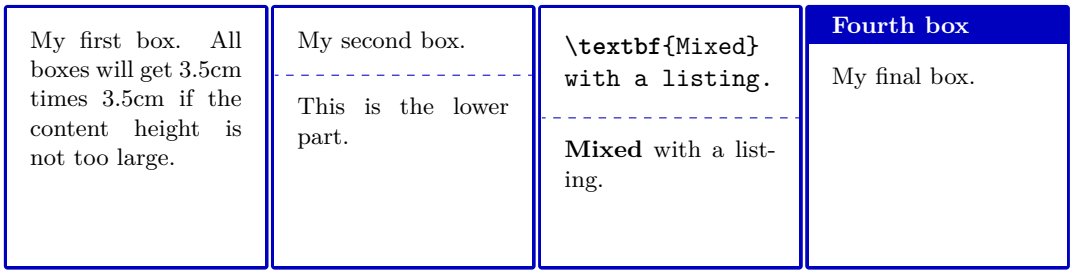
Now, we use another equal height group.

$$\int_0^1 x^2 = \frac{1}{3}.$$

`/tcb/minimum for equal height group=`⟨*id*⟩`:`⟨*length*⟩ (no default, initially unset)

Plants a ⟨*length*⟩ into the equal height group with the given ⟨*id*⟩. This ensures that the height will not drop below ⟨*length*⟩. Note that you cannot reduce a computed height value by using this key with a small value. The difference to applying `/tcb/height` → P. 39 directly is that the boxes are never too small for their content.

```
\tcbset{colframe=blue!75!black,colback=white,arc=0mm,
  before=,after=\hfill,fonttitle=\bfseries,left=2mm,right=2mm,
  width=3.5cm,
  equal height group=C,
  minimum for equal height group=C:3.5cm}

\begin{tcolorbox}
  My first box. All boxes will get 3.5cm times 3.5cm
  if the content height is not too large.
\end{tcolorbox}%
\begin{tcolorbox}
  My second box.
  \tcblower
  This is the lower part.
\end{tcolorbox}%
\begin{tcblisting}{}
\textbf{Mixed}
with a listing.
\end{tcblisting}
\begin{tcolorbox}[title={Fourth box}]
  My final box.
\end{tcolorbox}%
```



3.9 Box Content Additions

The following options introduce some arbitrary ⟨*code*⟩ to the content of a `tcolorbox`. These additions can be given at the beginning or at the ending of the title, the upper part, or the lower part.

/tcb/before title=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the title.

```
\tcbset{before title={\textcolor{yellow}{\large Important:}~},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Important: My title

This is a **tcolorbox**.

/tcb/after title=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the title.

```
\tcbset{after title={\hfill\colorbox{Navy}{approved}},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title approved

This is a **tcolorbox**.

/tcb/before upper=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the upper part.

```
\tcbset{before upper={\textit{The story:}\par},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

The story:
This is a **tcolorbox**.

/tcb/after upper=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the upper part.

```
\tcbset{after upper={\par\hfill\textit{Read more next week}},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

Read more next week

/tcb/before lower=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the color and font settings and *before* the content of the lower part.

```
\tcbset{before lower=\textit{Behold:~},colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

Behold: This is the lower part.

/tcb/after lower=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is placed *after* the content of the lower part.

```
\tcbset{after lower=\ \textit{This is the end.},
  colback=red!5!white,colframe=red!75!black}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part. *This is the end.*

/tcb/center title (style, no value, initially unset)

This style sets /tcb/before title → P. 46 to \centering.

```
\tcbset{center title,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/center upper (style, no value, initially unset)

This style sets /tcb/before upper → P. 46 to \centering.

```
\tcbset{center upper,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/center lower (style, no value, initially unset)

This style sets /tcb/before lower → P. 47 to \centering.

```
\tcbset{center lower,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

`/tcb/flushleft title` (style, no value, initially unset)

This style sets `/tcb/before title` → P. 46 to `\raggedright`.

```
\tcbset{flushleft title,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title which is
  quite long but always flushed left]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title which is quite long but always flushed left

This is a **tcolorbox**.

`/tcb/flushleft upper` (style, no value, initially unset)

This style sets `/tcb/before upper` → P. 46 to `\raggedright`.

```
\tcbset{flushleft upper,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}. Here, the
content is flushed left.
You see the effect on the right hand side.
\end{tcolorbox}
```

My title

This is a **tcolorbox**. Here, the content is flushed left. You see the effect on the right hand side.

`/tcb/flushleft lower` (style, no value, initially unset)

This style sets `/tcb/before lower` → P. 47 to `\raggedright`.

```
\tcbset{flushleft lower,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}. The upper part
is not flushed left or right. This could be
unpleasant for small boxes.
\tcblower
This is the lower part. Here, the
content is flushed left.
You see the effect on the right hand side.
\end{tcolorbox}
```

My title

This is a **tcolorbox**. The upper part is not flushed left or right. This could be unpleasant for small boxes.

This is the lower part. Here, the content is flushed left. You see the effect on the right hand side.

`/tcb/flushright title` (style, no value, initially unset)

This style sets `/tcb/before title` → P. 46 to `\raggedleft`.

```
\tcbset{flushright title,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

`/tcb/flushright upper` (style, no value, initially unset)

This style sets `/tcb/before upper` → P. 46 to `\raggedleft`.

```
\tcbset{flushright upper,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

`/tcb/flushright lower` (style, no value, initially unset)

This style sets `/tcb/before lower` → P. 47 to `\raggedleft`.

```
\tcbset{flushright lower,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

If /tcb/tabularx or /tcb/tabularx* are used, one cannot have a lower part.

/tcb/tabularx=⟨*preamble*⟩ (style)

This style sets /tcb/before upper → P. 46 and /tcb/after upper → P. 47 and several geometry keys to support a tabularx with the given ⟨*preamble*⟩. The packages tabularx [4], array, and colortbl have to be loaded separately.

```
% \usepackage{array,tabularx}
% \usepackage{colortbl} - or - \usepackage[table]{xcolor}
\newcolumntype{Y}{>{\raggedleft\arraybackslash}X}% see tabularx
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title}

\begin{tcolorbox}[tabularx={X||Y|Y|Y|Y||Y},title=My table]
Group & One     & Two     & Three    & Four     & Sum\\\hline\hline
Red   & 1000.00 & 2000.00 &  3000.00 &  4000.00 & 10000.00\\\hline
Green & 2000.00 & 3000.00 &  4000.00 &  5000.00 & 14000.00\\\hline
Blue  & 3000.00 & 4000.00 &  5000.00 &  6000.00 & 18000.00\\\hline\hline
Sum   & 6000.00 & 9000.00 & 12000.00 & 15000.00 & 42000.00
\end{tcolorbox}
```

My table

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

/tcb/tabularx*={⟨*code*⟩}{⟨*preamble*⟩} (style)

This is a variant of /tcb/tabularx which adds some ⟨*code*⟩ before the table starts.

```
% \usepackage{array,tabularx}
% \usepackage{colortbl} - or - \usepackage[table]{xcolor}
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title}

\begin{tcolorbox}[tabularx*={\arrayrulewidth0.5mm}{X|X|X},title=My table]
One     & Two     & Three \\\hline\hline
1000.00 & 2000.00 &  3000.00\\\hline
2000.00 & 3000.00 &  4000.00
\end{tcolorbox}
```

My table

One	Two	Three
1000.00	2000.00	3000.00
2000.00	3000.00	4000.00

/tcb/tikz upper=⟨*options*⟩ (style)

This style adds a centered `tikzpicture` environment to the start and end of the upper part. The ⟨*options*⟩ may be given as Ti*k*Z picture options.

```
% \usepackage{tikz}

\begin{tcolorbox}[tikz upper,fonttitle=\bfseries,colback=white,colframe=black,
                  title=\tikzname\ drawing]
  \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
  \fill[red] (45:5mm) circle (1mm);
  \fill[red] (135:5mm) circle (1mm);
  \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);
\end{tcolorbox}
```

Ti*k*Z drawing

/tcb/tikz lower=⟨*options*⟩ (style)

This style adds a centered `tikzpicture` environment to the start and end of the lower part. The ⟨*options*⟩ may be given as Ti*k*Z picture options.

```
% \usepackage{tikz}
% \usetikzlibrary{skins,listings}
\tcbset{tikz lower,listing side text,fonttitle=\bfseries,
  bicolor,colback=LightBlue!50!white,colbacklower=white,colframe=black,
  righthand width=3cm}

\begin{tcblisting}{title=\tikzname\  drawing}
\path[fill=yellow,draw=yellow!75!red]
    (0,0) circle (1cm);
\fill[red] (45:5mm) circle (1mm);
\fill[red] (135:5mm) circle (1mm);
\draw[line width=1mm,red]
    (215:5mm) arc (215:325:5mm);
\end{tcblisting}
```

Ti*k*Z drawing

```
\path[fill=yellow,draw=yellow!75!red]
    (0,0) circle (1cm);
\fill[red] (45:5mm) circle (1mm);
\fill[red] (135:5mm) circle (1mm);
\draw[line width=1mm,red]
    (215:5mm) arc (215:325:5mm);
```

/tcb/tikznode upper=⟨*options*⟩ (style)

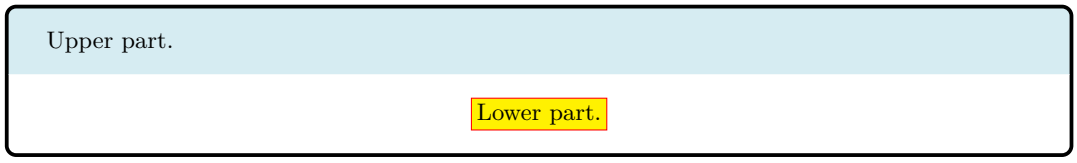
This style places the upper part content into a centered Ti*k*Z node. The ⟨*options*⟩ may be given as Ti*k*Z node options. This style is especially useful for boxes with multiline texts which are fitted to the text width.

```
% \usepackage{tikz}
\newtcbox{\headline}[1][]{enhanced,before=\begin{center},after=\end{center},
  fontupper=\Large\bfseries,colframe=red!50!black,colback=red!10!white,
  drop fuzzy shadow=yellow,tikznode upper,#1}

\headline{Important\\Headline}
```

/tcb/tikznode lower=⟨*options*⟩ (style)

This style places the lower part content into a centered Ti*k*Z node. The ⟨*options*⟩ may be given as Ti*k*Z node options.

```
% \usepackage{tikz}
\begin{tcolorbox}[bicolor,colback=LightBlue!50!white,colbacklower=white,
  colframe=black,tikznode lower={inner sep=2pt,draw=red,fill=yellow}]
Upper part.
\tcblower
Lower part.
\end{tcolorbox}
```



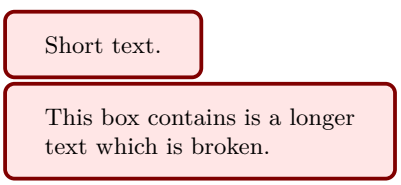
/tcb/tikznode=⟨*options*⟩ (style)

Shortcut for setting /tcb/tikznode upper and /tcb/tikznode lower the same time.

/tcb/varwidth upper=⟨*length*⟩ (style, default /tcb/width → P. 23)

This style places the upper part content into a `varwidth` environment. This style needs the `varwidth` package [1] to be loaded manually. The resulting box has a maximal width of ⟨*length*⟩. This option is only senseful for a \tcbox → P. 11.

```
% \usepackage{varwidth}
\newtcbox{\varbox}{colframe=red!50!black,
  colback=red!10!white,varwidth upper}

\varbox{Short text.}
\varbox{This box contains is a longer text
  which is broken.}
```



`/tcb/code`=⟨*code*⟩ (no default, initially unset)

The given ⟨*code*⟩ is executed immediately. This option is useful to place some arbitrary code into an option list.

```
\tcbset{colback=red!5!white,colframe=red!75!black,
  code={Useless at this spot but functional.},
  fonttitle=\bfseries}

\begin{tcolorbox}[code={\newcommand{\mycommand}{\textit{working}}},
  title=My \mycommand\ title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Useless at this spot but functional.

My *working* title

This is a **tcolorbox**.

3.10 Overlays

With an overlay, arbitrary ⟨*graphical code*⟩ can be added to a `tcolorbox`. This code is executed *after* the frame and interior are drawn and *before* the text content is drawn. Therefore, you can decorate the `tcolorbox` with your own extensions. Common special cases are *watermarks* which are implemented using overlays. See Subsection 7.3 from page 114 if you want to add *watermarks*.

If you use the core package only, the ⟨*graphical code*⟩ has to be `pgf` code and there is not much assistance for positioning. Therefore, the usage of the `/tcb/enhanced` → P. 150 mode from the library skins is recommended which allows `tikz` code and gives access to `/tcb/geometry nodes` → P. 92 for positioning.

`/tcb/overlay`=⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the frame and interior and *before* the text content.

```
% \tcbuselibrary{skins} % preamble
\tcbset{frogbox/.style={enhanced,colback=green!10,colframe=green!65!black,
  enlarge top by=5.5mm,
  overlay={\foreach \x in {2cm,3.5cm} {
    \begin{scope}[shift={([xshift=\x]frame.north west)}]
      \path[draw=green!65!black,fill=green!10,line width=1mm] (0,0) arc (0:180:5mm);
      \path[fill=black] (-0.2,0) arc (0:180:1mm);
    \end{scope}}}]}}

\begin{tcolorbox}[frogbox,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

```
% \usetikzlibrary{patterns} % preamble
% \tcbuselibrary{skins}     % preamble
\tcbset{ribbonbox/.style={enhanced,colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,
  overlay={\path[fill=blue!75!white,draw=blue,double=white!85!blue,
    preaction={opacity=0.6,fill=blue!75!white},
    line width=0.1mm,double distance=0.2mm,
    pattern=fivepointed stars,pattern color=white!75!blue]
    ([xshift=-0.2mm,yshift=-1.02cm]frame.north east)
    -- ++(-1,1) -- ++(-0.5,0) -- ++(1.5,-1.5) -- cycle;}}}

\begin{tcolorbox}[ribbonbox,title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

`/tcb/no overlay` (style, no default, initially set)

Removes the overlay if set before.

`/tcb/overlay broken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay unbroken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 but *is not* broken actually or if the box is set to be `/tcb/unbreakable` → P. 256, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay first`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay middle`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay last`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay unbroken and first`=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting `/tcb/overlay unbroken` and `/tcb/overlay first` together. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay middle and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting `/tcb/overlay middle` and `/tcb/overlay last` together. `/tcb/overlay` → P. 55 overwrites this key.

`/tcb/overlay unbroken and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting `/tcb/overlay unbroken` and `/tcb/overlay last` together. `/tcb/overlay` → P. 55 overwrites this key.

N 2014-09-19 `/tcb/overlay first and middle`=⟨*graphical code*⟩ (no default, initially unset)

This is an optimized abbreviation for setting `/tcb/overlay first` and `/tcb/overlay middle` together. `/tcb/overlay` → P. 55 overwrites this key.

This example demonstrates the application of break sequence specific overlay options. Here, we define an environment `myexample` based on `tcolorbox` where the visible drawing is done totally by overlay keys.
Here, the first application of `myexample` produces an unbroken `tcolorbox`. The frame is drawn by the code given with `/tcb/overlay unbroken`.
The second application of `myexample` is broken into several parts which are drawn by the codes given with `/tcb/overlay first`, `/tcb/overlay middle`, and `/tcb/overlay last`.

```
% Preamble:
%\usepackage{tikz,lipsum}
%\tcbuselibrary{skins,breakable}
%\newcounter{example}
\colorlet{colexam}{red!75!black}
```

```
\newtcolorbox[use counter=example]{myexample}{%
  empty,title={Example \thetcbcounter},attach boxed title to top left,
  boxed title style={empty,size=minimal,toprule=2pt,top=4pt,
    overlay={\draw[colexam,line width=2pt]
      ([yshift=-1pt]frame.north west)--([yshift=-1pt]frame.north east);}},
  coltitle=colexam,fonttitle=\Large\bfseries,
  before=\par\medskip\noindent,parbox=false,boxsep=0pt,left=0pt,right=3mm,top=4pt,
  breakable,pad at break=0mm,
  overlay unbroken={\draw[colexam,line width=1pt]
    ([yshift=-1pt]title.north east)--([xshift=-0.5pt,yshift=-1pt]title.north-|frame.east)
    --([xshift=-0.5pt]frame.south east)--(frame.south west); },
  overlay first={\draw[colexam,line width=1pt]
    ([yshift=-1pt]title.north east)--([xshift=-0.5pt,yshift=-1pt]title.north-|frame.east)
    --([xshift=-0.5pt]frame.south east); },
  overlay middle={\draw[colexam,line width=1pt] ([xshift=-0.5pt]frame.north east)
    --([xshift=-0.5pt]frame.south east); },
  overlay last={\draw[colexam,line width=1pt] ([xshift=-0.5pt]frame.north east)
    --([xshift=-0.5pt]frame.south east)--(frame.south west);},%
}

\begin{myexample}
\lipsum[1]
\end{myexample}

\begin{myexample}
\lipsum[2-11]
\end{myexample}

\lipsum[12]% following text
```

Example 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Example 2

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed

lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Floating box from floatplacement

This floating box is placed at the top of a page.

I am floating

3.11 Floating Objects

/tcb/floatplacement=⟨*values*⟩ (no default, initially htb)

Sets ⟨*values*⟩ as default values for the usage of /tcb/float and /tcb/float*. Feasible are the usual parameters for floating objects.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,
   watermark color=red!15!white}

\begin{tcolorbox}[floatplacement=t,float,
                  title=Floating box from |floatplacement|,
                  watermark text={I am floating}]
  This floating box is placed at the top of a page.
\end{tcolorbox}
```

/tcb/float=⟨*values*⟩ (default from floatplacement)

Turns the box to a floating object where ⟨*values*⟩ are the usual parameters for such floating objects. If they are not used, the placement uses the default values given by floatplacement.

```
\begin{tcolorbox}[float, title=Floating box from |float|,
    enhanced,watermark text={I'm also floating}]
  This box floats to a feasible place automatically. You do not have to
  use a numbering for this floating object.
\end{tcolorbox}
```

Floating box from float

This box floats to a feasible place automatically. You do not have to use a numbering for this floating object.

I'm also floating

/tcb/float*=⟨*values*⟩ (default from floatplacement)

Identical to /tcb/float, but for wide boxes spanning the whole page width of two column documents or in conjunction with the packages multicol or paracol. Note that you have to set width=\textwidth additionally, if the box should span the whole page width in these cases!

```
\begin{tcolorbox}[float*=b, title=Floating box from |float*|,width=\textwidth,
    enhanced,watermark text={I'm also floating}]
  In this single column document, you will see no difference to |float|.
\end{tcolorbox}
```

/tcb/nofloat (style, initially set)

Turns the floating behavior off.

Floating box from float*

In this single column document, you will see no difference to float.

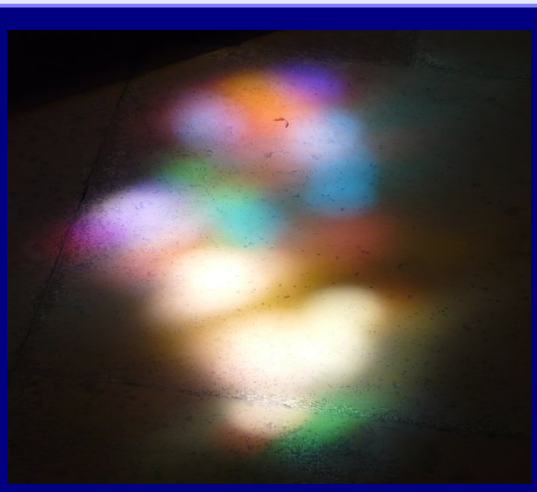
I'm also floating

N 2014-09-19 **/tcb/every float=**⟨*code*⟩ (no default, initially empty)

For floating objects, the /tcb/before → P. 65 and /tcb/after → P. 65 settings are ignored. Instead, the given ⟨*code*⟩ is inserted before a floating box. If the box is /tcb/breakable → P. 255, the given ⟨*code*⟩ is inserted before every part of the break sequence. The most common use case is every float=\centering.

```
\tcbox[float=htb,title={Floating box},every float=\centering,
  colback=blue!50!black,colframe=blue!50!white,colbacktitle=blue!10!white,
  coltitle=black,center title]
  {\includegraphics[height=6cm]{lichtspiel.jpg}}
```

Floating box



3.12 Side by Side

Further side by side options for code examples are /tcb/listing side text→ P. 214, /tcb/text side listing→ P. 214, /tcb/listing outside text→ P. 214, and /tcb/text outside listing→ P. 214.

/tcb/sidebyside=true|false (default `true`, initially `false`)

Normally, the upper part and the lower part of the box have their positions as their names suggest. If `sidebyside` is set to `true`, the upper part is drawn *left-handed* and the lower part is drawn *right-handed*. Both parts are drawn together with the geometry settings of the upper part but the space is divided horizontally according to the following options. Colors, fonts, and box content additions are used individually. The resulting box is unbreakable.

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

\begin{tcolorbox}[title=My title,sidebyside]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/sidebyside align=⟨*alignment value*⟩ (no default, initially `center`)

Sets the vertical alignment for the left-handed and right-handed part. Feasible values are `center`, `top`, and `bottom`.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,nobeforeafter,
  left=2mm,right=2mm,sidebyside,sidebyside gap=6mm,width=(\linewidth-2mm)/3}

\begin{tcolorbox}[adjusted title=center,sidebyside align=center]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=top,sidebyside align=top]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=bottom,sidebyside align=bottom]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}
```

center

This is a text which is too long for one line.

This is a short text.

top

This is a text which is too long for one line.

This is a short text.

bottom

This is a text which is too long for one line.

This is a short text.

/tcb/sidebyside gap=⟨*length*⟩ (no default, initially 10mm)

Sets the horizontal distance between the left-handed and right-handed part to ⟨*length*⟩.

```
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,nobeforeafter,
  sidebyside,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[adjusted title=Wide gap,sidebyside gap=30mm]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=Narrow gap,sidebyside gap=1mm]
This is a text which is too long for one line.
\tcblower
This is a short text.
\end{tcolorbox}
```

Wide gap

This is a text which is too long for one line.

This is a short text.

Narrow gap

This is a text which is too long for one line.

This is a short text.

/tcb/lefthand width=⟨*length*⟩ (no default, initially unset)

Sets the width of the left-handed part to the given ⟨*length*⟩.

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

\begin{tcolorbox}[title=My title,sidebyside,lefthand width=3cm]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/righthand width=⟨*length*⟩ (no default, initially unset)

Sets the width of the right-handed part to the given ⟨*length*⟩.

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

\begin{tcolorbox}[title=My title,sidebyside,righthand width=3cm]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/lefthand ratio=⟨*fraction*⟩ (no default, initially 0.5)

Sets the width of the left-handed part to the given ⟨*fraction*⟩ of the available space. ⟨*fraction*⟩ is a value between 0 and 1.

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

\begin{tcolorbox}[title=My title,sidebyside,lefthand ratio=0.25]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

/tcb/righthand ratio=⟨*fraction*⟩ (no default, initially 0.5)

Sets the width of the right-handed part to the given ⟨*fraction*⟩ of the available space. ⟨*fraction*⟩ is a value between 0 and 1.

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

\begin{tcolorbox}[title=My title,sidebyside,righthand ratio=0.25]
This is the upper (\textit{left-handed}) part.
\tcblower
This is the lower (\textit{right-handed}) part.
\end{tcolorbox}
```

My title

This is the upper (*left-handed*) part.

This is the lower (*right-handed*) part.

3.13 Embedding into the Surroundings

Typically, but not necessarily, a `tcolorbox` is put inside a separate paragraph and has some vertical space before and after it. This behavior is controlled by the keys `before` and `after`.

`/tcb/before`=⟨*code*⟩ (no default, initially see `/tcb/autoparskip`)
Sets the ⟨*code*⟩ which is executed before the colored box. It is not used for floating boxes.

`/tcb/after`=⟨*code*⟩ (no default, initially see `/tcb/autoparskip`)
Sets the ⟨*code*⟩ which is executed after the colored box. It is not used for floating boxes.

`/tcb/parskip` (style, no value)
Sets the keys `before` and `after` to values which are recommended, if the package `parskip` *is* used and there is no better idea for `before` and `after`.

```
\tcbset{parskip/.style={before={\par\pagebreak[0]\parindent=0pt},
                        after={\par}}}
```

`/tcb/noparskip` (style, no value)
Sets the keys `before` and `after` to values which are recommended, if the package `parskip` is *not* used and there is no better idea for `before` and `after`.

```
\tcbset{noparskip/.style={before={\par\smallskip\pagebreak[0]\parindent=0pt},
                          after={\par\smallskip}}}
```

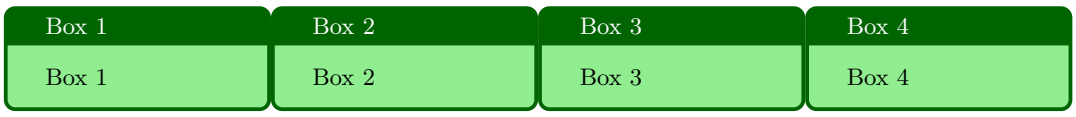
`/tcb/autoparskip` (style, no value, initially set)
Tries to detect the usage of the package `parskip` and sets the keys `before` and `after` accordingly. Actually, the following is done:

- If the length of `\parskip` is greater than `0pt` at the beginning of the document, `/tcb/parskip` is executed. Here, the usage of package `parskip` is *assumed*.
- Otherwise, if the length of `\parskip` is not greater than `0pt` at the beginning of the document, `/tcb/noparskip` is executed. Here, the absence of package `parskip` is *assumed*.

`autoparskip` is the default for the package `tcolorbox`, if `before` or `after` are not changed otherwise.

`/tcb/nobeforeafter` (style, no value)
Abbreviation for clearing the keys `before` and `after`. The colored box is not put into a paragraph and there is no space before or after the box.

```
\tcbset{myone/.style={colback=LightGreen,colframe=DarkGreen,
  equal height group=nobefaf,width=\linewidth/4,nobeforeafter}}
\begin{tcolorbox}[myone,title=Box 1]Box 1\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 2]Box 2\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 3]Box 3\end{tcolorbox}%
\begin{tcolorbox}[myone,title=Box 4]Box 4\end{tcolorbox}
```



`/tcb/baseline`=⟨*length*⟩ (no default, initially `0pt`)
Used to set the `\pgfsetbaseline` value of the resulting `tcolorbox`.

N 2014-10-10 **/tcb/before skip=**⟨*length*⟩ (style, no default)

Inserts some vertical space of the given ⟨*length*⟩ before the colored box. This style sets /tcb/before → P. 65.

```
Some text.
\begin{tcolorbox}[before skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

Some text.

This is a **tcolorbox**.

N 2014-10-10 **/tcb/after skip=**⟨*length*⟩ (style, no default)

Inserts some vertical space of the given ⟨*length*⟩ after the colored box. This style sets /tcb/after → P. 65.

```
\begin{tcolorbox}[after skip=1cm,
    colframe=red!50!white]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
Some text.
```

This is a **tcolorbox**.

Some text.

N 2014-10-10 **/tcb/beforeafter skip=**⟨*length*⟩ (style, no default)

Inserts some vertical space of the given ⟨*length*⟩ before *and* after the colored box. This style sets /tcb/before → P. 65 and /tcb/after → P. 65.

```
\tcbset{beforeafter skip=0pt,
  colframe=red!50!white}
\begin{tcolorbox}
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}
  Second box.
\end{tcolorbox}
```

This is a **tcolorbox**.

Second box.

3.14 Bounding Box

Normally, every `tcolorbox` has a bounding box which fits exactly to the dimensions of the outer frame. Therefore, LaTeX reserves exactly the space needed for the box. This behavior can be changed by enlarging (or shrinking) the bounding box. If the bounding box is enlarged, the `tcolorbox` will get some clearance around it. If the bounding box is shrunk, i. e. enlarged with negative values, the `tcolorbox` will overlap to other parts of the page. For example, the `tcolorbox` could be stretched into the page margin.

The following examples use `/tcb/show bounding box` → P. 128 to display the actual bounding box. For this, the library LIB skins has to be included and `/tcb/enhanced` → P. 150 has to be set.

`/tcb/enlarge top initially by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩. If the box is *breakable*, only the first box of the break sequence gets enlarged. `/tcb/enlarge top by` → P. 68 overwrites this key.

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

\begin{tcolorbox}[enlarge top initially by=-5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge top initially by=5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/enlarge bottom finally by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩. If the box is *breakable*, only the last box of the break sequence gets enlarged. `/tcb/enlarge bottom by` → P. 68 overwrites this key.

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

\begin{tcolorbox}[enlarge bottom finally by=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge bottom finally by=-5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

`/tcb/enlarge top at break by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩, *if* the box is `/tcb/breakable` → P. 255. In this case, it is applied to *middle* and *last* parts in a break sequence. `/tcb/enlarge top by` overwrites this key.

`/tcb/enlarge bottom at break by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩, *if* the box is `/tcb/breakable` → P. 255. In this case, it is applied to *first* and *middle* parts in a break sequence. `/tcb/enlarge bottom by` overwrites this key.

`/tcb/enlarge top by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the top of the box by ⟨*length*⟩. `/tcb/enlarge top initially by` → P. 67 and `/tcb/enlarge top at break by` are set to ⟨*length*⟩.

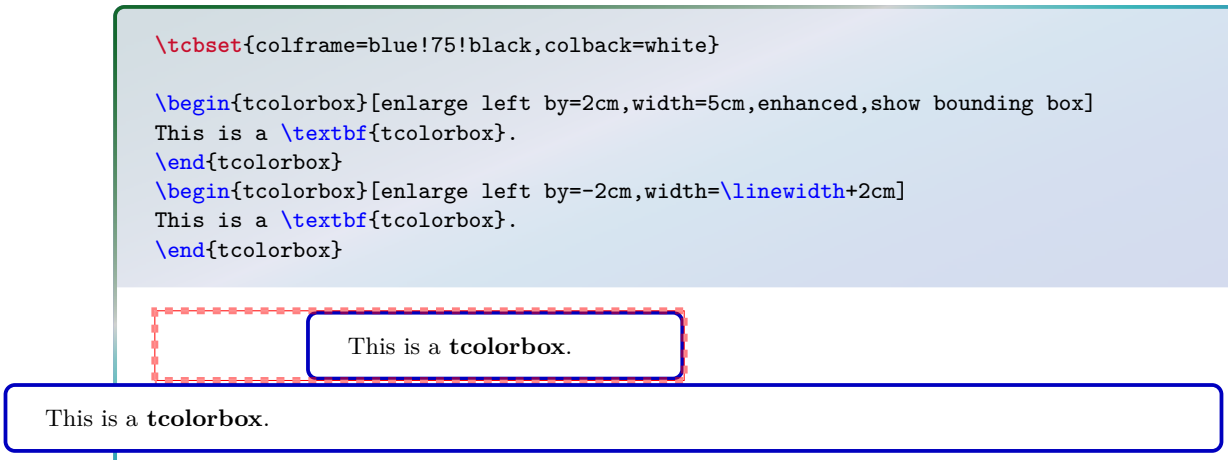
`/tcb/enlarge bottom by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the bottom of the box by ⟨*length*⟩. `/tcb/enlarge bottom finally by` → P. 67 and `/tcb/enlarge bottom at break by` are set to ⟨*length*⟩.

`/tcb/enlarge left by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the left side of the box by ⟨*length*⟩.

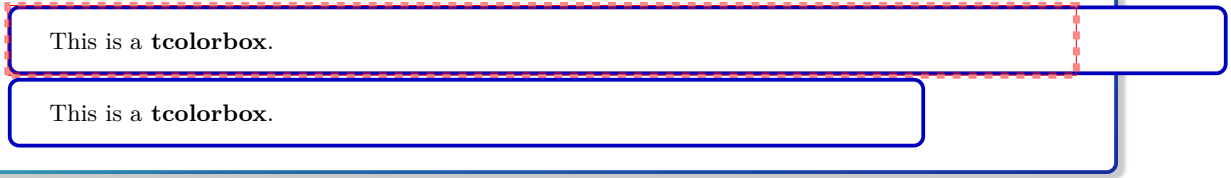
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge left by=2cm,width=5cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge left by=-2cm,width=\linewidth+2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



`/tcb/enlarge right by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to the right side of the box by ⟨*length*⟩.

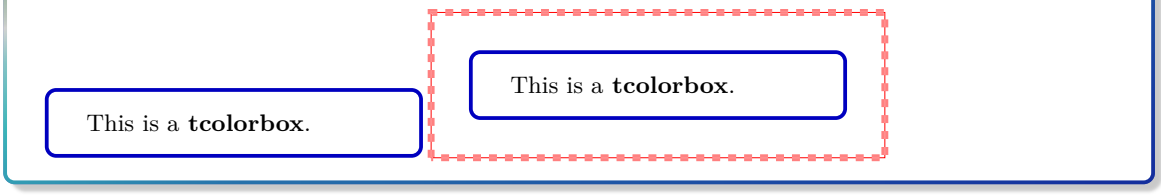
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enlarge right by=-2cm,width=\linewidth+2cm,
  enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge right by=2cm,width=\linewidth-2cm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



`/tcb/enlarge by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the bounding box distance to all sides of the box by ⟨*length*⟩.

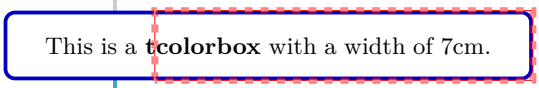
```
\tcbset{colframe=blue!75!black,colback=white,width=5cm,nobeforeafter}

\begin{tcolorbox}
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge by=5mm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



`/tcb/grow to left by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the current box size by ⟨*length*⟩ and enlarges (shrinks) the bounding box distance to the left side of the box by −⟨*length*⟩.

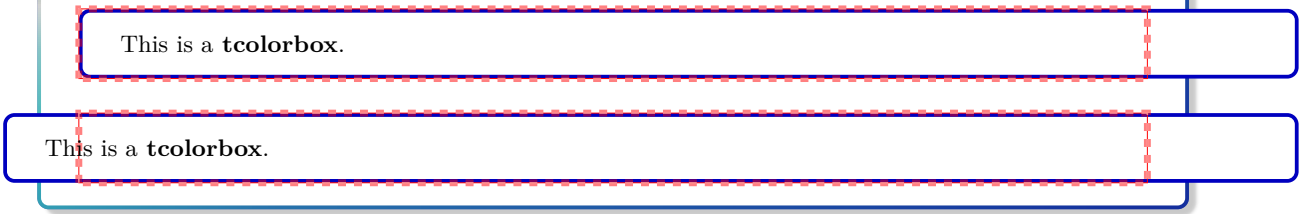
```
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[width=5cm,grow to left by=2cm,enhanced,show bounding box]
This is a \textbf{tcolorbox} with a width of 7cm.
\end{tcolorbox}
```



`/tcb/grow to right by=`⟨*length*⟩ (no default, initially `0mm`)

Enlarges the current box size by ⟨*length*⟩ and enlarges (shrinks) the bounding box distance to the right side of the box by −⟨*length*⟩.

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

\begin{tcolorbox}[grow to right by=2cm,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}

\bigskip

\begin{tcolorbox}[grow to right by=2cm,grow to left by=1cm,
  enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



Floating box from `toggle enlargement`

This page is an even page. Therefore, the left and right enlargements are toggled (with some luck; otherwise use `forced`). This box stretches to the right margin on odd pages and to the left margin on even pages. The current document is one-sided – this feature makes sense for two-sided documents only.

`/tcb/toggle enlargement`=⟨*toggle preset*⟩ (default `evenpage`, initially `none`)

According to the ⟨*toggle preset*⟩, the left and the right enlargements of the bounding box are switched or not. Feasible values are:

- `none`: no switching.
- `forced`: the values of the left and right enlargement are switched.
- `evenpage`: if the page is an even page, the values of the left and right enlargement are switched. It is recommended that one use this setting in conjunction with `/tcb/check odd page` → P. 80.

```
% \usepackage{changepage} for 'check odd page'
\tcbset{colframe=blue!75!black,colback=white,
  grow to left by=20mm,grow to right by=-5mm}

\begin{tcolorbox}[toggle enlargement=none,enhanced,show bounding box]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=forced]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=evenpage,check odd page]
  This page is an \ifthenelse{\isodd{\thepage}}{odd}{even} page.
  Therefore, the left and right enlargements
  \ifthenelse{\isodd{\thepage}}{are not}{are} toggled.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

This page is an even page. Therefore, the left and right enlargements are toggled.

```
% \usepackage{changepage} for 'check odd page'
\begin{tcolorbox}[colframe=red!60!black,colback=red!15!white,check odd page,
  fonttitle=\bfseries,title=Floating box from \texttt{toggle enlargement},
  width=\textwidth,grow to right by=2cm,toggle enlargement=evenpage,float=t]
  This page is an \ifthenelse{\isodd{\thepage}}{odd}{even} page.
  Therefore, the left and right enlargements
  \ifthenelse{\isodd{\thepage}}{are not}{are} toggled (with some luck; otherwise
  use |forced|). This box stretches to the right margin on odd pages and to the left
  margin on even pages. The current document is one-sided -- this feature makes
  sense for two-sided documents only.
\end{tcolorbox}
```

The following keys should not be used with breakable boxes or boxes with a lower part.

/tcb/shrink tight (style, no value, initially unset)

The total colored box is shrunk to the dimensions of the upper part. There should be no lower part and no title. This style sets the /tcb/boxsep → P. 25 to 0pt and other geometry keys to fitting values. This option is likely to be used with the following extrusion keys.

```
\tcbset{colframe=blue!75!black,colback=white,arc=0mm,boxrule=0.4pt,
        nobeforeafter,tcbox raise base,shrink tight}

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

Lorem \tcbox{ipsum} dolor sit amet, consectetuer adipiscing elit.
```

This is a **tcolorbox**.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

U 2014-09-19 **/tcb/extrude left by=⟨*length*⟩** (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the left side. The inner width and the bounding box is kept unchanged and the operation is additiv!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude left by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

U 2014-09-19 **/tcb/extrude right by=⟨*length*⟩** (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the right side. The inner width and the bounding box is kept unchanged and the operation is additiv!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude right by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude top by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the top side. The inner width and the bounding box is kept unchanged and the operation is additiv!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude top by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude bottom by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to the bottom side. The inner width and the bounding box is kept unchanged and the operation is additiv!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis.
\tcbox[extrude bottom by=1cm]{Curabitur} dictum gravida mauris.
Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

/tcb/extrude by=⟨*length*⟩ (style, no default, initially unset)

The (upper part of the) colored box is extruded by the given ⟨*length*⟩ to all sides. The inner width and the bounding box is kept unchanged and the operation is additiv!

```
\tcbset{enhanced,colframe=red,colback=yellow!25!white,
  frame style={opacity=0.25},interior style={opacity=0.5},
  nobeforeafter,tcbox raise base,shrink tight,extrude by=2mm}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit,
vestibulum ut, placerat ac, adipiscing vitae, felis. \tcbox{Curabitur} dictum
gravida mauris. \tcbox[colframe=Green,interior style={opacity=0.0}]{Nam}
arcu libero, nonummy eget, consectetuer id, \tcbox{vulputate} a, magna. Donec
vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus
et malesuada fames ac turpis egestas. \tcbox{Mauris ut leo.}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

3.15 Layered Boxes and Every Box Settings

A `tcolorbox` may contain another `tcolorbox` and so on. The package takes track of the nesting level using a counter `tcblayer`. Counter values may be used for doing some fancy things, but you should never change the counter value yourself.

The package takes special care for the first four layers or nesting levels, called managed layers. Here, footnote texts are administrated to find their intended place and specific layer dependent options may be set by changing `/tcb/every box on layer n` → P. 74. If needed, the number of managed layers can be increased by setting `\tcbsetmanagedlayer` → P. 74 to a higher value than 4.

The following styles have a considerable influence on how layered boxes are processed. Note especially that nested boxes are getting a `/tcb/reset` → P. 81 by default. You can change this, but be prepared for suprises if you do.

If the defaults are *not changed*, a `tcolorbox` gets its options in the following order. Following options overwrite preceding options.

1. On package load, all options are set to default values.
2. Every `\tcbset` → P. 10 command adds or changes options for the following boxes inside the current TEX group.
3. While entering a `tcolorbox`, a `/tcb/every box on layer n` → P. 74 or `/tcb/every box on higher layers` → P. 74 option list is applied. With default settings this means:
   - For layer 1 (lowest layer), the `/tcb/every box` option list is applied. Not overwritten options given by a preceding `\tcbset` → P. 10 survive.
   - For layer 2 and above (nested boxes), a `/tcb/reset` → P. 81 followed by `/tcb/every box` option list is applied. Every resettable options given by a preceding `\tcbset` → P. 10 and by the sourrounding box(es) are reset.
4. The ⟨*options*⟩ given to the `tcolorbox` are applied. Or, if the box was generated by `\newtcolorbox` → P. 12 or friends, the ⟨*options*⟩ given there are applied.
5. If the box was generated by `\newtcolorbox` → P. 12 or friends, some automated options are applied.

`/tcb/every box` (style)

By default, this style is empty.

```
% default setting:
\tcbset{every box/.style={}}
```

It may be changed by redefining this style.

```
% setting all boxes to be enhanced:
\tcbset{every box/.style={enhanced}}
```

The alternative for setting something for every box (on every layer) is `\tcbsetforeverylayer` → P. 10:

```
% setting all boxes to be enhanced:
\tcbsetforeverylayer{enhanced}
```

/tcb/every box on layer n (style)

Here, n has to be replaced by a number ranging from 1 to the highest managed layer number (4 by default).

```
% default settings:
\tcbset{
  every box on layer 1/.style={every box},
  every box on layer 2/.style={reset,every box},
  every box on layer 3/.style={reset,every box},
  every box on layer 4/.style={reset,every box},
  }
```

/tcb/every box on higher layers (style)

Higher layers are layers above the highest managed layer number (4 by default).

```
\tcbset{every box on higher layers/.style={reset,every box}}
```

\tcbsetmanagedlayer{⟨*number*⟩**}**

Replaces the highest managed layer number by ⟨*number*⟩ where 4 is the default. This macro can only be used inside the preamble. Using a ⟨*number*⟩ lower than 4 typically makes no sense, but is not forbidden.

```
% \usepackage{lipsum}
% \tcbuselibrary{skins,breakable}
\tcbset{enhanced jigsaw,breakable,colframe=red!75!black,fonttitle=\bfseries,
  colback=red!5!white,
  every box/.style={enhanced,watermark text=\thetcblayer,
    before=\par\smallskip,after=\par\smallskip},
  every box on layer 2/.style={reset,every box,colback=yellow!10!white,
    drop fuzzy shadow}}
\begin{tcolorbox}[title=Layer 1 Box]
Here comes a footnote\footnote{Footnote from layer 1 box}.
\lipsum[2]
  \begin{tcolorbox}[title=Layer 2 Box]
  abc\footnote{The footnote of abc}
  \end{tcolorbox}
  \begin{tcolorbox}[title=Another Box,ams equation]
    \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} = \infty.
  \end{tcolorbox}
Some text\footnote{Footnote from some text}.
  \begin{tcolorbox}[title=Yet Another Box]
    \tcboxfit[height=2cm]{\lipsum[1]}
    My text.
    \begin{tcolorbox}
      Another lipsum text\footnote{A lipsum text}. \lipsum[3]
      \begin{tcolorbox}[title=Layer 4,colframe=blue,colback=white]
        Layer 4\footnote{Layer 4 footnote}
      \end{tcolorbox}
      The End\footnote{Last footnote}.
    \end{tcolorbox}
  \end{tcolorbox}
\end{tcolorbox}
```

Layer 1 Box

Here comes a footnote[a]. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies

et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Layer 2 Box

abc[a]

[a]The footnote of abc

Another Box

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{1}$$

Some text[b].

Yet Another Box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My text.

Another lipsum text[a]. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Layer 4

Layer 4[a]

[a]Layer 4 footnote

The End[b].

[a]A lipsum text
[b]Last footnote

[a]Footnote from layer 1 box
[b]Footnote from some text

3.16 Capture Mode

`/tcb/capture`=⟨*mode*⟩ (no default, initially `minipage`)

The capture ⟨*mode*⟩ defines how the box content is processed.

Feasible values for ⟨*mode*⟩ are:

- `minipage`:
  This is the default ⟨*mode*⟩ for `tcolorbox` → P. 9. The content may have an upper and a lower part. Optionally, the box can be `/tcb/breakable` → P. 255. The box content is put into a minipage or into something similar to a minipage.
- `hbox`:
  This is the default ⟨*mode*⟩ for `\tcbox` → P. 11. The content cannot have a lower part and cannot be broken. The colored box is sized according to the dimensions of the content. A shortcut to set this mode is `/tcb/hbox`.
- `fitbox`: (needs the LIB fitting library)
  This is the default ⟨*mode*⟩ for `\tcboxfit` → P. 273. The content cannot have a lower part and cannot be broken. The content is sized according to the dimensions of the colored box. A shortcut to set this mode is `/tcb/fit` → P. 275.

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

\begin{tcolorbox}[capture=minipage]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[capture=hbox]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[capture=fitbox,height=9mm]% needs the 'fitting' library
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

This is a tcolorbox.

This is a tcolorbox.

`/tcb/hbox` (style, no default)

Shortcut for `capture=hbox`.

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

\begin{tcolorbox}[hbox]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/minipage` (style, no default)

Shortcut for `capture=minipage`.

3.17 Text Characteristics

`/tcb/parbox=true|false` (default `true`, initially `true`)

The text inside a `tcolorbox` is formatted using a LaTeX `minipage` if the box is unbreakable. If breakable, the box tries a mimicry of a `minipage`. In a `minipage` or `parbox`, paragraphs are formatted slightly different as the main text. If the key value is set to `false`, the normal main text behavior is restored. In some situations, this has some unwanted side effects. It is recommended that one use this experimental setting only where you really want to have this feature.

```
% \usepackage{lipsum}  % preamble
\tcbset{width=(\linewidth-2mm)/2,nobeforeafter,arc=1mm,
  colframe=blue!75!black,colback=white,fonttitle=\bfseries,fontupper=\small,
  left=2mm,right=2mm,top=1mm,bottom=1mm,equal height group=parbox}

\begin{tcolorbox}[parbox,adjusted title={parbox=true (normal)}]
  \lipsum[1-2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[parbox=false,adjusted title={parbox=false}]
  \lipsum[1-2]
\end{tcolorbox}%
```

parbox=true (normal)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

parbox=false

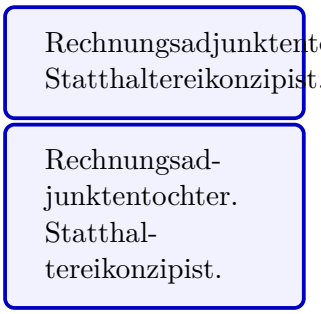
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/hyphenationfix=true|false` (default `true`, initially `false`)

Long words at the beginning of paragraphs in very narrow boxes will not be hyphenated using `pdflatex`. This problem is circumvented by applying the `hyphenationfix` option.

```
\tcbset{colframe=blue!75!black,
  fontupper=\normalsize,
  colback=blue!5!white,width=4cm}

\begin{tcolorbox}
Rechnungsadjunktentochter.\par
Statthaltereikonzipist.
\end{tcolorbox}

\begin{tcolorbox}[hyphenationfix]
Rechnungsadjunktentochter.\par
Statthaltereikonzipist.
\end{tcolorbox}
```



`parbox=false` and `hyphenationfix` should not be used together. They are targeting different box types and they do not blend very well.

3.18 Files

`/tcb/tempfile=`⟨*file name*⟩ (no default, initially `\jobname.tcbtemp`)

Sets ⟨*file name*⟩ as name for the temporary file which is used inside `tcbwritetemp` → P. 88 and `\tcbusetemp` → P. 88 implicitely.

3.19 `\tcbox` Specials

The following options are applicable for `\tcbox` → P. 11 and `\tcboxmath` → P. 233 only.

`/tcb/tcbox raise=`⟨*length*⟩ (no default, initially `0pt`)

Raises the `\tcbox` → P. 11 by the given ⟨*length*⟩.

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

Test\dotfill
\tcbox[tcbox raise base]{Hello World 1}\dotfill
\tcbox{Hello World 2}\dotfill
\tcbox[tcbox raise=5mm]{Hello World 3}
```

`/tcb/tcbox raise base` (style, no value, initially unset)

Raises the `\tcbox` → P. 11 such that the base of its content matches the base of the environmental line; see example above.

`/tcb/on line` (style, no value, initially unset)

Combines `/tcb/tcbox raise base` with `/tcb/nobeforeafter` → P. 65. The resulting box behaves analogue to `\fbox`.

3.20 Counters, Labels, and References

/tcb/phantom=⟨*code*⟩ (no default, initially unset)

The ⟨*code*⟩ is put in a box at the upper left corner of the `tcolorbox`. If the `tcolorbox` is breakable, the ⟨*code*⟩ is executed for the first box of the break sequence only. If there already was some phantom code given, the new ⟨*code*⟩ is appended.

The ⟨*code*⟩ is intended to be used for counter stepping, labelling, and related operations which do not produce visible text.

- The ⟨*code*⟩ is executed before the title and box content, i. e. counter values are ensured to be increased before usage.
- Labels are ensured to reference the correct page number.
- The ⟨*code*⟩ is executed only once even during fitting operations for title and box content.
- In combination with the `hyperref` package, the hyper anchor is set to the upper left corner of the `tcolorbox`, i. e. links inside the pdf document will jump to the box pleasantly.
- Since the ⟨*code*⟩ is executed inside a TEX group, only global operations can survive this group.

Examples for the `phantom` usage are given in Section 9.8 from page 225, e. g. Example 9.1 on page 226.

/tcb/nophantom (no value, initially set)

Removes the phantom code if set before.

/tcb/label=⟨*marker*⟩ (no default, initially unset)

The ⟨*marker*⟩ is set as label text for a reference with the `\ref` macro. Typically, this option is used for numbered boxes, see Subsection 4.1 from page 82, e. g. `/tcb/new/auto counter` → P. 82.

/tcb/label type=⟨*type*⟩ (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. ⟨*type*⟩ has to be a cross-reference type *known* to `cleveref` like `theorem`, `algorithm`, `result`, etc. References made with `cleveref` will use this type. Note that using `label type` will result in compilation errors, if `cleveref` is not loaded. For an example, see Theorem 10.3.5 on page 250.

/tcb/no label type (no value, initially set)

Removes a `/tcb/label type`, if set before.

/tcb/step=⟨*counter*⟩ (no default, initially unset)

Shortcut for `phantom={\refstepcounter{#1}}`. The given ⟨*counter*⟩ is increased and ready for labelling. This option is not needed when using the convenient automated numbering introduced with version 2.40, see Subsection 4.1 from page 82.

/tcb/step and label={⟨*counter*⟩}{⟨*marker*⟩} (no default, initially unset)

Shortcut for using `/tcb/step` and `/tcb/label`. This option is not needed when using the convenient automated numbering introduced with version 2.40, see Subsection 4.1 from page 82.

`/tcb/list entry=`⟨*text*⟩ (no default, initially unset)

If the «list of tcolorbox(es)» feature described in Subsection 4.2 from page 87 is used, this key describes the ⟨*text*⟩ for an entry into the generated list, e. g.

```
list entry={\protect\numberline{\thetcbcounter}My beautiful Example}
```

See Section 9.8 from page 225 for a complete example.

N 2014-09-19 `/tcb/list text=`⟨*text*⟩ (style, no default)

This is a shortcut for setting `/tcb/list entry` to `\protect\numberline{\thetcbcounter}`⟨*text*⟩. So, the following settings are identical:

```
list text={My beautiful Example},
list entry={\protect\numberline{\thetcbcounter}My beautiful Example}
```

See Section 9.8 from page 225 for a complete example.

`/tcb/add to list={`⟨*list*⟩`}{`⟨*type*⟩`}` (no default, initially unset)

If the «list of tcolorbox(es)» feature described in Subsection 4.2 from page 87 is used, list entries are generated automatically. With this key, you can enforce an entry to the given ⟨*list*⟩ with the given ⟨*type*⟩. This issues:
`\addcontentsline{`⟨*list*⟩`}{`⟨*type*⟩`}{`⟨*entry text*⟩`}`

`/tcb/check odd page=true|false` (default `true`, initially `false`)

If set to `true`, the even/odd page testing from the package `changepage` is applied. The `/tcb/toggle enlargement` →P. 70 check and the `/tcb/toggle left and right` →P. 32 check will use the `\ifoddpage` macro from this package. This options is independent from `/tcb/phantom` →P. 79. Note that you have to include the package `changepage` by hand[3]:

```
\usepackage{changepage}
\strictpagecheck
```

The macro `\ifoddpage` can be used inside overlay or watermark code to test if the box is on an odd page. This will work also for boxes in a break sequence. Note that you cannot use the test inside the normal box content.

```
% \usepackage{changepage}
\tcbset{colframe=blue!75!black,colback=white}

\begin{tcolorbox}[enhanced,check odd page,
  watermark text={\ifoddpage Odd\else Even\fi\ page!}]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Even Page!

[3] If `changepage` is not included, a compilation error will arise.

3.21 Externalization

If the *externalization* library of the `tikz` package is used and `/tcb/graphical environment` → P. 89 is set to `tikzpicture`, a `tcolorbox` could trigger the externalization process which will arise a compilation error.

To avoid this, there are two possible strategies:

- Ensure, that `\tikzexternaldisable` is set before a `tcolorbox` is used. If you typically use the pattern `\tikzexternalenable` *some picture* `\tikzexternaldisable`, there is nothing to care about.
- If *externalization* is enabled globally, use `/tcb/shield externalize` to shield any `tcolorbox`. The preamble code could look like this:

```
\usetikzlibrary{external}
\tikzexternalize
\tcbset{shield externalize}
```

`/tcb/shield externalize=true|false` (default `true`, initially `false`)

If set to `true`, the drawing part of the `tcolorbox` is not being externalized which is a good thing at the current state of art. Nevertheless, if the `tcolorbox` contains a `tikzpicture`, this picture is still externalized. Pictures drawn with help of `/tcb/tikz upper` → P. 52 or alike are *not* externalized.

If a `tcolorbox` is used inside a node of an encircling `tikzpicture` which is externalized, do *not* use `\tikzexternaldisable` in front of the `tcolorbox`. `/tcb/shield externalize` is deactivated automatically inside a `tikzpicture`.

`/tcb/external=`⟨*file name*⟩ (no default, initially unset)

Convenience option which calls `\tikzsetnextfilename{`⟨*file name*⟩`}`. Typically, it may be used inside the option list of a `tcolorbox` to set the externalization ⟨*file name*⟩ for the first `tikzpicture` which is discovered *inside* the box content. The package `tikz` [20] or the library LIB skins has to be loaded to use this option. Additionally, `\usetikzlibrary{external}` has to be used.

`/tcb/remake=true|false` (default `true`, initially `false`)

Convenience option which calls `/tikz/external/remake next`. Typically, it may be used inside the option list of a `tcolorbox` to force the remake of the first `tikzpicture` which is discovered *inside* the box content. The package `tikz` [20] or the library LIB skins has to be loaded to use this option. Additionally, `\usetikzlibrary{external}` has to be used.

3.22 Miscellaneous

`/tcb/reset` (no value, initially set)

Sets (nearly) all `tcolorbox` settings (including loaded libraries) back to their default values *plus* any settings given by `\tcbsetforeverylayer` → P. 10. `/tcb/savedelimiter` → P. 19 and `/tcb/capture` → P. 76 keep their values. This option is useful for boxes in boxes where the inner box should not inherit the settings of the outer box. Note that for boxes inside boxes the `reset` is done automatically, if the standard settings of the package are used (v2.40 and above), see Section 3.15 from page 73.

4 Initialization Option Keys

The *initialization* options are only applicable for the generation of new environments and commands based on `tcolorbox` and friends. Particularly, they can be used for

- `\newtcolorbox` → P. 12,
- `\newtcbox` → P. 13,
- `\newtcblisting` → P. 203,
- `\newtcbinputlisting` → P. 205,
- `\newtcbtheorem` → P. 231, and
- `\newtcboxfit` → P. 274.

Typically, these options may generate counters and alike. It is **strongly** recommended that one use initialization options inside the preamble only. Otherwise, you may get trouble when using LaTeX's `\include` features.

4.1 Numbered Boxes

Counters assigned using the initialization options are administrated automatically. Especially, they are increased for each new box. Independent from the real counter name, the counter value can be referenced by `\thetcbcounter`, e. g. inside the title of the box. The real counter name is stored inside `\tcbcounter`.

`/tcb/new/auto counter` (no value, initially unset)

Creates a new counter automatically. With `/tcb/new/number format` → P. 84 and `/tcb/new/number within` → P. 84, the appearance and behavior of the counter can be changed. The counter value is referenced by `\thetcbcounter`.

Definition in the preamble:

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

```
\begin{pabox}[label={myautocounter}]{Title with number}
This box is automatically numbered with \ref{myautocounter} on page
\pageref{myautocounter}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{pabox}
```

Examp. 4.1: Title with number

This box is automatically numbered with 4.1 on page 82. Inside the box, the 4.1 can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@pabox`.

`/tcb/new/use counter from=`⟨*tcolorbox*⟩ (no default, initially unset)

Here, a counter from another ⟨*tcolorbox*⟩ is reused. Note that the settings for `/tcb/new/number format` → P. 84 and `/tcb/new/number within` → P. 84 are inherited and cannot be changed. The counter value is referenced by `\thetcbcounter`.

```
\newtcolorbox[use counter from=pabox]{mybox}[2][]{%
colback=blue!5!white,colframe=blue!75!black,fonttitle=\bfseries,
title=Some Box \thetcbcounter: #2,#1}

\begin{mybox}[label={myusecounterfrom}]{Title with continued number}
This box is automatically numbered with \ref{myusecounterfrom} on page
\pageref{myusecounterfrom}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{mybox}
```

Some Box 4.2: Title with continued number

This box is automatically numbered with 4.2 on page 83. Inside the box, the 4.2 can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@pabox`.

`/tcb/new/use counter=`⟨*counter*⟩ (no default, initially unset)

Here, an ordinary existing LaTeX ⟨*counter*⟩ is used for numbering. With `/tcb/new/number format` → P. 84 and `/tcb/new/number within` → P. 84, the appearance and behavior of the counter can be changed. The counter value is referenced by `\thetcbcounter`.

```
% \newcounter{myexample}%  preamble
\newtcolorbox[use counter=myexample,number format=\Alph]{mybox}[2][]{%
colback=green!5!white,colframe=green!55!black,fonttitle=\bfseries,
title=Some Box \thetcbcounter: #2,#1}

\begin{mybox}[label={myusecounter}]{Title with \LaTeX\ number}
This box is automatically numbered with \ref{myusecounter} on page
\pageref{myusecounter}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{mybox}
```

Some Box A: Title with LaTeX number

This box is automatically numbered with A on page 83. Inside the box, the A can also be referenced by `\thetcbcounter`. The real counter name is `myexample`.

N 2014-09-19 `/tcb/new/use counter*=`⟨*counter*⟩ (no default, initially unset)

An existing LaTeX ⟨*counter*⟩ is used for numbering. In contrast to `/tcb/new/use counter`, the options `/tcb/new/number format` → P. 84 and `/tcb/new/number within` → P. 84 are ignored. Use this for counters which are already configured outside the `tcolorbox` package, e. g. the standard `figure` counter.

`/tcb/new/no counter` (no value, initially set)

The created boxes are not numbered. This is the default. The option may be used to overrule a previous option.

`/tcb/new/number within`=⟨*counter*⟩ (no default, initially unset)

The automatic counter is set to zero, if ⟨*counter*⟩ is increased. Additionally, during output, the value of ⟨*counter*⟩ is prepended to the value of the automatic counter.

To prepend the automatic counter with the chapter number and to reset it with every new chapter, use:

```
number within=chapter
```

See `/tcb/new/use counter` → P. 83 for a complete example.

`/tcb/new/number format`=⟨*format macro*⟩ (no default, initially `\arabic`)

Declares the format of the automatic counter. The ⟨*format macro*⟩ can be any valid LaTeX number formatting macro like `\arabic`, `\roman`, etc.

To display the counter value in large roman numbers, use:

```
number format=\Roman
```

See `/tcb/new/auto counter` → P. 82 for a complete example.

`/tcb/new/number freestyle`=⟨*code*⟩ (no default, initially unset)

Allows advanced control over the complete number format. This option overrules the format given by `/tcb/new/number within` and `/tcb/new/number format`. Nevertheless, you can combine it with `/tcb/new/number within` to get the desired reset property.

The ⟨*code*⟩ is some formatting code which should contain `\tcbcounter` to reference the automated counter. Since this ⟨*code*⟩ is expanded, you have to secure each macro with `\noexpand` with *exception* of `\tcbcounter`.

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section,
  number freestyle={(Q/\noexpand\thesection/\noexpand\Alph{\tcbcounter})},
  ]{phbox}[2][]{%
  colback=yellow!15!white,colframe=blue!75!black,fonttitle=\bfseries,
  title=Question~\thetcbcounter: #2,#1}
```

```
\begin{phbox}[label={myfreestyle}]{Title with freestyle number}
This box is automatically numbered with \ref{myfreestyle} on page
\pageref{myfreestyle}. Inside the box, the \thetcbcounter\ can
also be referenced by |\thetcbcounter|.
The real counter name is \texttt{\tcbcounter}.
\end{phbox}
```

Question (Q/4/A): Title with freestyle number

This box is automatically numbered with (Q/4/A) on page 84. Inside the box, the (Q/4/A) can also be referenced by `\thetcbcounter`. The real counter name is `tcb@cnt@phbox`.

The following options /tcb/new/crefname and /tcb/new/Crefname need to be set inside the preamble.

/tcb/new/crefname={⟨*singular*⟩}{⟨*plural*⟩} (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. It creates a cross-reference type for the new `tcolorbox`'es, where the lowercase ⟨*singular*⟩ and ⟨*plural*⟩ forms of the cross-reference are given. See /tcb/label type → P. 79 and [5] for more information.

/tcb/new/Crefname={⟨*singular*⟩}{⟨*plural*⟩} (no default, initially unset)

This option key can be used only in conjunction with the `cleveref` package [5] which has to be loaded separately. It creates a cross-reference type for the new `tcolorbox`'es, where the uppercase ⟨*singular*⟩ and ⟨*plural*⟩ forms of the cross-reference are given. See /tcb/label type → P. 79 and [5] for more information.

Definition in the preamble:

```
% \usepackage{cleveref}
\newtcolorbox[auto counter,number within=section,
  Crefname={Bluebox}{Blueboxes},crefname={bluebox}{blueboxes}]%
  {mybluebox}[2][]{colback=blue!5!white,colframe=blue!75!black,fonttitle=\bfseries,
     title=Bluebox \thetcbcounter: #2,#1}
```

```
% \usepackage{cleveref}
% \usepackage{varioref}
\begin{mybluebox}[label={myreference}]{My title}
This is an example.
\end{mybluebox}

\Cref{myreference}, \cref{myreference}.\\
\Cpageref{myreference}, \cpageref{myreference}.\\
\nameCref{myreference}, \namecref{myreference}.\\
\labelcref{myreference}, \labelcpageref{myreference}.\\
With \texttt{varioref}:\\
\Vref{myreference}, \vref{myreference}.\\
\Vref*{myreference}, \vref*{myreference}.
```

Bluebox 4.1: My title

This is an example.

Bluebox 4.1, bluebox 4.1.
Page 85, page 85.
Bluebox, bluebox.
4.1, 85.
With `varioref`:
Bluebox 4.1, bluebox 4.1.
Bluebox 4.1, bluebox 4.1.

N 2014-09-19 **/tcb/new/blend into=⟨*name*⟩** (style, no default, initially unset)

Used to comfortably blend into an existing schema of naming and numbering for some selected cases. For example, a `tcolorbox` can be used to display and entitle an image pretending to be a standard `figure` environment. Here, `/tcb/title` → P. 14 is used instead of the standard `\caption` and `/tcb/list text` → P. 80 can be used instead of the optional parameter of the standard `\caption`.

Feasible values for ⟨*name*⟩ are:

- `figures`: blend into the standard `figure` environment.
- `tables`: blend into the standard `table` environment.
- `listings`: blend into the standard `lstlistings` environment of the package `listings` [6].

Note that `blend into=listings` can only be used in the document content or, preferably, inside a `\AtBeginDocument` clause! Using it without `\AtBeginDocument` inside the preamble does not work since the `listings` packages initializes its counter also inside `\AtBeginDocument`.

```
\begin{figure}[htb]
  \centering\includegraphics[height=4cm]{lichtspiel.jpg}
  \caption{A standard figure}
\end{figure}

\newtcolorbox[blend into=figures]{myfigure}[2][]{float=htb,capture=hbox,
  title={#2},every float=\centering,#1}

\begin{myfigure}{A tcolorbox figure}
  \includegraphics[height=4cm]{lichtspiel.jpg}
\end{myfigure}
```

Figure 1: A standard figure

Figure 2: A tcolorbox figure

4.2 Lists of tcolorboxes

For figures and tables, LaTeX provides the `\listoffigures` and `\listoftables` commands to create lists of these numbered entities. Also, a `tcolorbox` can be part of such a kind of list.

1. Assign a list ⟨*name*⟩ by the *initialization* option `/tcb/new/list inside`.
2. Optionally, a new ⟨*type*⟩ for list entries may be assigned by the *initialization* option `/tcb/new/list type`.
3. List entries a generated automatically within each new `tcolorbox` using the above initialization.
   - If `/tcb/list entry` → P. 80 is set, the entry is generated with it.
   - Otherwise, if `/tcb/title` → P. 14 is set, the entry is generated with it.
   - Otherwise, the entry is generated with the current number and the environment name.
4. The generated list is displayed by `\tcblistof`.

`/tcb/new/list inside`=⟨*name*⟩ (no default, initially unset)

Assigns a list or contents file to the generated `tcolorbox`es. Entries to this list are saved to a file which gets the ⟨*name*⟩ as file name extension. The list is referenced by this name in `\tcblistof`. For example:

```
list inside=exam
```

See Section 9.8 from page 225 for a complete example.

`/tcb/new/list type`=⟨*type*⟩ (no default, initially `tcolorbox`)

Optionally, some ⟨*type*⟩ can be assigned to the list entries. For a new ⟨*type*⟩, a macro `\l@`⟨*type*⟩ has to exist which controls the format of the list entry. The default type is defined by

```
\newcommand*\l@tcolorbox{\@dottedtocline{1}{1.5em}{2.3em}}
```

This is identical to the `\l@section` setting of LaTeX. `\l@tcolorbox` can be redefined or a new ⟨*type*⟩ can be assigned.

`\tcblistof``[`⟨*macro*⟩`]{`⟨*name*⟩`}{`⟨*title text*⟩`}`

Displays the generated list of `tcolorbox`es with the given ⟨*name*⟩. The heading is generated by ⟨*macro*⟩`{`⟨*title text*⟩`}` where `\section` is the default setting for ⟨*macro*⟩.
To display the list inside a subsection, use for example:

```
\tcblistof[\subsection]{exam}{List of Exercises}
```

The result of the example is found as Subsection 9.9 on page 228.
The core of the list is generated by `\@starttoc{`⟨*name*⟩`}` which can be wrapped into an own macro.

5 Saving and Loading of Verbatim Texts

The following macros are slightly modified versions of the original macros from the known packages `moreverb` and `verbatim`. They are used implicitly inside of a `tcolorbox` environment, but they can be used outside also.

`\begin{tcbverbatimwrite}{`⟨*file name*⟩`}`
⟨*environment content*⟩
`\end{tcbverbatimwrite}`

Saves the ⟨*environment content*⟩ to a file named by ⟨*file name*⟩. TEX macros inside the environment are not expanded.

```
\begin{tcbverbatimwrite}{\jobname_verbexp.tex}
  This text is saved \textit{as is}.
\end{tcbverbatimwrite}

Now, we are using the file:\par
\input{\jobname_verbexp.tex}
```

Now, we are using the file:
This text is saved *as is*.

This environment may be used inside an own environment. Note, that inside the environment definition `\tcbverbatimwrite` has to be used instead of `\begin{tcbverbatimwrite}` and `\endtcbverbatimwrite` instead of `\end{tcbverbatimwrite}`.

```
\newenvironment{myverbatim}{%
  \begingroup\tcbverbatimwrite{\jobname_myverb.tex}}%
  {\endtcbverbatimwrite\endgroup}

\begin{myverbatim}
  This is the text which is saved by my own environment.
\end{myverbatim}

Now, we are using the file:\par
\input{\jobname_myverb.tex}
```

Now, we are using the file:
This is the text which is saved by my own environment.

`\begin{tcbwritetemp}`
⟨*environment content*⟩
`\end{tcbwritetemp}`

Has the same function as `tcbverbatimwrite`, but uses the key value of `tempfile` for the file name.

```
\begin{tcbwritetemp}
  This text is saved \textit{as is}.
\end{tcbwritetemp}

Now, we are using the file:\par
\tcbusetemp
```

Now, we are using the file:
This text is saved *as is*.

`\tcbusetemp`

Loads the current temporary file which was saved by `tcbwritetemp`.

6 Technical Overview and Customization

This section provides a technical overview of the skin concept of `tcolorbox`. For most applications of `tcolorbox`, one will not need to know the bells and whistles described herein. You may proceed to Section 7 on page 101 where the customization options for most users are documented.

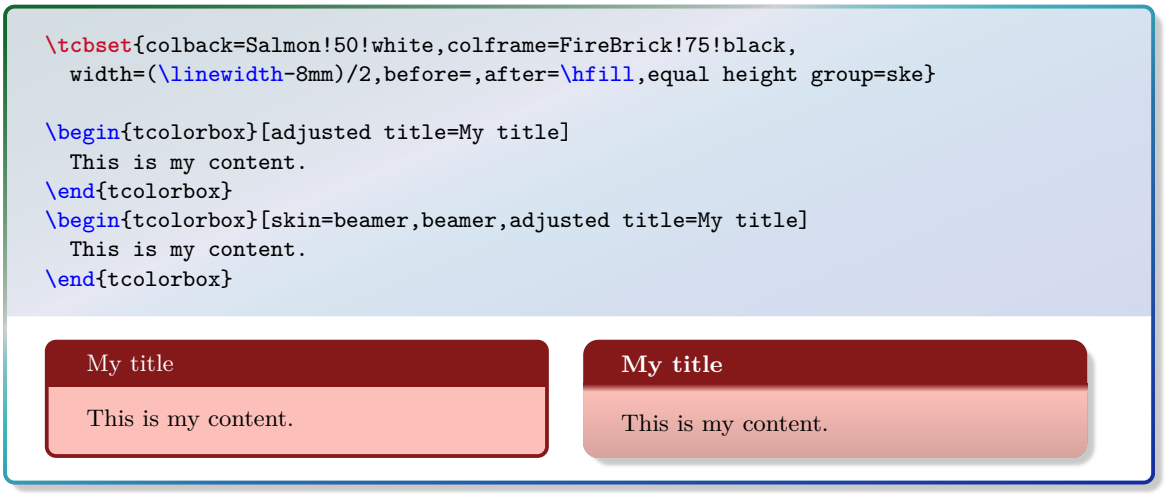
The following explanations also cover options and settings from the LIB skins library, see Section 7 on page 101.

6.1 Skins and Drawing Engines

From a technical point of view, a *skin* is a style definition for the appearance of a `tcolorbox`. The core package provides some additional option keys for skins but only two skins called `standard` →P. 148 and `standard jigsaw` →P. 149. The LIB skins library adds several more skins. To change to a skin, only one option from the core package has to be set.

`/tcb/skin`=⟨*name*⟩ (style, no default, initially `standard`)
Sets the current skin to ⟨*name*⟩. This is a style definition which sets all the following keys, i. e. for many use cases there is nothing more to do.

```
\tcbset{colback=Salmon!50!white,colframe=FireBrick!75!black,
  width=(\linewidth-8mm)/2,before=,after=\hfill,equal height group=ske}

\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}
\begin{tcolorbox}[skin=beamer,beamer,adjusted title=My title]
  This is my content.
\end{tcolorbox}
```



`/tcb/skin first`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the skin for the *first* part of the break sequence is set to ⟨*name*⟩, see Subsection 11.6 on page 264. Typically, this key is set by a `/tcb/skin`.

`/tcb/skin middle`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the skin for the *middle* parts (if any) of the break sequence is set to ⟨*name*⟩, see Subsection 11.6 on page 264. Typically, this key is set by a `/tcb/skin`.

`/tcb/skin last`=⟨*name*⟩ (style, no default, initially `standard`)
If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the skin for the *last* part of the break sequence is set to ⟨*name*⟩, see Subsection 11.6 on page 264. Typically, this key is set by a `/tcb/skin`.

`/tcb/graphical environment`=⟨*name*⟩ (no default, initially `pgfpicture`)
Sets the graphical environment for the `tcolorbox` to ⟨*name*⟩. Feasible values are `pgfpicture` and `tikzpicture` or environments which inherit from one of these two. This key is set by a `/tcb/skin` and may seldom be used directly.

The skin of a `tcolorbox` is drawn by up to four *engines*. Afterwards, the text content is drawn which is not part of a skin. The four steps are:

1. The *frame* of the box, drawn by `/tcb/frame engine`.
2. The *interior* of the box. The interior of a box with title is drawn differently from a box without title. `/tcb/interior titled engine` or `/tcb/interior engine` → P. 91 is used to draw the interior.
3. The *segmentation* (line) of the box, if there is a lower part; drawn by `/tcb/segmentation engine` → P. 91.
4. The *title area* of the box, if there is a title and `/tcb/title filled` → P. 20 is set to `true`; drawn by `/tcb/title engine` → P. 91.

`/tcb/frame engine`=⟨*name*⟩ (no default, initially `standard`)

Sets the *frame* drawing engine for a box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 89. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathfirstjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathmiddlejigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathlast`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `pathlastjigsaw`: a `tikz` path which is controlled by `/tcb/frame style` → P. 101,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/interior titled engine`=⟨*name*⟩ (no default, initially `standard`)

Sets the *interior* drawing engine for a titled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 89. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathlast`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/interior engine=`⟨*name*⟩ (no default, initially `standard`)

Sets the *interior* drawing engine for an untitled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 89. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `pathlast`: a `tikz` path which is controlled by `/tcb/interior style` → P. 102,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/segmentation engine=`⟨*name*⟩ (no default, initially `standard`)

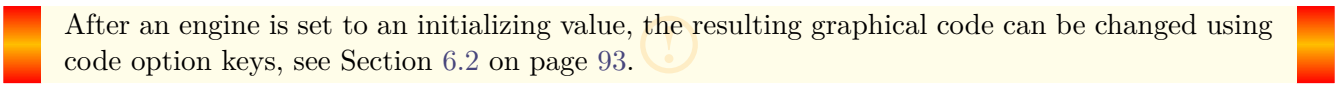
Sets the *segmentation* (line) drawing engine for a box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 89. Feasible values for ⟨*name*⟩ are:

- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/segmentation style` → P. 104,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.

`/tcb/title engine=`⟨*name*⟩ (no default, initially `standard`)

Sets the *title area* drawing engine for a titled box to ⟨*name*⟩. Typically, this key is set by a `/tcb/skin` → P. 89. Feasible values for ⟨*name*⟩ are:

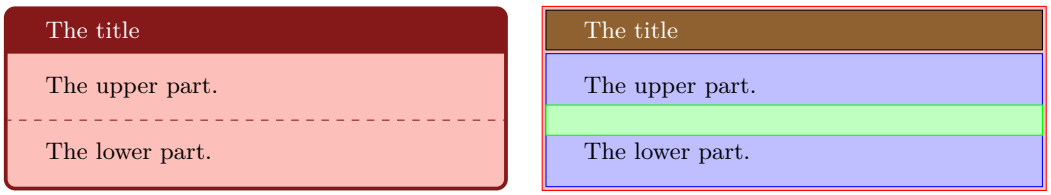
- `standard`: the original code from the core package,
- `path`: a `tikz` path which is controlled by `/tcb/title style` → P. 104,
- `pathfirst`: a `tikz` path which is controlled by `/tcb/title style` → P. 104,
- `pathmiddle`: a `tikz` path which is controlled by `/tcb/title style` → P. 104,
- `pathlast`: a `tikz` path which is controlled by `/tcb/title style` → P. 104,
- `freelance`: deprecated.
- `spartan`: a quite spartan code.
- `empty`: draw nothing.



`/tcb/geometry nodes=true|false` (default `true`, initially `false`)

If set to `true`, up to four `tikz` nodes are defined for a `tcolorbox` which are named `frame`, `interior`, `segmentation`, and `title`. These nodes describe the boundaries of the equally named parts of a `tcolorbox`. They are used by most engines based on Ti*k*Z. Typically, this key is set automatically by a `/tcb/skin` → P. 89.

```
\tcbset{colback=Salmon!50!white,colframe=FireBrick!75!black,
  width=(\linewidth-8mm)/2,before=,after=\hfill,equal height group=geon}

\begin{tcolorbox}[adjusted title=The title]
  The upper part.\tcblower The lower part.
\end{tcolorbox}
\begin{tcolorbox}[enhanced,adjusted title=The title,
  frame code={\path[draw=red,fill=red!25]
      (frame.south west) rectangle (frame.north east);},
  interior titled code={\path[draw=blue,fill=blue!25]
      (interior.south west) rectangle (interior.north east);},
  segmentation code={\path[draw=green,fill=green!25]
      (segmentation.south west) rectangle (segmentation.north east);},
  title code={\path[draw=black,fill=brown!75!black]
      (title.south west) rectangle (title.north east);}]
  The upper part.\tcblower The lower part.
\end{tcolorbox}
```



6.2 Code Option Keys

The following code options are applicable for all skins. The used ⟨*graphical code*⟩ can be any `pgf` code. For all skins with exception of `standard` → P. 148 and `standard jigsaw` → P. 149, the ⟨*graphical code*⟩ can also be any Ti*k*Z code.

`/tcb/frame code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *frame* of the box.

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

\begin{tcolorbox}[enhanced,frame code={
  \foreach \n in {north east,north west,south east,south west}
  {\path [fill=red!75!black] (interior.\n) circle (3mm); }; }]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is the lower part.

`/tcb/frame empty` (style, no value)

This is a shortcut for setting `/tcb/frame code` to empty. This option removes the drawing of the frame. Alternatively, use `/tcb/frame hidden` → P. 102.

`/tcb/interior titled code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *interior* of the box, if the box comes with a title.

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

\begin{tcolorbox}[enhanced,title=My title,interior titled code={
  \path[draw=red!5!white,line width=5mm,line cap=round]
    ([xshift=3mm,yshift=-3mm]interior.north west)
    --([xshift=-3mm,yshift=3mm]interior.south east)
    ([xshift=3mm,yshift=3mm]interior.south west)
    --([xshift=-3mm,yshift=-3mm]interior.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

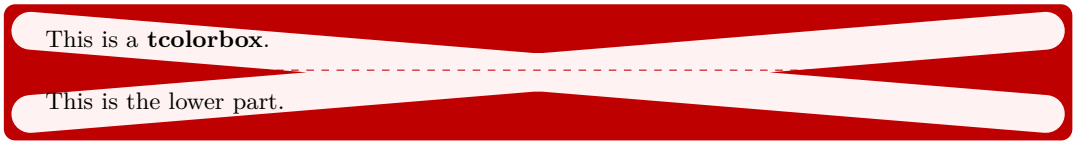
`/tcb/interior titled empty` (style, no value)

This is a shortcut for setting `/tcb/interior titled code` to empty. This option removes the drawing of the untitled interior. Alternatively, use `/tcb/interior hidden` → P. 103.

/tcb/interior code=⟨*graphical code*⟩ (code, default from **standard**)

The given ⟨*graphical code*⟩ is used for drawing the *interior* of the box, if the box is without a title.

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

\begin{tcolorbox}[enhanced,interior code={
  \path[draw=red!5!white,line width=5mm,line cap=round]
    ([xshift=3mm,yshift=-3mm]interior.north west)
    --([xshift=-3mm,yshift=3mm]interior.south east)
    ([xshift=3mm,yshift=3mm]interior.south west)
    --([xshift=-3mm,yshift=-3mm]interior.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



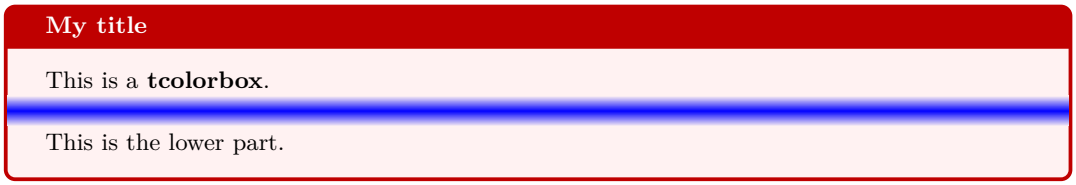
/tcb/interior empty (style, no value)

This is a shortcut for setting /tcb/interior code to empty. This option removes the drawing of the interior. Alternatively, use /tcb/interior hidden → P. 103.

/tcb/segmentation code=⟨*graphical code*⟩ (code, default from **standard**)

The given ⟨*graphical code*⟩ is used for drawing the *segmentation* area of the box.

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

\begin{tcolorbox}[enhanced,title=My title,segmentation code={
  \path[top color=red!5!white,bottom color=red!5!white,middle color=blue]
  (segmentation.south west) rectangle (segmentation.north east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tcb/segmentation empty (style, no value)

This is a shortcut for setting /tcb/segmentation code to empty. This option removes the drawing of the segmentation line. Alternatively, use /tcb/segmentation hidden → P. 104.

`/tcb/title code=`⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *title* area of the box.

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

\begin{tcolorbox}[enhanced,title=My title,title code={
  \path[draw=yellow,solid,decorate,line width=2mm,
    decoration={coil,aspect=0,segment length=10.1mm}]
    ([xshift=1mm]title.west) -- ([xshift=-1mm]title.east);}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

This is the lower part.

`/tcb/title empty` (style, no value)

This is a shortcut for setting `/tcb/title code` to empty. This option removes the drawing of the title area. Alternatively, use `/tcb/title hidden` → P. 105.

6.3 Subskins

A subskin is a new /tcb/skin → P. 89 based on an existing skin which is extended or changed.

Never use geometry settings or bounding box options inside a subskin definition! If one skin is replaced by another skin, the overall bounding box should stay constant. Especially, if a skin is used for a breakable box, unpredictable and unpleasant results could arise otherwise. If you want to change the geometry also, use an additional style. See the skin beamer → P. 168 and the style /tcb/beamer → P. 168 as pattern.

\tcbsubskin{⟨*name*⟩}{⟨*base skin*⟩}{⟨*options*⟩}

Creates a new skin ⟨*name*⟩ which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. The new skin ⟨*name*⟩ can be used as value for the keys /tcb/skin → P. 89, /tcb/skin first → P. 89, /tcb/skin middle → P. 89, and /tcb/skin last → P. 89. As ⟨*base skin*⟩, one can take standard → P. 148, empty → P. 177, enhanced → P. 150, or any skin from the LIB skins library, see Section 7 on page 101.

```
% \tcbuselibrary{skins}
\tcbsubskin{mycross}{empty}{frame code={%
  \draw[red,line width=5pt] (frame.south west)--(frame.north east);
  \draw[red,line width=5pt] (frame.north west)--(frame.south east);},
  skin first=mycross,skin middle=mycross,skin last=mycross }

\begin{tcolorbox}[skin=mycross]
  This is my content.
\end{tcolorbox}
```

/tcb/skin first is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin first → P. 89.
See a detailed example on page 182.

/tcb/skin middle is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin middle → P. 89.
See a detailed example on page 182.

/tcb/skin last is subskin of={⟨*base skin*⟩}{⟨*options*⟩} (no default, initially unset)

Creates a new unnamed skin which inherits all properties of an existing ⟨*base skin*⟩ plus the given ⟨*options*⟩. This skin is set as /tcb/skin last → P. 89.
See a detailed example on page 182.

6.4 Drawing Scheme

Depending on the complexity of a `tcolorbox` definition, the resulting box is drawn in a more or less complex series of steps.

To document and demonstrate these drawing steps, we consider the following box definition:

```
\newtcolorbox{testbox}[1][]{enhanced,title=Test Box,
  boxrule=1mm,titlerule=0.5mm,colframe=blue!50!black,
  interior style={top color=blue!20!green!50!white,bottom color=blue!20!yellow!50!white},
  colbacktitle=blue!50!green!90!white,segmentation style={solid},
  fonttitle=\bfseries,drop fuzzy shadow,borderline={0.3mm}{0.35mm}{yellow!50!white},
  underlay={\path[fill image opacity=0.15,fill image scale=0.9,
    fill stretch picture={\draw[blue,line width=2mm] circle (1);}]
    (interior.south west) rectangle (interior.north east);},
  watermark text={Watermark},watermark color={green!20!white},
  finish={\begin{tcbclipframe}
    \path[bottom color=black,top color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.south east) -- (frame.north east) -- cycle;
    \path[top color=white,bottom color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.north east) -- (frame.north west) -- cycle;
    \end{tcbclipframe}},#1}
```

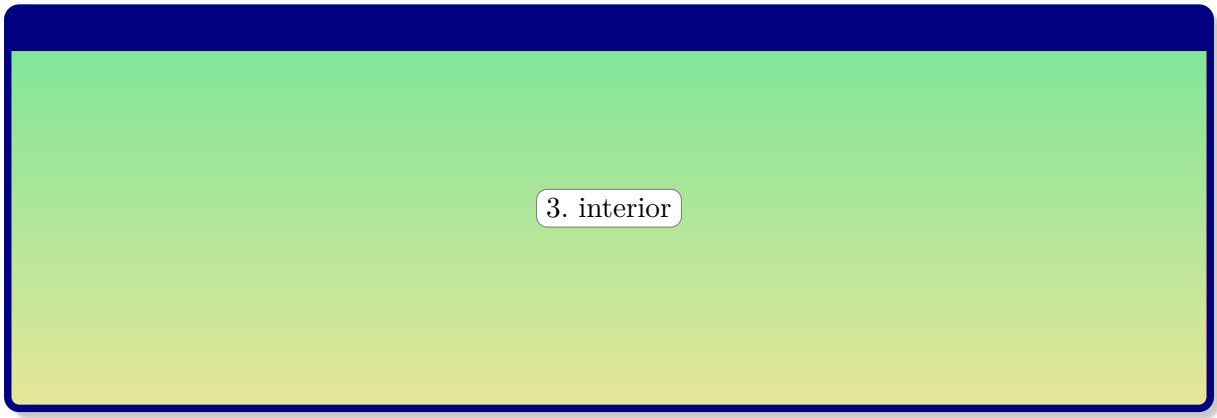
For this definition, we get the maximal number of drawing steps:

1. shadow

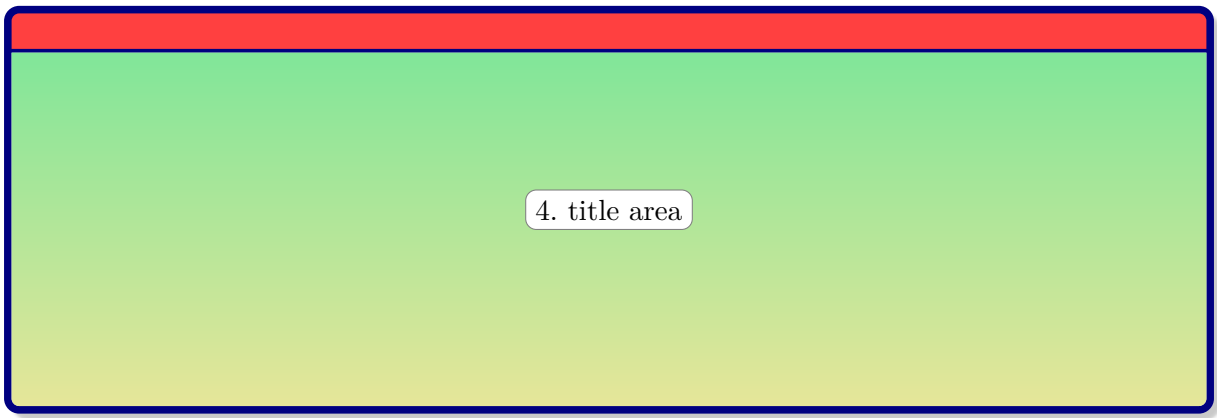
- Section 7.6 on page 129

2. frame

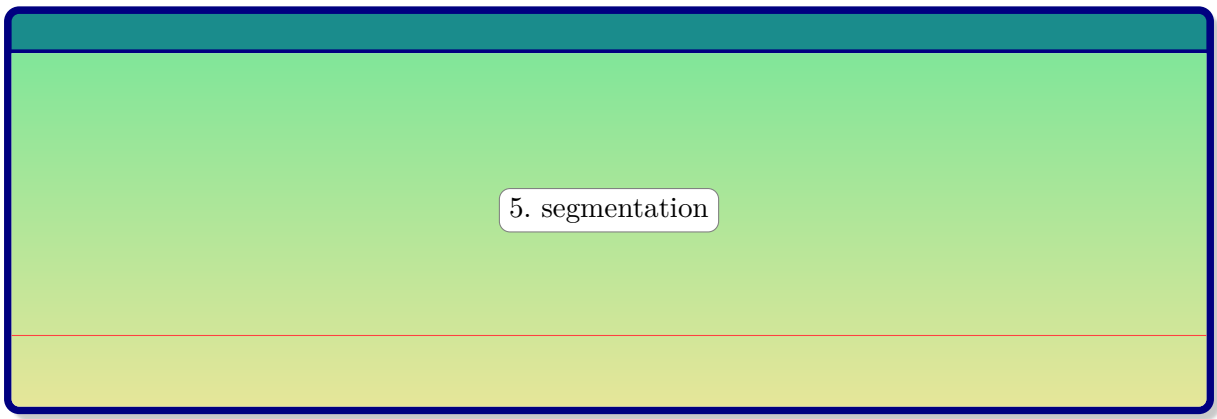
- `/tcb/colframe` → P. 20, `/tcb/opacityframe` → P. 37
- `/tcb/frame code` → P. 93
- `/tcb/frame style` → P. 101



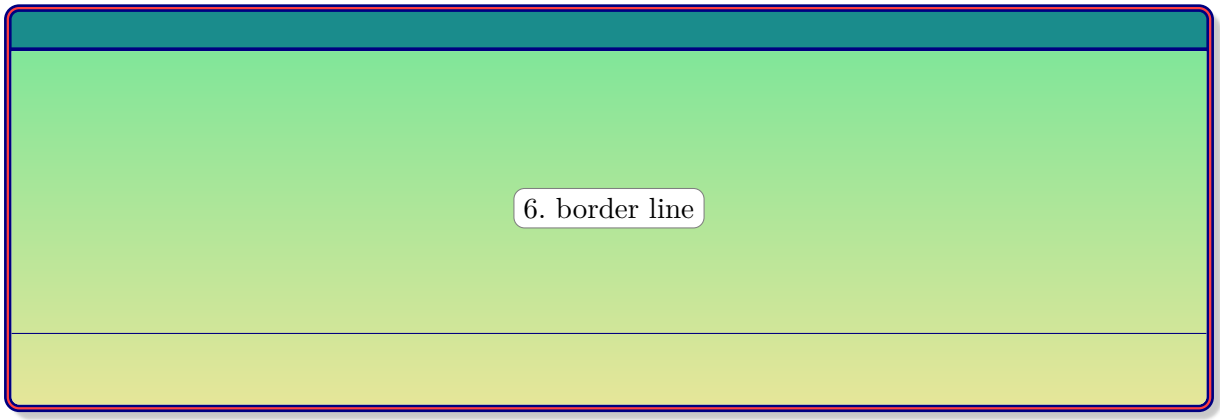
- `/tcb/colback` → P. 20, `/tcb/opacityback` → P. 37
- `/tcb/interior code` → P. 94, `/tcb/interior titled code` → P. 93
- `/tcb/interior style` → P. 102



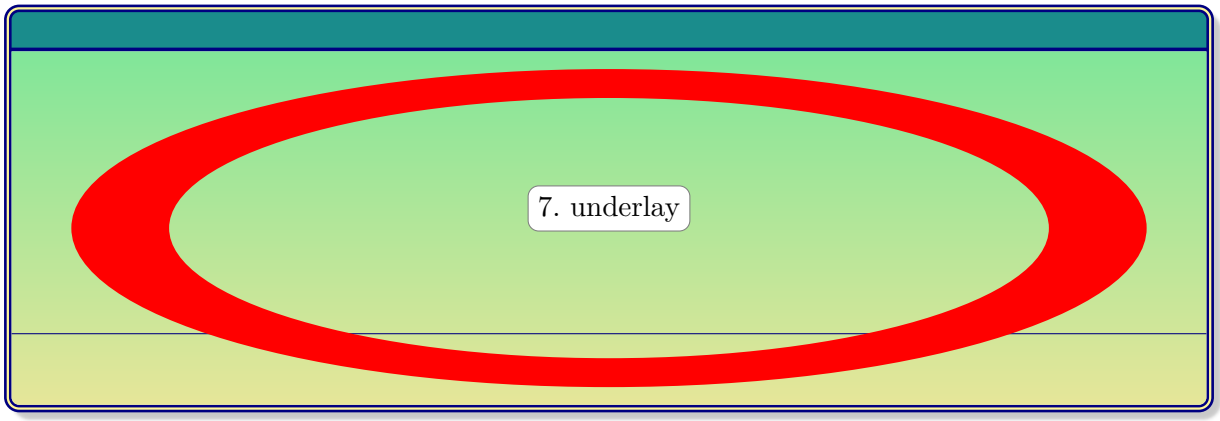
- `/tcb/colbacktitle` → P. 20, `/tcb/opacitybacktitle` → P. 37
- `/tcb/title code` → P. 95
- `/tcb/title style` → P. 104



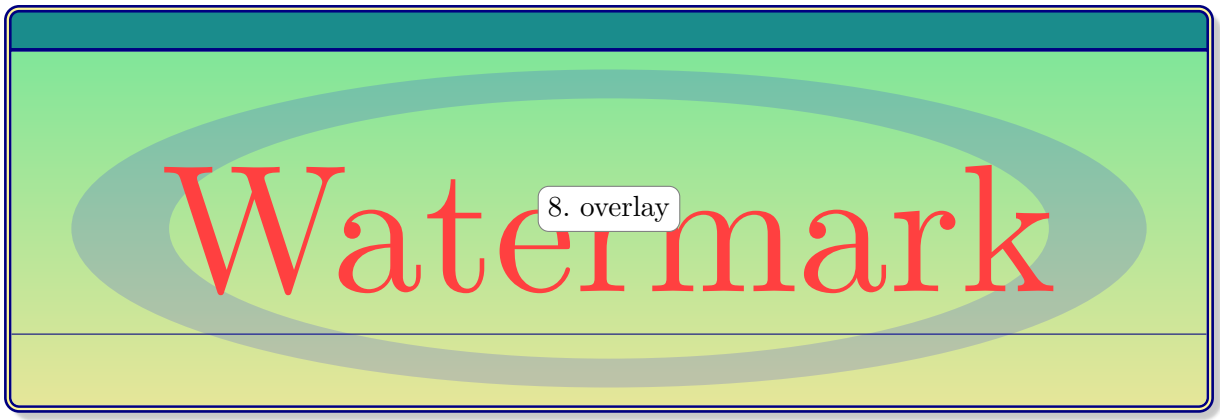
- `/tcb/lower separated` → P. 18
- `/tcb/segmentation code` → P. 94
- `/tcb/segmentation style` → P. 104



- Section 7.5 on page 126



- Section 7.2 on page 107
- Section 7.8 on page 141



- Section 3.10 on page 55
- Section 7.3 on page 114

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

9. text content

Lower part

- `/tcb/colupper` → P. 21, `/tcb/collower` → P. 21, `/tcb/coltitle` → P. 21
- `/tcb/fontupper` → P. 22, `/tcb/fontlower` → P. 22, `/tcb/fonttitle` → P. 22
- `/tcb/opacityupper` → P. 38, `/tcb/opacitylower` → P. 38, `/tcb/opacitytitle` → P. 38

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

10. finish

Lower part

- Section 7.9 on page 143

All together, the box is drawn:

```
% \usepackage{lipsum}
\begin{testbox}
\lipsum[2]
\tcblower
Lower part
\end{testbox}
```

Test Box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lower part

7 Library LIB skins

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

This also loads the package `tikz` [20]. Typically but not necessarily, the following skins use `tikz` instead of `pgf`.

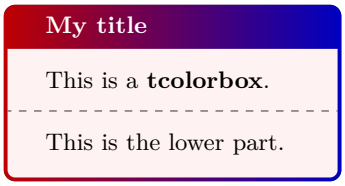
7.1 Style Option Keys

The following style options are applicable for all skins which use engines of type `path`, `pathfirst`, `pathmiddle`, or `pathlast`. Especially, the skin `enhanced` → P. 150 supports *all* of them and `standard` → P. 148 *none*.

`/tcb/frame style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *frame* of the box. This option is available if the `/tcb/frame engine` → P. 90 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

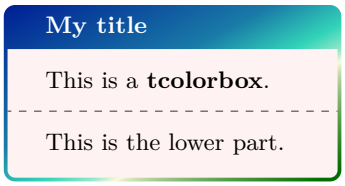
```
\tcbset{colback=red!5!white,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style={left color=red!75!black,
               right color=blue!75!black}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame style image=`⟨*file name*⟩ (no default, initially unset)

Fills the frame with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/frame style` together with `/tikz/fill stretch image` → P. 191.

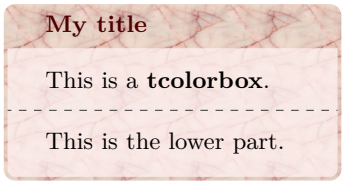
```
\tcbset{colback=red!5!white,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style image=blueshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame style tile={⟨graphics options⟩}{⟨file name⟩}` (no default, initially unset)

Fills the frame with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/frame style` → P. 101 together with `/tikz/fill tile image` → P. 195.

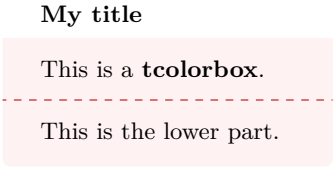
```
\tcbset{colback=red!5!white,coltitle=red!30!black,
  opacityback=0.75,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  frame style tile={width=1cm}{pink_marble.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/frame hidden` (style, no value)

This is a shortcut for `frame style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the frame. Alternatively, use `/tcb/frame empty` → P. 93.

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

\begin{tcolorbox}[enhanced,title=My title,
  frame hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

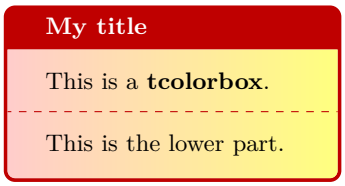


`/tcb/interior style=⟨tikz keys⟩` (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *interior* of the box. They are used for the titled and for the untitled version as well.

This option is available if the `/tcb/interior titled engine` → P. 90 or `/tcb/interior engine` → P. 91 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

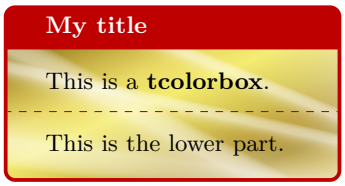
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style={left color=red!20!white,
                  right color=yellow!50!white}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior style image=`⟨*file name*⟩ (no default, initially unset)

Fills the interior with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/interior style` → P. 102 together with `/tikz/fill stretch image` → P. 191.

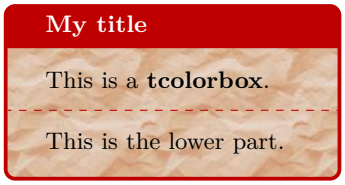
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style image=goldshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior style tile={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the interior with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/interior style` → P. 102 together with `/tikz/fill tile image` → P. 195.

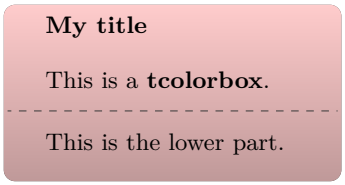
```
\tcbset{colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  interior style tile={width=2cm}{crinklepaper.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/interior hidden` (style, no value)

This is a shortcut for `interior style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the interior. Alternatively, use `/tcb/interior empty` → P. 94 and/or `/tcb/interior titled empty` → P. 93.

```
\tcbset{frame style={top color=red!20!white,
  bottom color=red!20!white!75!black},
  fonttitle=\bfseries,coltitle=black}

\begin{tcolorbox}[enhanced,title=My title,
  interior hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

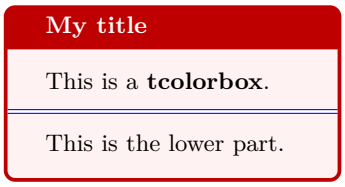


`/tcb/segmentation style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *segmentation* line of the box.

This option is available if the `/tcb/segmentation engine` → P. 91 is set to `path`. It is *not* available for `standard`.

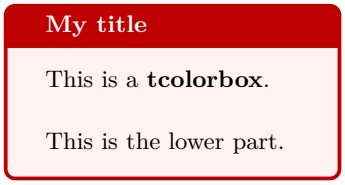
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  segmentation style={double=white,draw=blue,
                  double distance=1pt,solid}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/segmentation hidden` (style, no value)

This is a shortcut for `segmentation style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the segmentation line. See also `/tcb/lower separated` → P. 18 which has the same effect for most skins. Alternatively, use `/tcb/segmentation empty` → P. 94.

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

\begin{tcolorbox}[title=My title,
  enhanced,segmentation hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

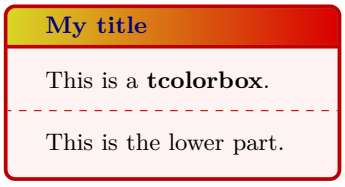


`/tcb/title style=`⟨`tikz` *keys*⟩ (style, no default)

The ⟨`tikz` *keys*⟩ are used inside the `tikz` path command for drawing the *title area* of the box.

This option is available if the `/tcb/title engine` → P. 91 is set to `path`, `pathfirst`, `pathmiddle`, or `pathlast`. It is *not* available for `standard`.

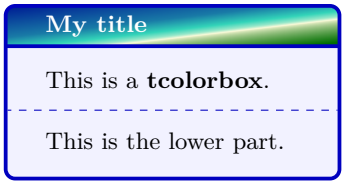
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  coltitle=blue!50!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style={left color=blue!15!yellow,
               right color=red!85!black}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title style image=`⟨*file name*⟩ (no default, initially unset)

Fills the title area with an external image referenced by ⟨*file name*⟩. For advanced features like blending of a picture with the background, use `/tcb/title style` → P. 104 together with `/tikz/fill stretch image` → P. 191.

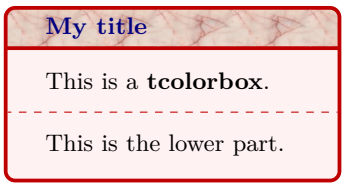
```
\tcbset{colback=blue!5!white,colframe=blue!75!black,
  fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style image=blueshade.png]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title style tile={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the title area with a tile pattern based on an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. For advanced features like blending of a picture with the background, use `/tcb/title style` → P. 104 together with `/tikz/fill tile image` → P. 195.

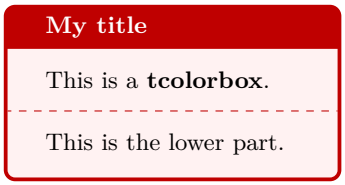
```
\tcbset{colback=red!5!white,colframe=red!75!black,
  coltitle=blue!50!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced,title=My title,
  title style tile={width=1cm}{pink_marble.png}]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



`/tcb/title hidden` (style, no value)

This is a shortcut for `title style={draw=none,fill=none}`. Depending on the skin, this option switches off the drawing of the title background. See also `/tcb/title filled` → P. 20 for a similar effect. Alternatively, use `/tcb/title empty` → P. 95.

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

\begin{tcolorbox}[title=My title,
  enhanced,title hidden]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```

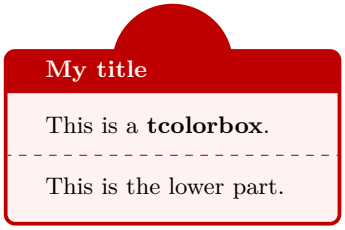


The combined Ti*k*Z style applied to frame, interior, and title background can used by authors in customizing code.

/tikz/tcb fill frame (style, no value)

This is a Ti*k*Z style which is finally applied to the *frame* of the box.

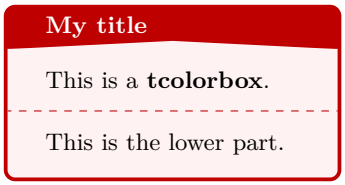
```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  frame code app={\path[tcb fill frame]
    ([yshift=-2mm]frame.north)
        circle (8mm); }  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tikz/tcb fill interior (style, no value)

This is a Ti*k*Z style which is finally applied to the *interior* of the box.

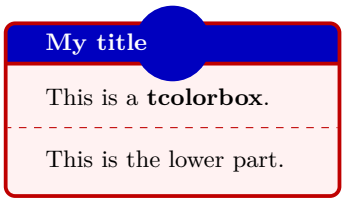
```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  interior titled code app={\path[tcb fill interior]
      ([yshift=-0.1pt]interior.north east)
    --([yshift=3pt]interior.north)
    --([yshift=-0.1pt]interior.north west)
    --cycle;}  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



/tikz/tcb fill title (style, no value)

This is a Ti*k*Z style which is finally applied to the *title area* of the box.

```
% \tcbuselibrary{hooks}
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  colbacktitle=blue!75!black,
  title code app={\path[tcb fill title]
    (title) circle (5mm); }  }

\begin{tcolorbox}[title=My title]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part.
\end{tcolorbox}
```



7.2 Boxed Title Option Keys

The following options place the title text into an own \tcbox→ P. 11. This boxed title can be customized independently from the main box using /tcb/boxed title style→ P. 109. The placement can be influenced by ⟨*boxtitle options*⟩.

/tcb/attach boxed title to top left=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the top left corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/attach boxed title to top center=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the top of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/attach boxed title to top right=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the top right corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top right]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

/tcb/attach boxed title to bottom left=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the bottom left corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom left]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

My title

/tcb/attach boxed title to bottom center=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the bottom of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom center]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

My title

/tcb/attach boxed title to bottom right=⟨*boxtitle options*⟩ (style, default empty)

The title is boxed with a \tcbox→ P. 11 and attached to the bottom right corner of the main box.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to bottom right]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

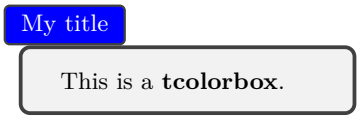
My title

The ⟨*boxtitle options*⟩ of the keys described above are shift values. The dimensions of the boxed title are stored into two macros **\tcboxedtitleheight** and **\tcboxedtitlewidth**. These macros can be used inside the following ⟨*boxtitle options*⟩:

/tcb/boxtitle/xshift=⟨*length*⟩ (no default, initially 0pt)

The boxed title is shifted by ⟨*length*⟩ in the horizontal direction.

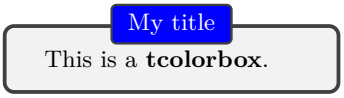
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left={xshift=-2mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/boxtitle/yshift=⟨*length*⟩ (no default, initially 0pt)

The boxed title is shifted by ⟨*length*⟩ in the vertical direction.

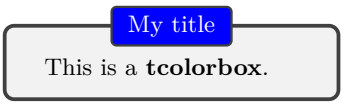
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center=
    {yshift=-\tcboxedtitleheight/2},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/boxtitle/yshifttext=⟨*length*⟩ (no default, initially 0pt)

The text inside the main box by ⟨*length*⟩ to give room for e. g. a sunken title.

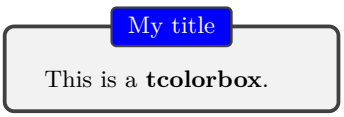
```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



/tcb/boxtitle/yshift*=⟨*length*⟩ (no default, initially 0pt)

Sets /tcb/boxtitle/yshift and /tcb/boxtitle/yshifttext the same time. /tcb/boxtitle/yshifttext is only set if necessary.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top center={yshift*=-3mm},
  boxed title style={size=small,colback=blue}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



The bounding box of the resulting total **tcolorbox** is adapted automatically to the *vertical* dimensions of the boxed title. Possible horizontal enlargements are *not* automatically computed.

```
\begin{tcolorbox}[enhanced,title=My title,
  attach boxed title to top left=
    {xshift=-2mm,yshift=-2mm},
  boxed title style={size=small,colback=blue},
  show bounding box]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

The boxed title options are implemented as an underlay, see Section 7.8 on page 141. Therefore, a boxed title is not drawn, if a skin does not support underlays like `standard`→P. 148. Still, the room for the boxed titles gets reserved in these cases.

A TikZ node `title` is produced by a boxed title which can be used inside `/tcb/frame code`→P. 93, `/tcb/interior code`→P. 94, underlays, overlays, and finishes.

A boxed title is almost always the first underlay. The only exceptions are underlays defined by `/tcb/underlay boxed title`→P. 142 which are drawn before. Additionally, underlays defined by `/tcb/underlay boxed title`→P. 142 are only drawn, if a boxed title is actually set. They are ignored, if there is no boxed title.

`/tcb/boxed title style`=⟨*options*⟩ (style, initially empty)

By default, a boxed title is dimensioned with `/tcb/size`→P. 30=`title` and inherits `/tcb/colframe`→P. 20 of the main box. Also, the `/tcb/colback`→P. 20 is inherited from the main `/tcb/colbacktitle`→P. 20. Font and color of the title text are set as usual. All other ⟨*options*⟩ are set by the `boxed title style` key. Since a boxed title is set by `\tcbox`→P. 11, all `tcolorbox` options are applicable here.

```
\begin{tcolorbox}[enhanced,title=My title,
  fonttitle=\bfseries,coltitle=green!25!black,
  attach boxed title to top center=
    {yshift=-2mm,yshifttext=-1mm},
  boxed title style={colframe=green!75!black,
    colback=yellow!50!green}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

```
\begin{tcolorbox}[enhanced,title=My title,
  colframe=red!50!black,colback=red!10!white,
  arc=1mm,colbacktitle=red!10!white,
  fonttitle=\bfseries,coltitle=red!50!black,
  attach boxed title to top left=
    {xshift=3.2mm,yshift=-0.50mm},
  boxed title style={enhanced,
    skin=enhancedfirst jigsaw,
    size=small,arc=1mm,bottom=-1mm,
    interior style={top color=red!30!white,
      bottom color=red!20!white}}]
  This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

My title

This is a **tcolorbox**.

```
\begin{tcolorbox}[enhanced,title=My title,
  colframe=blue!50!black,colback=blue!10!white,colbacktitle=blue!5!yellow!10!white,
  fonttitle=\bfseries,coltitle=black,attach boxed title to top center=
    {yshift=-0.25mm-\tcboxedtitleheight/2,yshifttext=2mm-\tcboxedtitleheight/2},
  boxed title style={enhanced,boxrule=0.5mm,
    frame code={ \path[tcb fill frame] ([xshift=-4mm]frame.west)
    -- (frame.north west) -- (frame.north east) -- ([xshift=4mm]frame.east)
    -- (frame.south east) -- (frame.south west) -- cycle; },
    interior code={ \path[tcb fill interior] ([xshift=-2mm]interior.west)
    -- (interior.north west) -- (interior.north east)
    -- ([xshift=2mm]interior.east) -- (interior.south east) -- (interior.south west)
    -- cycle;}  }]
  \lipsum[2]
\end{tcolorbox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{enhanced,skin=enhancedlast jigsaw,
  attach boxed title to top left={xshift=-4mm,yshift=-0.5mm},
  fonttitle=\bfseries\sffamily,varwidth boxed title=0.7\linewidth,
  colbacktitle=blue!45!white,colframe=red!50!black,
  interior style={top color=blue!10!white,bottom color=red!10!white},
  boxed title style={empty,arc=0pt,outer arc=0pt,boxrule=0pt},
  underlay boxed title={
    \fill[blue!45!white] (title.north west) -- (title.north east)
      -- +(\tcboxedtitleheight-1mm,-\tcboxedtitleheight+1mm)
      -- ([xshift=4mm,yshift=0.5mm]frame.north east) -- +(0mm,-1mm)
      -- (title.south west) -- cycle;
    \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north west)
      -- +(-0.4,0) -- +(0,-0.3) -- cycle;
    \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north east)
      -- +(0,-0.3) -- +(0.4,0) -- cycle;  },
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{enhanced,
  attach boxed title to top left={xshift=1cm,yshift=-2mm},
  fonttitle=\bfseries,varwidth boxed title=0.7\linewidth,
  colbacktitle=green!45!white,coltitle=green!10!black,colframe=green!50!black,
  interior style={top color=yellow!10!white,bottom color=green!10!white},
  boxed title style={enhanced,boxrule=0.75mm,colframe=white,
    borderline={0.1mm}{0mm}{green!50!black},
    borderline={0.1mm}{0.75mm}{green!50!black},
    interior style={top color=green!10!white,bottom color=green!10!white,
      middle color=green!50!white},
    drop fuzzy shadow},
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

My title

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[2][]{skin=enhancedlast jigsaw,interior hidden,
  boxsep=0pt,top=0pt,colframe=red,coltitle=red!50!black,
  fonttitle=\bfseries\sffamily,
  attach boxed title to bottom center,
  boxed title style={empty,boxrule=0.5mm},
  varwidth boxed title=0.5\linewidth,
  underlay boxed title={
    \draw[white,line width=0.5mm]
      ([xshift=0.3mm-\tcboxedtitleheight*2,yshift=0.3mm]title.north west)
      --([xshift=-0.3mm+\tcboxedtitleheight*2,yshift=0.3mm]title.north east);
    \path[draw=red,top color=white,bottom color=red!50!white,line width=0.5mm]
    ([xshift=0.25mm-\tcboxedtitleheight*2,yshift=0.25mm]title.north west)
    cos +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
    sin +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
    -- ([xshift=0.25mm,yshift=0.25mm]title.south west)
    -- ([yshift=0.25mm]title.south east)
    cos +(\tcboxedtitleheight,\tcboxedtitleheight/2)
    sin +(\tcboxedtitleheight,\tcboxedtitleheight/2); },
  title={#2},#1}

\begin{mybox}{My title}
  \lipsum[2]
\end{mybox}
```

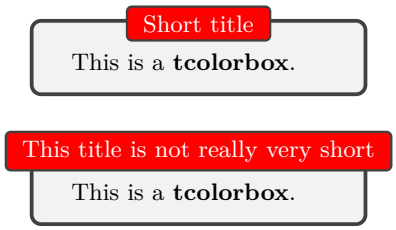
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My title

/tcb/hbox boxed title (no value, initially set)

The title text content is captured with a horizontal box. Especially, there are no linebreak possible.

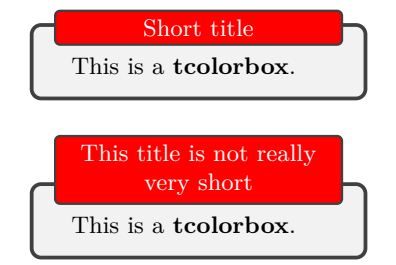
```
\newtcolorbox{mybox}[1]{hbox boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/minipage boxed title=⟨*length*⟩ (initially unset)

The title text content is captured with a minipage with a width of ⟨*length*⟩. By default, the resulting boxed title is somewhat smaller than the main box.

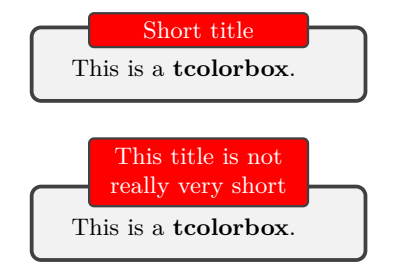
```
\newtcolorbox{mybox}[1]{minipage boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/minipage boxed title*=⟨*length*⟩ (initially unset)

The title text content is captured with a minipage with a width of main box width plus ⟨*length*⟩. By default, the resulting boxed title is somewhat smaller than the main box.

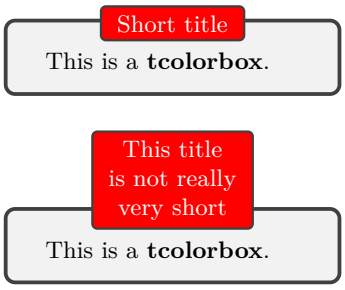
```
\newtcolorbox{mybox}[1]{minipage boxed title*=-2cm,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/tikznode boxed title=⟨*options*⟩ (initially unset)

The title text content is captured with a Ti*k*Z node with given Ti*k*Z ⟨*options*⟩. The text is centered by default

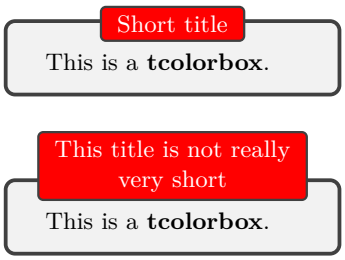
```
\newtcolorbox{mybox}[1]{tikznode boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title\\is not really\\very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/varwidth boxed title=⟨*length*⟩ (initially unset)

The title text content is captured with a `varwidth` environment with a width of ⟨*length*⟩. This style needs the `varwidth` package [1] to be loaded manually. By default, the resulting boxed title is somewhat smaller than the main box.

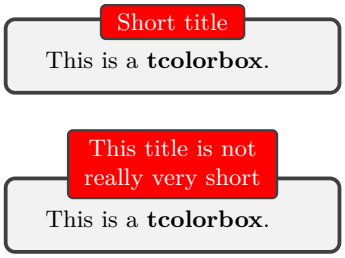
```
% \usepackage{varwidth}
\newtcolorbox{mybox}[1]{varwidth boxed title,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



/tcb/varwidth boxed title*=⟨*length*⟩ (initially unset)

The title text content is captured with a `varwidth` environment with a width of main box width plus ⟨*length*⟩. This style needs the `varwidth` package [1] to be loaded manually. By default, the resulting boxed title is somewhat smaller than the main box.

```
% \usepackage{varwidth}
\newtcolorbox{mybox}[1]{varwidth boxed title*=-2cm,
  enhanced,attach boxed title to top center=
    {yshift=-3mm,yshifttext=-1mm},
  boxed title style={size=small,colback=red},
  center title,title={#1}}

\begin{mybox}{Short title}
  This is a \textbf{tcolorbox}.
\end{mybox}\bigskip

\begin{mybox}{This title is not really very short}
  This is a \textbf{tcolorbox}.
\end{mybox}
```



7.3 Watermark Option Keys

The following watermark options are applicable for all skins which use `tikzpicture` as `/tcb/graphical environment`→P. 89. Therefore, the skin `standard`→P. 148 does not support these watermarks, but all other skins, e. g. `enhanced`→P. 150.

The watermark options rely on the more general overlay options described in Section 3.10 from page 55. Therefore, *watermarks* and *overlays* cannot be used mixed. But a mixture is possible with the LIB hooks library, see Section 13.

`/tcb/watermark text=`⟨*text*⟩ (no default, initially unset)

Writes some ⟨*text*⟩ in the center of the interior region of a `tcolorbox`. This ⟨*text*⟩ is written *after* the frame and interior are drawn and *before* the text content is drawn. It is zoomed or stretched according the values of `/tcb/watermark zoom`→P. 117 or `/tcb/watermark stretch`→P. 119.

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

\begin{tcolorbox}[enhanced,title=My title,watermark text=My Watermark]
\lipsum[1]
\tcblower
\lipsum[2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My Watermark

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark text on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default, initially unset)

This option writes some ⟨*text*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark text`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable`→P. 255.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.
- `first and middle`: first and middle parts of a break sequence.

`/tcb/watermark graphics`=⟨*file name*⟩ (no default, initially unset)

Draws an external picture referenced by ⟨*file name*⟩ in the center of the interior region of a `tcolorbox`. The picture is drawn *after* the frame and interior are drawn and *before* the text content is drawn. It is zoomed or stretched according the values of `/tcb/watermark zoom` → P. 117 or `/tcb/watermark stretch` → P. 119.

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

\begin{tcolorbox}[enhanced,title=My title,watermark graphics=Basilica_5.png,
  watermark opacity=0.15]
\lipsum[1-2]
\tcblower
This example uses a public domain picture from\\
\url{http://commons.wikimedia.org/wiki/File:Basilica_5.png}
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This example uses a public domain picture from
http://commons.wikimedia.org/wiki/File:Basilica_5.png

`/tcb/watermark graphics on`=⟨*part*⟩ `is` ⟨*file name*⟩ (no default, initially unset)

This option draws a picture referenced by ⟨*file name*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark graphics`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable` → P. 255.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.

`/tcb/watermark tikz=`⟨*graphical code*⟩ (no default, initially unset)

Draws the given `tikz` ⟨*graphical code*⟩ in the center of the interior region of a `tcolorbox`. The code is executed *after* the frame and interior are drawn and *before* the text content is drawn. The result is zoomed or stretched according the values of `/tcb/watermark zoom` → P. 117 or `/tcb/watermark stretch` → P. 119.

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

\begin{tcolorbox}[enhanced,title=My title,
  watermark tikz={\draw[line width=2mm] circle (1cm)
    node{\fontfamily{ptm}\fontseries{b}\fontsize{20mm}{20mm}\selectfont ?};}]
\lipsum[1]
\tcblower
\lipsum[2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark tikz on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default, initially unset)

This option draws the given `tikz` ⟨*graphical code*⟩ in the center of the interior region of a `tcolorbox` as described for `/tcb/watermark tikz`. But this is done only for boxes named ⟨*part*⟩ of a break sequence, see `/tcb/breakable` → P. 255.

Feasible values for ⟨*part*⟩ are:

- `broken`: all broken box parts,
- `unbroken`: unbroken boxes only,
- `first`: first parts of a break sequence,
- `middle`: middle parts of a break sequence,
- `last`: last parts of a break sequence,
- `unbroken and first`: unbroken boxes and first parts of a break sequence,
- `middle and last`: middle and last parts of a break sequence.

`/tcb/no watermark` (style, no default, initially set)

Removes the watermark if set before. This is an alias for `/tcb/no overlay` → P. 56.

`/tcb/watermark opacity=`⟨*fraction*⟩ (no default, initially `1.00`)

Sets the opacity value ∈ [0, 1] for a watermark.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark text=Watermark,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Opacity 1.00,watermark opacity=1.00]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Opacity 0.50,watermark opacity=0.50]
\lipsum[2]
\end{tcolorbox}%
```

Opacity 1.00

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Opacity 0.50

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark zoom=`⟨*fraction*⟩ (no default, initially `0.75`)

Sets the zoom value for a watermark. The zoom respects the aspect ratio. The value 1.0 means to fill the whole box until the watermark touches the frame.

```
\tcbset{enhanced,colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark text=Watermark,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Zoom 1.0,watermark zoom=1.0]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Zoom 0.5,watermark zoom=0.5]
\lipsum[2]
\end{tcolorbox}%
```

Zoom 1.0

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Zoom 0.5

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark shrink=`⟨*fraction*⟩ (no default, initially unset)

Identically to `/tcb/watermark zoom` → P. 117, but the watermark never gets enlarged. Thus, the watermark keeps its original size or is shrunk.

`/tcb/watermark overzoom=`⟨*fraction*⟩ (no default, initially unset)

Sets the overzoom value for a watermark. The overzoom respects the aspect ratio. The value 1.0 means to fill the whole box until the watermark touches all four sides of the frame.

```
\tcbset{enhanced,colback=white,colframe=blue!50!black,fonttitle=\bfseries,
  watermark opacity=0.5,
  watermark graphics=lichtspiel.jpg,nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Zoom 1.0,watermark zoom=1.0]
\lipsum[1]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Overzoom 1.0,watermark overzoom=1.0]
\lipsum[1]
\end{tcolorbox}%
```

Zoom 1.0

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Overzoom 1.0

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

If a `/tcb/watermark overzoom` value of `1.0` is used in connection with invisible top and bottom rules which still have a thickness greater than `0pt`, the space of these invisible rules may not be covered by the watermark. For example, this situation may occur during the breaking of `/tcb/enhanced` → P. 150 boxes. To avoid this optical glitch, just set `/tcb/pad at break` → P. 259 to any desired value.

/tcb/watermark stretch=⟨*fraction*⟩ (no default, initially unset)

Sets the stretch value for a watermark. The stretch value is applied to width and height in relation to the box dimensions. It does not respect the aspect ratio. The value 1.0 means to fill the whole box.

```
\tcbset{enhanced,colback=white,colframe=blue!50!black,fonttitle=\bfseries,
  watermark graphics=lichtspiel.jpg,watermark opacity=0.5,
  nobeforeafter,width=(\linewidth-2mm)/2}

\begin{tcolorbox}[title=Stretch 1.00,watermark stretch=1.00]
\lipsum[2]
\end{tcolorbox}\hfill%
\begin{tcolorbox}[title=Stretch 0.50,watermark stretch=0.50]
\lipsum[2]
\end{tcolorbox}%
```

Stretch 1.00

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Stretch 0.50

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

/tcb/watermark color=⟨*color*⟩ (no default, initially mixed background and frame color)

Sets the color for the watermark.

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

\begin{tcolorbox}[enhanced,title=My title,watermark text=My Watermark,
  watermark color=yellow!50!red]
\lipsum[1]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/clip watermark=true|false` (default `true`, initially `true`)

Sets the watermark to be clipped to the interior area.

```
\tcbset{enhanced,colback=white,colframe=blue!50!white,fonttitle=\bfseries,
  watermark opacity=0.5,watermark stretch=1.00,arc=3mm,
  watermark graphics=lichtspiel.jpg}

\begin{tcolorbox}[title=Clip (default),clip watermark]
\lipsum[1]
\end{tcolorbox}

\begin{tcolorbox}[title=No clip,clip watermark=false]
\lipsum[1]
\end{tcolorbox}%
```

Clip (default)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

No clip

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

7.4 Clip Environments

The following clip environments are applicable for all skins which use engines of type `path`, `pathfirst`, `pathmiddle`, or `pathlast`. Especially, the skin `enhanced` → P. 150 supports *all* of them and `standard` → P. 148 *none*. The typical area of application is inside overlay code, see Section 3.10 from page 55.

```
\begin{tcbclipframe}
  ⟨environment content⟩
\end{tcbclipframe}
```

Defines a `Tikz` scope which clips to the frame area path.

```
\makeatletter
\newtcolorbox{picturebox}[2][]{%
  enhanced,frame hidden,interior hidden,fonttitle=\bfseries,
  overlay={\begin{tcbclipframe}\node at (frame)
    {\includegraphics[width=\tcb@width,height=\tcb@height]{#2}};\end{tcbclipframe}%
    \begin{tcbclipinterior}\fill[white,opacity=0.75]
    (frame.south west) rectangle (frame.north east);\end{tcbclipinterior}},#1}
\makeatother

\begin{picturebox}[title=My Picture Box]{lichtspiel.jpg}
\lipsum[1]
\end{picturebox}
```

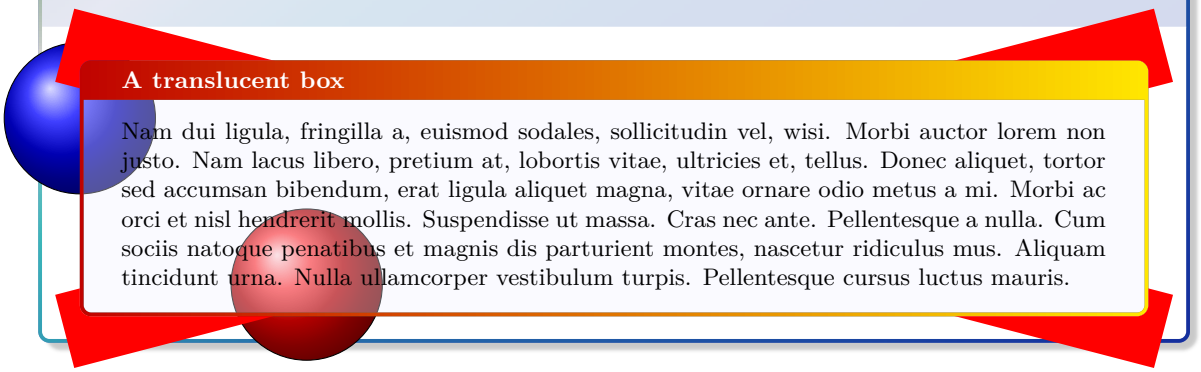
My Picture Box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\begin{tcbinvclipframe}
 ⟨environment content⟩
\end{tcbinvclipframe}
```

Defines a Tikz scope which clips to the *outside* of the frame area path.

```
\tcbset{enhanced jigsaw,fonttitle=\bfseries,opacityback=0.35,colback=blue!5!white,
  frame style={left color=red!75!black,right color=red!10!yellow}}

\begin{tikzpicture}% draw two balls
  \path[use as bounding box] (0,0.8) rectangle +(0.1,0.1);
  \shadedraw [shading=ball] (0,0) circle (1cm);
  \shadedraw [ball color=red] (3,-2.2) circle (1cm);
\end{tikzpicture}

\begin{tcolorbox}[title=A translucent box,
  overlay={\begin{tcbinvclipframe}
    \draw[red,line width=1cm] ([xshift=-2mm,yshift=2mm]frame.north west)
      --([xshift=2mm,yshift=-2mm]frame.south east);
    \draw[red,line width=1cm] ([xshift=-2mm,yshift=-2mm]frame.south west)
      --([xshift=2mm,yshift=2mm]frame.north east);
  \end{tcbinvclipframe}}]
  \lipsum[2]
\end{tcolorbox}
```



```
\begin{tcbclipinterior}
 ⟨environment content⟩
\end{tcbclipinterior}
```

Defines a Tikz scope which clips to the interior area path.

```
\begin{tcolorbox}[enhanced,title=My Title,
  overlay={\begin{tcbclipinterior}
    \draw[red,line width=1cm] (interior.north west)--(interior.south east);
    \draw[red,line width=1cm] (interior.south west)--(interior.north east);
  \end{tcbclipinterior}}]
\lipsum[1]
\end{tcolorbox}
```

My Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\begin{tcbcliptitle}
 ⟨environment content⟩
\end{tcbcliptitle}
```

Defines a Tikz scope which clips to the title area path.

```
\begin{tcolorbox}[enhanced,title=My Title,colframe=blue,colback=yellow!10!white,
  overlay={\begin{tcbcliptitle}\node at (title)
  {\includegraphics[width=\linewidth]{lichtspiel.jpg}};\end{tcbcliptitle}}]
\lipsum[1]
\end{tcolorbox}
```

My Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/clip title=true|false` (default `true`, initially `false`)

Sets the title to be clipped to the title area.

```
\tcbset{enhanced,width=5cm,colframe=red!50!white,coltitle=black,
  colbacktitle=yellow!50!white}

\begin{tcolorbox}[title=\mbox{This is a title which is unbreakable and far too long}]
This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[title=\mbox{This is a title which is unbreakable and far too long},
  clip title]
This is a tcolorbox.
\end{tcolorbox}
```

This is a title which is unbreakable and far too long

This is a tcolorbox.

This is a title which is unbreak

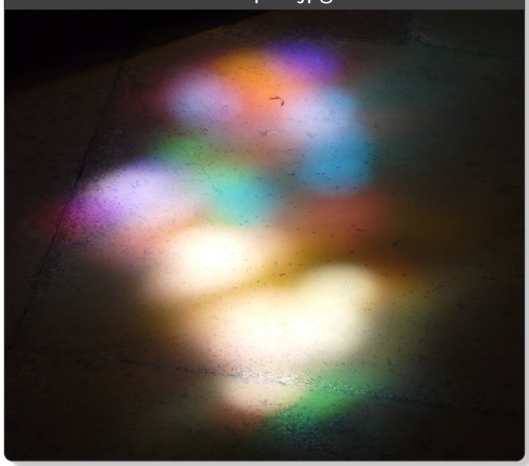
This is a tcolorbox.

`/tcb/clip upper=true|false` (default `true`, initially `false`)

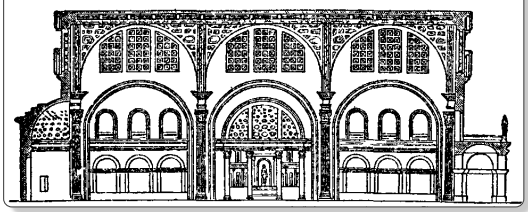
Sets the upper part to be clipped to the interior area.

```
\newcommand{\mygraphics}[2][]{%
  \tcbox[enhanced,boxsep=0pt,top=0pt,bottom=0pt,left=0pt,
    right=0pt,boxrule=0.4pt,drop fuzzy shadow,clip upper,
    colback=black!75!white,toptitle=2pt,bottomtitle=2pt,nobeforeafter,
    center title,fonttitle=\small\sffamily,title=\detokenize{#2}]
  {\includegraphics[width=\the\dimexpr(\linewidth-4mm)/2\relax]{#2}}}

\mygraphics{lichtspiel.jpg}\hfill
\mygraphics{Basilica_5.png}
```

lichtspiel.jpg



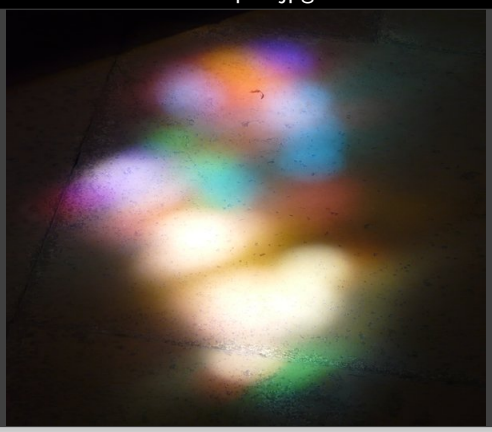
Basilica_5.png



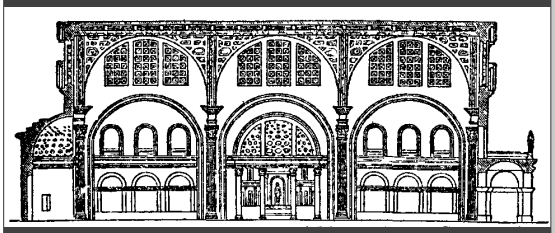
The example for /tcb/clip upper → P. 124 sizes the box according to the dimensions of the picture. To do it the other way around, the watermark options provide an easy solution.

```
\newcommand{\mygraphics}[2][]{%
  \tcbox[enhanced,capture=minipage,boxsep=0pt,top=0pt,bottom=0pt,left=0pt,
    right=0pt,boxrule=0.4pt,drop fuzzy shadow,nobeforeafter,
    colback=black!75!white,toptitle=2pt,bottomtitle=2pt,
    center title,fonttitle=\small\sffamily,title=\detokenize{#2},
    width=(\linewidth-4mm)/2,height=6cm,colbacktitle={black},
    watermark zoom=1.0,watermark graphics={#2}]{}}

\mygraphics{lichtspiel.jpg}\hfill
\mygraphics{Basilica_5.png}
```

lichtspiel.jpg



Basilica_5.png



/tcb/clip lower=true|false (default `true`, initially `false`)

Sets the lower part to be clipped to the interior area.

```
\tcbset{enhanced,width=5cm,colframe=red!50!black,text and listing}

\begin{tcblisting}{}
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
\end{tcblisting}

\begin{tcblisting}{clip lower}
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
\end{tcblisting}
```

Donaudampfschifffahrtska-
pitänsmützenfransen

```
Donau\-dampf\-schiff\-fahrts\-ka\-pi\-t\"ans\-m\"ut\-zen\-fran\-sen
```

Donaudampfschifffahrtska-
pitänsmützenfransen

```
Donau\-dampf\-schiff\-fa
```

7.5 Border Line Option Keys

The following border line options are applicable for most skins which use `tikzpicture` as `/tcb/graphical environment` → P. 89. Therefore, the skin `standard` → P. 148 does not support these border lines, but most other skins, e. g. `enhanced` → P. 150.

The border lines are independent from the normal `tcolorbox` rules. They may be used with or without the `/tcb/segmentation engine` → P. 91.

The border lines are stackable, i. e. several different border lines can be used on the same `tcolorbox`. They are drawn *after* the box frame and box interior and *before* overlays or watermarks.

Technically, the normal `tcolorbox` rules result from a Ti*k*Z *filling* process. The border lines are created by a Ti*k*Z *drawing* process. This can be used to apply different effects.

`/tcb/borderline={`⟨*width*⟩`}{`⟨*offset*⟩`}{`⟨*options*⟩`}` (no default, initially unset)

Adds a new border line to the stack of border lines. This border line is drawn with the given ⟨*width*⟩ and gets a ⟨*offset*⟩ computed from the frame outline. A positive ⟨*offset*⟩ value moves the borderline inside the `tcolorbox` and a negative ⟨*offset*⟩ value moves it outside without changing the bounding box.

The border line is drawn along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩. Note that the Ti*k*Z `line width` option should not be used here.

The border lines adapt to the rounded corners of the `tcolorbox`. An inside border line will switch to sharp corners if necessary, an outside border line will always be rounded except for `/tcb/sharp corners` → P. 34.

```
\begin{tcolorbox}[enhanced,title=Rounded corners,fonttitle=\bfseries,boxsep=5pt,
  arc=8pt,
  borderline={0.5pt}{0pt}{red},
  borderline={0.5pt}{5pt}{blue,dotted},
  borderline={0.5pt}{-5pt}{green} ]
This is a tcolorbox.
\end{tcolorbox}
\bigskip
\begin{tcolorbox}[enhanced,title=Sharp corners,fonttitle=\bfseries,boxsep=5pt,
  arc=8pt,sharp corners=downhill,
  borderline={0.5pt}{0pt}{red},
  borderline={0.5pt}{5pt}{blue,dotted},
  borderline={0.5pt}{-5pt}{green} ]
This is a tcolorbox.
\end{tcolorbox}
```

Rounded corners

This is a tcolorbox.

Sharp corners

This is a tcolorbox.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,arc=3mm,boxrule=1.5mm,boxsep=1.5mm,
  colback=yellow!20!white,
  colframe=blue,
  borderline={1mm}{1mm}{white},
  borderline={1mm}{2mm}{red} ]
  \lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,arc=3mm,boxrule=1.5mm,
  frame hidden,colback=blue!10!white,
  borderline={1mm}{0mm}{blue,dotted} ]
  \lipsum[2]
\end{tcolorbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
% \usepackage{lipsum}
\begin{tcolorbox}[enhanced,skin=enhancedmiddle,
  frame hidden,interior hidden,top=0mm,bottom=0mm,boxsep=0mm,
  borderline={0.75mm}{0mm}{red},
  borderline={0.75mm}{0.75mm}{red!50!yellow},
  borderline={0.75mm}{1.5mm}{yellow}, ]
  \lipsum[3]
\end{tcolorbox}
```

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```
% \usepackage{lipsum}
\newtcolorbox{mygreenbox}[2][]{%
  enhanced,width=\linewidth-6pt,
  enlarge top by=3pt,enlarge bottom by=3pt,
  enlarge left by=3pt,enlarge right by=3pt,
  title={#2},frame hidden,boxrule=0pt,top=1mm,bottom=1mm,
  colframe=green!30!black, colbacktitle=green!50!yellow,
  coltitle=black, colback=green!25!white,
  borderline={0.5pt}{-0.5pt}{green!75!blue},
  borderline={1pt}{-3pt}{green!50!blue},#1}

\begin{mygreenbox}{My title}
  \lipsum[4]
\end{mygreenbox}
```

My title

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

/tcb/no borderline (no default, initially set)

Removes all border lines if set before.

/tcb/show bounding box=⟨*color*⟩ (default **red**, initially unset)

Displays the bounding box borderline of a **tcolorbox**. Its intended use is debugging and fine tuning. It should not be part of a final document. The optional ⟨*color*⟩ is the base color for the bounding box borderline.

```
\tcbset{enhanced,nobeforeafter,width=4cm,fonttitle=\bfseries}

\begin{tcolorbox}[show bounding box,title=Normal]
This is a tcolorbox.
\end{tcolorbox}%
\begin{tcolorbox}[show bounding box=blue,title=Shadow,drop fuzzy shadow]
This is a tcolorbox.
\end{tcolorbox}%
\begin{tcolorbox}[show bounding box=green,title=Enlarged,drop fuzzy shadow,
  enlarge by=2mm]
This is a tcolorbox.
\end{tcolorbox}
```

Normal

This is a tcolorbox.

Shadow

This is a tcolorbox.

Enlarged

This is a tcolorbox.

7.6 Shadow Option Keys

The following shadow options are applicable for most skins which use `tikzpicture` as `/tcb/graphical environment` → P. 89. Therefore, the skin `standard` → P. 148 does not support these shadows, but most other skins, e. g. `enhanced` → P. 150.

The shadows are stackable, i. e. several different shadows can be used on the same `tcolorbox`. They are drawn *before* the box frame is drawn.

`/tcb/no shadow` (no default)

Removes all shadows if set before.

`/tcb/shadow={`⟨*xshift*⟩`}{`⟨*yshift*⟩`}{`⟨*offset*⟩`}{`⟨*options*⟩`}` (no default)

Adds a new shadow to the stack of shadows. This shadow is follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩. The ⟨*offset*⟩ value is a distance value from the frame outline. A positive ⟨*offset*⟩ value shrinks the shadow and a negative ⟨*offset*⟩ value enlarges the shadow. The shadow is filled along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩.

The shadows adapt to the rounded corners of the `tcolorbox`. An shrinked shadow will switch to sharp corners if necessary, an enlarged shadow may become more rounded depending on several factors. But `/tcb/sharp corners` → P. 34 have sharp shadows.

Shadows are not considered for the bounding box computation by default. Large shadows may be overlaped by the following content. But, the bounding box can be adapted if necessary.

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

\begin{tcolorbox}[title=My own shadow,
  shadow={2mm}{-1mm}{0mm}{black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  shadow={-1mm}{-2mm}{0mm}{fill=blue,
    opacity=0.5}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Double shadow,
  shadow={-1.5mm}{-1.5mm}{0mm}{fill=blue,
    opacity=0.25},
  shadow={1.5mm}{-1.5mm}{0mm}{fill=red,
    opacity=0.25}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Far shadow,
  shadow={5.5mm}{-3.5mm}{2mm}{fill=black,
    opacity=0.25}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Halo shadow,
  shadow={0mm}{0mm}{-1.5mm}%
    {fill=yellow!75!red,opacity=0.5}]
This is a tcolorbox.
\end{tcolorbox}
```

My own shadow

This is a tcolorbox.

Another shadow

This is a tcolorbox.

Double shadow

This is a tcolorbox.

Far shadow

This is a tcolorbox.

Halo shadow

This is a tcolorbox.

`/tcb/fuzzy shadow={⟨xshift⟩}{⟨yshift⟩}{⟨offset⟩}{⟨step⟩}{⟨options⟩}` (no default)

Adds a new fuzzy shadow to the stack of shadows. Actually, this option adds several shadows which appear like a shadow with a fuzzy border. This fuzzy shadow is follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩. The ⟨*offset*⟩ value is a distance value from the frame outline. A positive ⟨*offset*⟩ value shrinks the shadow and a negative ⟨*offset*⟩ value enlarges the shadow. The `{`⟨*step*⟩`}` value describes a shrink offset used for the combination of the partial shadows. The shadow is filled along a Ti*k*Z path with the given Ti*k*Z ⟨*options*⟩ but any `opacity` value will be ignored.

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

\begin{tcolorbox}[title=My own shadow,
  fuzzy shadow={2mm}{-1mm}{0mm}{0.1mm}%
               {black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  fuzzy shadow={-1mm}{-2mm}{0mm}{0.2mm}%
               {fill=blue}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Double shadow,
  fuzzy shadow={-1.5mm}{-1.5mm}{0mm}{0.1mm}%
               {blue},
  fuzzy shadow={1.5mm}{-1.5mm}{0mm}{0.1mm}%
               {red}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip
\begin{tcolorbox}[title=Far shadow,
  fuzzy shadow={5.5mm}{-3.5mm}{0mm}{0.3mm}%
               {black}]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Glow shadow,
  fuzzy shadow={0mm}{0mm}{-1.5mm}{0.15mm}%
               {yellow!75!red}]
This is a tcolorbox.
\end{tcolorbox}
```

My own shadow

This is a tcolorbox.

Another shadow

This is a tcolorbox.

Double shadow

This is a tcolorbox.

Far shadow

This is a tcolorbox.

Glow shadow

This is a tcolorbox.

```
\newtcolorbox{mybox}[1][]{enhanced,
  fuzzy shadow={1.0mm}{-1.0mm}{0.12mm}{0mm}{blue!50!white},
  fuzzy shadow={-1.0mm}{-1.0mm}{0.12mm}{0mm}{red!50!white},
  fuzzy shadow={-1.0mm}{1.0mm}{0.12mm}{0mm}{green!50!white},
  fuzzy shadow={1.0mm}{1.0mm}{0.12mm}{0mm}{yellow!50!white},#1
}

\begin{mybox}[title=A multi shadow box]
This is a tcolorbox.
\end{mybox}
```

A multi shadow box

This is a tcolorbox.

`/tcb/drop shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

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

\begin{tcolorbox}[drop shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop fuzzy shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

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

\begin{tcolorbox}[drop fuzzy shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop fuzzy shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop midday shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

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

\begin{tcolorbox}[drop midday shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop midday shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop fuzzy midday shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

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

\begin{tcolorbox}[drop fuzzy midday shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop fuzzy midday shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/halo=`⟨*size*⟩ `with` ⟨*color*⟩ (style, default `0.9mm with yellow`)

Adds a new halo shadow with the given ⟨*color*⟩ which overlaps the colorbox an all sides by ⟨*size*⟩.

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

\begin{tcolorbox}[title=My own halo,halo]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Another halo,
  halo=2mm with green]
This is a tcolorbox.
\end{tcolorbox}
```

My own halo

This is a tcolorbox.

Another halo

This is a tcolorbox.

`/tcb/fuzzy halo=`⟨*size*⟩ `with` ⟨*color*⟩ (style, default `0.9mm with yellow`)

Adds a new fuzzy halo shadow with the given ⟨*color*⟩ which overlaps the colorbox an all sides by ⟨*size*⟩ plus `0.48mm`.

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

\begin{tcolorbox}[title=My own halo,fuzzy halo]
This is a tcolorbox.
\end{tcolorbox}
\par\bigskip\bigskip
\begin{tcolorbox}[title=Another halo,
  fuzzy halo=2mm with green]
This is a tcolorbox.
\end{tcolorbox}
```

My own halo

This is a tcolorbox.

Another halo

This is a tcolorbox.

```
\begin{tcolorbox}[blank,enhanced jigsaw,boxsep=2pt,arc=2pt,
  fuzzy halo=2mm with red!50!white,
  fuzzy halo=1mm with white]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

For all following shadows, the optionally given ⟨*color*⟩ for the shadow can be changed equivalent to the preceding examples.

`/tcb/drop shadow southeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop shadow` → P. 131.

```
\begin{tcolorbox}[drop shadow southeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow south=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop midday shadow` → P. 131.

```
\begin{tcolorbox}[drop shadow south,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow southwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow southwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow west=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow west,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow northwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow northwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow north=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

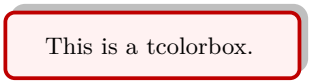
```
\begin{tcolorbox}[drop shadow north,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

`/tcb/drop shadow northeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow northeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop shadow east=`⟨*color*⟩ (style, default `black!50!white`)

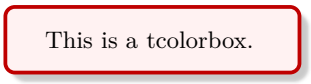
Adds a new shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop shadow east,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```

`/tcb/drop fuzzy shadow southeast=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop fuzzy shadow` → P. 131.

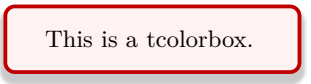
```
\begin{tcolorbox}[drop fuzzy shadow southeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop fuzzy shadow south=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows. This shadow is identical to `/tcb/drop fuzzy midday shadow` → P. 131.

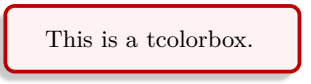
```
\begin{tcolorbox}[drop fuzzy shadow south,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop fuzzy shadow southwest=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

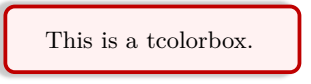
```
\begin{tcolorbox}[drop fuzzy shadow southwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/drop fuzzy shadow west=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

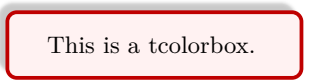
```
\begin{tcolorbox}[drop fuzzy shadow west,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



/tcb/drop fuzzy shadow northwest=⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

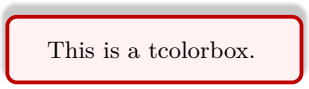
```
\begin{tcolorbox}[drop fuzzy shadow northwest,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



/tcb/drop fuzzy shadow north=⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

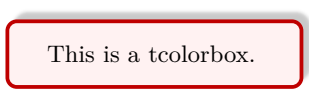
```
\begin{tcolorbox}[drop fuzzy shadow north,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



/tcb/drop fuzzy shadow northeast=⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

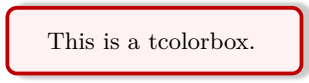
```
\begin{tcolorbox}[drop fuzzy shadow northeast,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



/tcb/drop fuzzy shadow east=⟨*color*⟩ (style, default `black!50!white`)

Adds a new fuzzy shadow with standard dimensions to the stack of shadows.

```
\begin{tcolorbox}[drop fuzzy shadow east,
  enhanced,colback=red!5!white,colframe=red!75!black]
  This is a tcolorbox.
\end{tcolorbox}
```



`/tcb/lifted shadow={⟨xshift⟩}{⟨yshift⟩}{⟨bend⟩}{⟨step⟩}{⟨options⟩}` (no default)

Adds a new lifted shadow to the stack of shadows. Actually, this option adds several shadows which appear like a shadow with a fuzzy border. This lifted shadow is follows the outline of the `tcolorbox` but is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩ on the lower left corner and by −⟨*xshift*⟩ and ⟨*yshift*⟩ on the lower right corner. Additionally, there is a ⟨*bend*⟩ in the middle. The `{`⟨*step*⟩`}` value describes a shrink offset used for the combination of the partial shadows. The shadow is filled along a TikZ path with the given TikZ ⟨*options*⟩ but any `opacity` value will be ignored.

```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.1pt,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=My own shadow,
  lifted shadow={1mm}{-2mm}{3mm}{0.1mm}%
                {black!50!white}]
This is a tcolorbox.
\end{tcolorbox}
```

My own shadow

This is a tcolorbox.

`/tcb/drop lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.4pt,sharp corners,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[drop lifted shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop lifted shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

Another shadow

This is a tcolorbox.

`/tcb/drop small lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new small lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

```
\tcbset{enhanced,colback=red!5!white,
  boxrule=0.4pt,sharp corners,
  colframe=red!75!black,fonttitle=\bfseries}

\tcbox[drop small lifted shadow,size=fbox]
  {This is a tcolorbox.}
\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop small lifted shadow=black]
This is a tcolorbox.
\end{tcolorbox}
```

This is a tcolorbox.

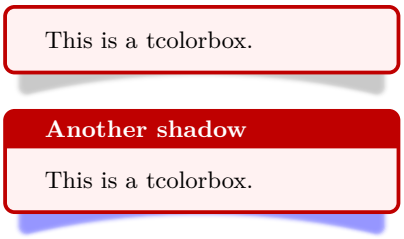
Another shadow

This is a tcolorbox.

`/tcb/drop large lifted shadow=`⟨*color*⟩ (style, default `black!50!white`)

Adds a new large lifted shadow with standard dimensions to the stack of shadows. Optionally, the ⟨*color*⟩ for the shadow can be changed.

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

\begin{tcolorbox}[drop large lifted shadow]
This is a tcolorbox.
\end{tcolorbox}\par\bigskip
\begin{tcolorbox}[title=Another shadow,
  drop large lifted shadow=blue]
This is a tcolorbox.
\end{tcolorbox}
```



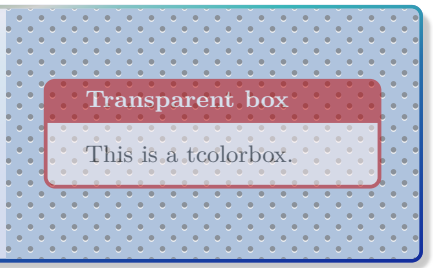
7.7 TikZ Picture Option Keys

The following general options are applicable for skins which use `tikzpicture` as `/tcb/graphical environment` → P. 89. Therefore, the skin `standard` → P. 148 does not support these options, but most other skins, e. g. `enhanced` → P. 150.

`/tcb/tikz=`⟨*tikz option list*⟩ (no default, initially empty)

Adds the given ⟨*tikz option list*⟩ to the main `tikzpicture` environment used to draw the color box, see [20]. If this option is applied a second time, the new ⟨*tikz option list*⟩ is appended to the current option list.

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

\begin{tcolorbox}[title=Transparent box,
  tikz={opacity=0.5,transparency group}]
This is a tcolorbox.
\end{tcolorbox}
```




```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  fontupper=\bfseries\Huge,
  center title,center upper}

\begin{tcolorbox}[title=Rotated box,
  tikz={rotate=30}]
Sold!
\end{tcolorbox}
```

`/tcb/tikz reset` (initially set)

Removes all options given by `/tcb/tikz`.

`/tcb/at begin tikz=`⟨*tikz code*⟩ (no default, initially empty)

The given ⟨*tikz code*⟩ is executed at the beginning of the `tikzpicture` environment after the TikZ option `execute at begin picture` was applied. If this option is applied a second time, the new ⟨*tikz code*⟩ is appended to the current code.

`/tcb/at begin tikz reset` (initially set)

Removes all code given by `/tcb/at begin tikz`.

`/tcb/at end tikz=`⟨*tikz code*⟩ (no default, initially empty)

The given ⟨*tikz code*⟩ is executed at the ending of the `tikzpicture` environment before the TikZ option `execute at end picture` was applied. If this option is applied a second time, the new ⟨*tikz code*⟩ is appended to the current code.

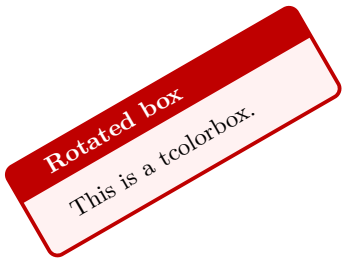
`/tcb/at end tikz reset` (initially set)

Removes all code given by `/tcb/at end tikz`.

/tcb/rotate=⟨*angle*⟩ (no default, initially unset)

Rotates the `tcolorbox` by the given ⟨*angle*⟩. Note that this is a Ti*k*Z coordinate transformation i.e. not all graphical elements like shadings will really be rotated.

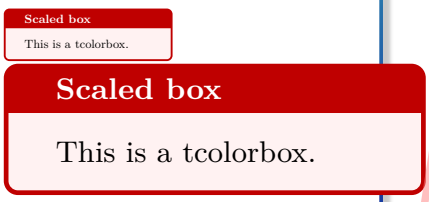
```
\tcbset{enhanced,colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[title=Rotated box,rotate=30]
This is a tcolorbox.
\end{tcolorbox}
```



/tcb/scale=⟨*fraction*⟩ (no default, initially unset)

Scales the `tcolorbox` by the given ⟨*fraction*⟩. Note that this is a Ti*k*Z coordinate transformation i.e. not all graphical elements like line widths will really be scaled.

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

\begin{tcolorbox}[title=Scaled box,scale=0.5]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[title=Scaled box,scale=1.25]
This is a tcolorbox.
\end{tcolorbox}
```



/tcb/remember (style, initially unset)

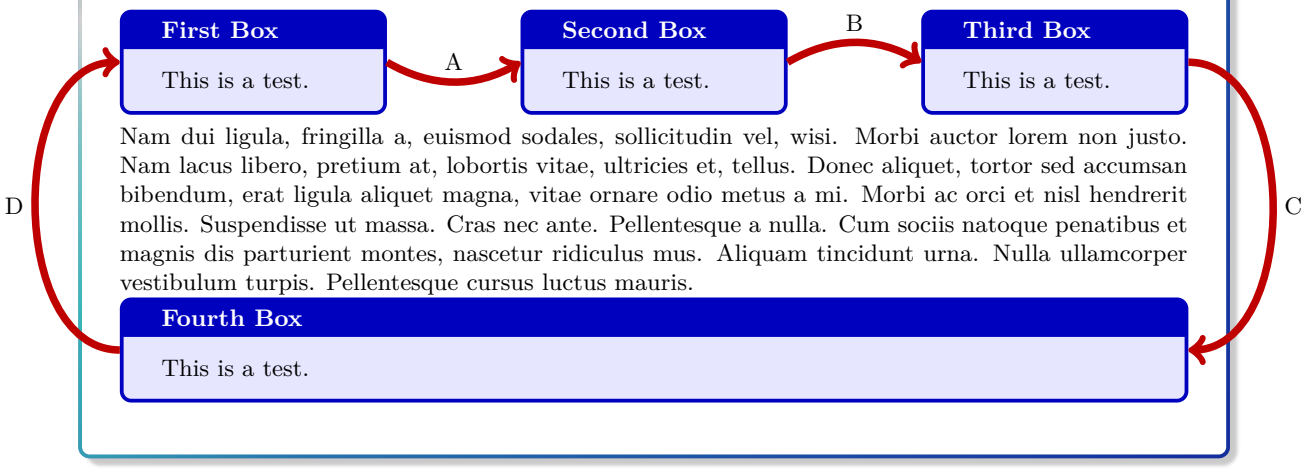
Shortcut for `tikz={remember picture}`. This allows one to reference nodes in other Ti*k*Z pictures.

```
\begin{tcolorbox}[enhanced,remember,colback=red!5!white,colframe=red!75!black,
  fonttitle=\bfseries,title=The four corners of a paper,
  overlay={\draw[red!50!white,line width=1mm,opacity=0.5,shorten >=3mm]
    (frame.north west) edge[->] (current page.north west)
    (frame.north east) edge[->] (current page.north east)
    (frame.south west) edge[->] (current page.south west)
    (frame.south east) edge[->] (current page.south east);}]
This is a tcolorbox.
\end{tcolorbox}
```

`/tcb/remember as=`⟨*name*⟩ (style, no default, initially unset)

The `frame` node will be remembered by the given ⟨*name*⟩ to be referenced in other TikZ pictures.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}[1][]{enhanced,colframe=blue!75!black,colback=blue!10!white,
  fonttitle=\bfseries,#1}

\begin{mybox}[title=First Box,nobeforeafter,width=\linewidth/4,remember as=one]
This is a test.
\end{mybox}
\hfill
\begin{mybox}[title=Second Box,nobeforeafter,width=\linewidth/4,remember as=two]
This is a test.
\end{mybox}
\hfill
\begin{mybox}[title=Third Box,nobeforeafter,width=\linewidth/4,remember as=three]
This is a test.
\end{mybox}

\lipsum[2]

\begin{mybox}[title=Fourth Box,remember as=four]
This is a test.
\end{mybox}

\begin{tikzpicture}[overlay,remember picture,line width=1mm,draw=red!75!black]
  \draw[->] (one.east) to[bend right] node[above] {A} (two.west);
  \draw[->] (two.east) to[bend left] node[above] {B} (three.west);
  \draw[->] (three.east) to[bend left=90] node[right] {C} (four.east);
  \draw[->] (four.west) to[bend left=90] node[left] {D} (one.west);
\end{tikzpicture}
```



Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7.8 Underlay Option Keys

Underlays are quite similar to overlays described in Section 3.10 on page 55. Underlays are drawn *after* the frame and interior are drawn and *before* overlays and the text content is drawn; see Section 6.4 on page 97 for the general drawing scheme.

The differences between underlays and overlays are:

- Underlays are not applicable for the skins `standard`→P. 148 and `standard jigsaw`→P. 149, whereas overlays are applicable also for these skins. The skin `spartan`→P. 186 supports underlays but no overlays.

If an underlay is used with the `standard`→P. 148 skin, it is silently ignored.

- Underlays are stackable, i. e. several different underlays can be used on the same `tcolorbox`. Overlays are not stackable by default (but with some help of the library LIB hooks).
- Boxed titles are implemented with underlays (Section 7.2 on page 107), watermarks are implemented with overlays (Section 7.3 on page 114).

`/tcb/underlay`=⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the frame and interior and *before* the text content.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,
  colframe=red!75!black,fonttitle=\bfseries,watermark color=yellow!50!white,
  underlay={\begin{tcbclipinterior}
    \draw[red!40!white,line width=1cm] (interior.south west)--(interior.north east);
    \end{tcbclipinterior}},
  attach boxed title to top center={yshift=-2mm},#1}

\begin{mybox}[title=My box,watermark text=My Watermark]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

My Watermark

`/tcb/no underlay` (style, no default, initially set)

Removes the underlay if set before.

`/tcb/underlay broken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/underlay unbroken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 255 but *is not* broken actually or if the box is set to be `/tcb/unbreakable` →P. 256, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/no underlay unbroken` (style, no default, initially set)

Removes the unbroken underlay if set before.

`/tcb/underlay first`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/no underlay first` (style, no default, initially set)

Removes the first underlay if set before.

`/tcb/underlay middle`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/no underlay middle` (style, no default, initially set)

Removes the middle underlay if set before.

`/tcb/underlay last`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` →P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/no underlay last` (style, no default, initially set)

Removes the last underlay if set before.

`/tcb/underlay boxed title`=⟨*graphical code*⟩ (no default, initially unset)

If the box has a *boxed title*, see Section 7.2 on page 107, then the ⟨*graphical code*⟩ is added to the box drawing process *before* the boxed title is drawn.

`/tcb/no underlay boxed title` (style, no default, initially set)

Removes the boxed title underlay if set before.

`/tcb/underlay unbroken and first`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay unbroken` and `/tcb/underlay first` together. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/underlay middle and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay middle` and `/tcb/underlay last` together. `/tcb/underlay` →P. 141 overwrites this key.

`/tcb/underlay unbroken and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay unbroken` and `/tcb/underlay last` together. `/tcb/underlay` →P. 141 overwrites this key.

N 2014-09-19 `/tcb/underlay first and middle`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/underlay first` and `/tcb/underlay middle` together. `/tcb/underlay` →P. 141 overwrites this key.

7.9 Finish Option Keys

Finishes are quite similar to underlays described in Section 7.8 on page 141 and overlays described in Section 3.10 on page 55. Finishes are drawn *after* the text content is drawn; see Section 6.4 on page 97 for the general drawing scheme. Therefore, a finish will reduce the readability of the text content.

Finishes are intended for special effects like highlights or glosses or text over text.

- Finishes are only applicable for the skins `enhanced`→ P. 150, `empty`→ P. 177, `freelance`→ P. 189, `bicolor`→ P. 163, `beamer`→ P. 168, and `widget`→ P. 173.

If a finish is used with the `standard`→ P. 148 skin, it is silently ignored.

- Finishes are stackable, i. e. several different finishes can be used on the same `tcolorbox`.

`/tcb/finish`=⟨*graphical code*⟩ (no default, initially unset)

Adds ⟨*graphical code*⟩ to the box drawing process. This ⟨*graphical code*⟩ is drawn *after* the text content.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,colframe=red!75!black,fonttitle=\bfseries,
  finish={\begin{tcbclipframe}
    \path[bottom color=black,top color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.south east) -- (frame.north east) -- cycle;
    \path[top color=white,bottom color=black!50!white,opacity=0.1]
      (frame.south west) -- (frame.north east) -- (frame.north west) -- cycle;
    \end{tcbclipframe}},#1}

\begin{mybox}[title=My box]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\newtcolorbox{mybox}[1][]{enhanced,colback=red!5!white,
  colbacktitle=red!85!black!50!white,colframe=red!75!black,fonttitle=\bfseries,
  finish={\node[draw,fill=white,fill opacity=0.85,inner sep=5mm,
    rounded corners] at (frame.center) {\Huge\bfseries Finish!};},#1}

\begin{mybox}[title=My box]
\lipsum[2]
\end{mybox}
```

My box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Finish!

`/tcb/no finish` (style, no default, initially set)

Removes the finish if set before.

`/tcb/finish broken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/finish unbroken`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 but *is not* broken actually or if the box is set to be `/tcb/unbreakable` → P. 256, then the ⟨*graphical code*⟩ is added to the box drawing process. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/no finish unbroken` (style, no default, initially set)

Removes the unbroken finish if set before.

`/tcb/finish first`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *first* part of the break sequence. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/no finish first` (style, no default, initially set)

Removes the first finish if set before.

`/tcb/finish middle`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *middle* parts (if any) of the break sequence. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/no finish middle` (style, no default, initially set)

Removes the middle finish if set before.

`/tcb/finish last`=⟨*graphical code*⟩ (no default, initially unset)

If the box is set to be `/tcb/breakable` → P. 255 and *is* broken actually, then the ⟨*graphical code*⟩ is added to the box drawing process for the *last* part of the break sequence. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/no finish last` (style, no default, initially set)

Removes the last finish if set before.

`/tcb/finish unbroken and first`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish unbroken` and `/tcb/finish first` together. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/finish middle and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish middle` and `/tcb/finish last` together. `/tcb/finish` → P. 143 overwrites this key.

`/tcb/finish unbroken and last`=⟨*graphical code*⟩ (no default, initially unset)

This is an abbreviation for setting `/tcb/finish unbroken` and `/tcb/finish last` together. `/tcb/finish` → P. 143 overwrites this key.

N 2014-09-19 `/tcb/finish first and middle`=⟨*graphical code*⟩ (no default, initially unset)

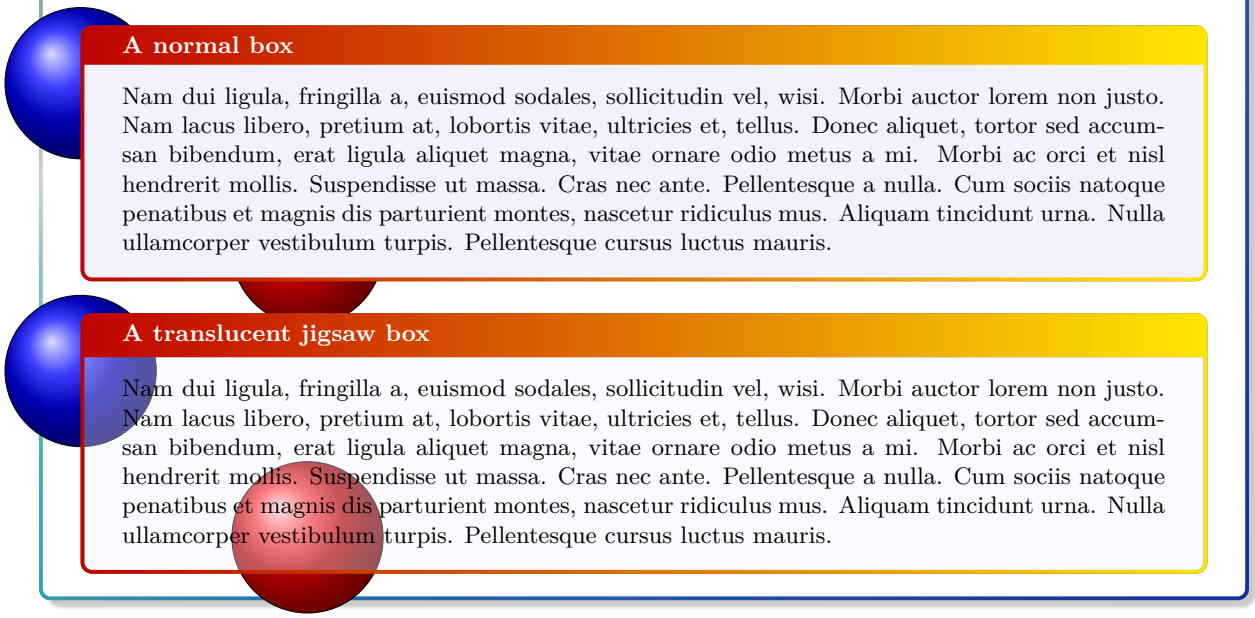
This is an abbreviation for setting `/tcb/finish first` and `/tcb/finish middle` together. `/tcb/finish` → P. 143 overwrites this key.

7.10 Jigsaw Skin Variants

As described in Section 6.1 on page 89, a `tcolorbox` is drawn by up to four *engines*. Typically, the *frame* engine fills the complete box area with color and the other engines fill certain areas with other colors. Finally, only the area which you see as *frame* of the box will display the frame color. For most applications, this is a good approach.

For certain boxes, a more delicate procedure is needed. E.g., if the box should be translucent, an already painted area cannot be made unpainted. Therefore, more elaborate frame engines saw holes into the frame where the interior area and optionally the title area will be painted. The resulting skins are called *jigsaw* skins. For `standard` → P. 148 and `enhanced` → P. 150, there are variants called `standard jigsaw` → P. 149 and `enhanced jigsaw` → P. 157.

```
\newcommand{\ballexample}{\begin{tikzpicture}
  \path[use as bounding box] (0,0.8) rectangle +(0.1,0.1);
  \shadedraw [shading=ball] (0,0) circle (1cm);
  \shadedraw [ball color=red] (3,-2.2) circle (1cm);
\end{tikzpicture}}

\tcbset{enhanced,colback=blue!5!white,
  frame style={left color=red!75!black,right color=red!10!yellow},
  fonttitle=\bfseries }

\ballexample

\begin{tcolorbox}[title=A normal box]
  \lipsum[2]
\end{tcolorbox}

\ballexample

\begin{tcolorbox}[title=A translucent jigsaw box,
  enhanced jigsaw,opacityback=0.35]
  \lipsum[2]
\end{tcolorbox}
```



A normal box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

A translucent jigsaw box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\tcbset{enhanced,colback=red!10!white,coltitle=black,
  frame style={left color=red!75!black,right color=red!10!yellow},
  fonttitle=\bfseries,interior hidden,title hidden}

\begin{tcolorbox}[title=A normal box with hidden interior and title]
  This is a tcolorbox.
\end{tcolorbox}

\begin{tcolorbox}[enhanced jigsaw,
  title=A jigsaw box with hidden interior and title]
  This is a tcolorbox.
\end{tcolorbox}
```

A normal box with hidden interior and title

This is a tcolorbox.

A jigsaw box with hidden interior and title

This is a tcolorbox.

```
\newtcolorbox{mybox}{skin=enhancedmiddle jigsaw,leftrule=5mm,rightrule=5mm,
  boxsep=0mm,top=0mm,bottom=0mm,
  frame style={top color=blue,bottom color=red},interior hidden}

\begin{mybox}
  \lipsum[2]
\end{mybox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7.11 Draft Mode

To reduce the compiliation time while drafting a document, the *draft mode* can be applied. Basically, it changes all skins to spartan→ P. 186 and sets the /tcb/fit algorithm→ P. 281 to squeeze. Especially, when fuzzy shadows are used, the speedup will be considerable high.

It is strongly recommended that the draft mode is *not* used for the final document. Use spartan→ P. 186 directly, if you want to stay with it. The draft mode implementation may change in future.

Normally, switching to the draft mode should not alter the geometry of your document. Since overlays are deactivated, any code placed there (e.g. counter changes) is not executed anymore! Also, /tcb/remember as→ P. 140 will not have any effect. You may exclude critical code with \tcbinterruptdraftmode / \tcbcontinuedraftmode from converting to draft mode.

\tcbstartdraftmode

Any following tcolorbox code is put into *draft mode*. All skin settings are overruled with spartan→ P. 186. Overlays, watermarks, shadows, borderlines, and rounded corners are deactivated for all tcolorbox layers.

\tcbstopdraftmode

The *draft mode* is deactivated for the following code.

\tcbinterruptdraftmode

If the compilation is in *draft mode*, the *draft mode* is deactivated until a following \tcbcontinuedraftmode is detected.

If the compilation is not in *draft mode*, nothing happens and a following \tcbcontinuedraftmode will not start the *draft mode*.

The pair \tcbinterruptdraftmode and \tcbcontinuedraftmode cannot be used nested.

\tcbcontinuedraftmode

Continues the *draft mode* which was suspended by a preceding \tcbinterruptdraftmode. Nothing happens, if there was no draft mode before \tcbinterruptdraftmode.

Code, which is place between \tcbinterruptdraftmode and \tcbcontinuedraftmode is shielded from *draft mode*.

/tcb/draftmode=true|false (default true, initially false)

If set to true, the *draft mode* is started. If set to false, the *draft mode* is stopped.

```
\newtcolorbox{mybeamer}[2][]{beamer,colback=Salmon!50!white,
  colframe=FireBrick!75!black,adjusted title={#2},#1}

\begin{mybeamer}{Beamer box}
This box looks like a box provided by the \texttt{beamer} class.
\end{mybeamer}\par\medskip
\begin{mybeamer}[draftmode]{Beamer box}
This box looks like a box provided by the \texttt{beamer} class.
\end{mybeamer}
```

Beamer box

This box looks like a box provided by the beamer class.

Beamer box

This box looks like a box provided by the beamer class.

7.12 Skin Family 'standard'

Note that the option keys /tcb/frame style→P. 101, /tcb/interior style→P. 102, /tcb/segmentation style→P. 104, and /tcb/title style→P. 104 are not be applicable to the standard skin. Also, watermarks (see Subsection 7.3) are not usable with the standard skin.

/tcb/skin=standard (skin)

This is the standard skin from the core package. All drawing engines are set to type `standard`. The drawing is based on `pgf` commands and does not need the `tikz` package.

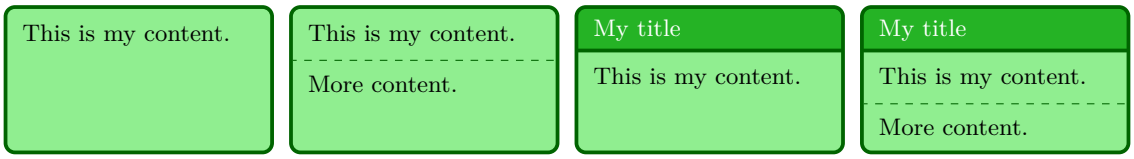
Environment and engines for the skin 'standard'

/tcb/graphical environment→P. 89:	pgfpicture
/tcb/frame engine→P. 90:	standard
/tcb/interior titled engine→P. 90:	standard
/tcb/interior engine→P. 91:	standard
/tcb/segmentation engine→P. 91:	standard
/tcb/title engine→P. 91:	standard

/tcb/standard (style, no value)

This is an abbreviation for setting `skin=standard`.

```
\tcbset{standard,equal height group=standard,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



/tcb/skin=standard jigsaw (skin)

This is the standard jigsaw skin from the core package. It differs from the skin standard → P. 148 by its frame engine, see Section 7.10 on page 145.

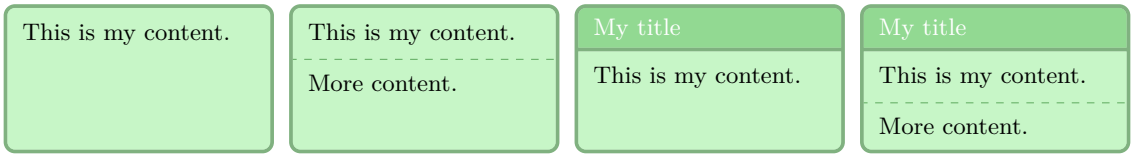
Environment and engines for the skin 'standard jigsaw'

/tcb/graphical environment → P. 89:	pgfpicture
/tcb/frame engine → P. 90:	standardjigsaw
/tcb/interior titled engine → P. 90:	standard
/tcb/interior engine → P. 91:	standard
/tcb/segmentation engine → P. 91:	standard
/tcb/title engine → P. 91:	standard

/tcb/standard jigsaw (style, no value)

This is an abbreviation for setting skin=standard jigsaw.

```
\tcbset{standard jigsaw,equal height group=standardjigsaw,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



7.13 Skin Family 'enhanced'

If you like the standard appearance of a `tcolorbox` but you want to have some 'enhanced' features, the `enhanced` skin is what you are looking for.

`/tcb/skin=enhanced` (skin)

This skin translates the drawing commands of the core package into `tikz` path commands. Therefore, it allows all `tikz` high level options for these paths and has more flexibility compared to the `standard` → P. 148 skin. You pay for this with some prolonged compilation time. The `tikz` path options can be given with the option keys `/tcb/frame style` → P. 101, `/tcb/interior style` → P. 102, `/tcb/segmentation style` → P. 104, and `/tcb/title style` → P. 104.

Environment and engines for the skin 'enhanced'

`/tcb/graphical environment` → P. 89:	`tikzpicture`
`/tcb/frame engine` → P. 90:	`path`
`/tcb/interior titled engine` → P. 90:	`path`
`/tcb/interior engine` → P. 91:	`path`
`/tcb/segmentation engine` → P. 91:	`path`
`/tcb/title engine` → P. 91:	`path`

`/tcb/enhanced` (style, no value)

This is an abbreviation for setting `skin=enhanced`.

```
\tcbset{enhanced,equal height group=enhanced,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

```
% \usetikzlibrary{shadings}          % preamble
\tcbset{skin=enhanced,fonttitle=\bfseries,
  frame style={upper left=blue,upper right=red,lower left=yellow,lower right=green},
  interior style={white,opacity=0.5},
  segmentation style={black,solid,opacity=0.2,line width=1pt}}

\begin{tcolorbox}[title=Nice box in rainbow colors]
  With the 'enhanced' skin, it is quite easy to produce fancy looking effects.
  \tcblower
  Note that this is still a \texttt{tcolorbox}.
\end{tcolorbox}
```

Nice box in rainbow colors

With the 'enhanced' skin, it is quite easy to produce fancy looking effects.

Note that this is still a `tcolorbox`.

```
% \usetikzlibrary{decorations.pathmorphing} % preamble
\tcbset{skin=enhanced,fonttitle=\bfseries,boxrule=1mm,
  frame style={draw=FireBrick,fill=Salmon},drop fuzzy shadow,
  interior style={draw=FireBrick,top color=Salmon!10,bottom color=Salmon!20},
  segmentation style={draw=FireBrick,solid,decorate,
        decoration={coil,aspect=0,segment length=10.1mm}}}

\begin{tcblisting}{title=A listing box with shadow and some specials}
Of course, skins can be used for listings also.
\begin{equation}
  \int\limits_1^2 \frac{1}{x}~dx = \ln(2).
\end{equation}
\end{tcblisting}
```

A listing box with shadow and some specials

```
Of course, skins can be used for listings also.
\begin{equation}
  \int\limits_1^2 \frac{1}{x}~dx = \ln(2).
\end{equation}
```

Of course, skins can be used for listings also.

$$\int_1^2 \frac{1}{x}\,dx = \ln(2). \tag{2}$$

/tcb/enhanced standard (style, no value)

For unbreakable boxes, this is identical to using /tcb/enhanced → P. 150. But, for breakable boxes, the *break sequence* is identical to the standard → P. 148 skin, see Section 11.6 from page 264.

/tcb/blank (style, initially unset)

This style relies on the skin enhanced → P. 150. All drawing operations are hidden and all margins are set to 0pt. See /tcb/blanker → P. 178 for switching off the drawing engines.

```
\begin{tcolorbox}[blank,watermark text=A blank box]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

A blank box

\tcbline

Sometimes, a line is only a line. With \tcblower → P. 9 you separate the box content into two functional units. \tcbline draws only a line which looks like the segmentation line between upper and lower part. Furthermore, you can use \tcbline more than just once. \tcbline always uses the path drawing engine. Therefore, the /tcb/segmentation style → P. 104 can be applied.

```
\tcbset{enhanced,colframe=blue!50!black,colback=white}

\begin{tcolorbox}[colupper=red!50!black,collower=green!50!black]
\lipsum[1]
\tcbline
\lipsum[2]
\tcblower
\lipsum[3]
\tcbline
\lipsum[4]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

\tcbline*

Equivalent to \tcbline, but in a breakable box, \tcbline* is removed if at a page/box break. Also, it is removed at the end of a box.

/tcb/skin=enhancedfirst (skin)

This is a flavor of enhanced → P. 150 which is used as a *first* part in a break sequence for enhanced → P. 150. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedfirst'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathfirst
/tcb/interior titled engine → P. 90:	pathfirst
/tcb/interior engine → P. 91:	pathfirst
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathfirst

```
\tcbset{skin=enhancedfirst,equal height group=enhancedfirst,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhancedmiddle (skin)

This is a flavor of enhanced → P. 150 which is used as a *middle* part in a break sequence for enhanced → P. 150. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedmiddle'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathmiddle
/tcb/interior titled engine → P. 90:	pathmiddle
/tcb/interior engine → P. 91:	pathmiddle
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathmiddle

```
\tcbset{skin=enhancedmiddle,equal height group=enhancedmiddle,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhancedlast (skin)

This is a flavor of enhanced → P. 150 which is used as a *last* part in a break sequence for enhanced → P. 150. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'enhancedlast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathlast
/tcb/interior titled engine → P. 90:	pathlast
/tcb/interior engine → P. 91:	pathlast
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathlast

```
\tcbset{skin=enhancedlast,equal height group=enhancedlast,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=enhanced jigsaw (skin)

This is the jigsaw variant of skin enhanced → P. 150. It differs by its frame engine, see Section 7.10 on page 145.

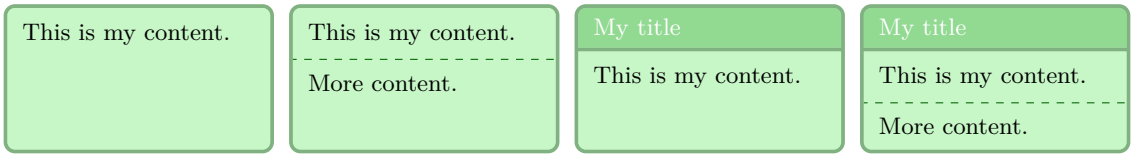
Environment and engines for the skin 'enhanced jigsaw'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathjigsaw
/tcb/interior titled engine → P. 90:	path
/tcb/interior engine → P. 91:	path
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	path

/tcb/enhanced jigsaw (style, no value)

This is an abbreviation for setting skin=enhanced jigsaw.

```
\tcbset{enhanced jigsaw,equal height group=enhancedjigsaw,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



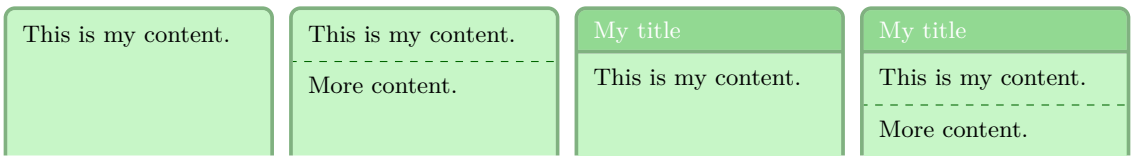
/tcb/skin=enhancedfirst jigsaw (skin)

This is the jigsaw variant of skin enhancedfirst → P. 154. It differs by its frame engine, see Section 7.10 on page 145.

Environment and engines for the skin 'enhancedfirst jigsaw'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathfirstjigsaw
/tcb/interior titled engine → P. 90:	pathfirst
/tcb/interior engine → P. 91:	pathfirst
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathfirst

```
\tcbset{skin=enhancedfirst jigsaw,equal height group=enhancedfirstjigsaw,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



/tcb/skin=enhancedmiddle jigsaw (skin)

This is the jigsaw variant of skin enhancedmiddle → P. 155. It differs by its frame engine, see Section 7.10 on page 145.

Environment and engines for the skin 'enhancedmiddle jigsaw'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathmiddlejigsaw
/tcb/interior titled engine → P. 90:	pathmiddle
/tcb/interior engine → P. 91:	pathmiddle
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathmiddle

```
\tcbset{skin=enhancedmiddle jigsaw,equal height group=enhancedmiddlejigsaw,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/marker (style, no value)

This styles relies on the skin enhancedmiddle jigsaw. It is intended to be used as an optical marker like a highlighter pen.

```
\begin{tcolorbox}[marker]
\lipsum[2]
\end{tcolorbox}
```

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This examples demonstrates the creation of several *text marker* environments based on enhancedmiddle → P. 155.

```
\tcbset{textmarker/.style={%
    skin=enhancedmiddle jigsaw,breakable,parbox=false,
    boxrule=0mm,leftrule=5mm,rightrule=5mm,boxsep=0mm,arc=0mm,outer arc=0mm,
    left=3mm,right=3mm,top=1mm,bottom=1mm,toptitle=1mm,bottomtitle=1mm,oversize}}

\newtcolorbox{yellow}{textmarker,colback=yellow!5!white,colframe=yellow}
\newtcolorbox{orange}{textmarker,colback=DarkOrange!5!white,
                      colframe=DarkOrange!75!yellow}
\newtcolorbox{red}{textmarker,colback=red!5!white,colframe=red}
\newtcolorbox{blue}{textmarker,colback=DeepSkyBlue!5!white,colframe=DeepSkyBlue}
\newtcolorbox{green}{textmarker,colback=Chartreuse!5!white,colframe=Chartreuse}
\newtcolorbox{rainbow}{textmarker,interior hidden,
  frame style={top color=blue,bottom color=red,middle color=green}}

\begin{yellow}
  \lipsum[1-3]
\end{yellow}

\begin{orange}
  \lipsum[4]
\end{orange}

\begin{red}
  \lipsum[5]
\end{red}

\begin{green}
  \lipsum[6]
\end{green}

\begin{blue}
  \lipsum[7]
\end{blue}

\begin{rainbow}
  \lipsum[8]
\end{rainbow}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

/tcb/skin=enhancedlast jigsaw (skin)

This is the jigsaw variant of skin enhancedlast → P. 156. It differs by its frame engine, see Section 7.10 on page 145.

Environment and engines for the skin 'enhancedlast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathlastjigsaw
/tcb/interior titled engine → P. 90:	pathlast
/tcb/interior engine → P. 91:	pathlast
/tcb/segmentation engine → P. 91:	path
/tcb/title engine → P. 91:	pathlast

```
\tcbset{skin=enhancedlast jigsaw,equal height group=enhancedlastjigsaw,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  colbacktitle=LimeGreen!75!DarkGreen,width=(\linewidth-6mm)/4,nobeforeafter,
  opacityframe=0.5,opacityback=0.5,opacitybacktitle=0.5,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

7.14 Skin Family 'bicolor'

/tcb/skin=bicolor (skin)

This skin is quite similar to the standard→P. 148 and enhanced→P. 150 skin. But instead of a segmentation line, the optional lower part of the box is filled with a different color or drawn with a different style.

Environment and engines for the skin 'bicolor'

/tcb/graphical environment→P. 89:	tikzpicture
/tcb/frame engine→P. 90:	path
/tcb/interior titled engine→P. 90:	*special*
/tcb/interior engine→P. 91:	*special*
/tcb/segmentation engine→P. 91:	*special*
/tcb/title engine→P. 91:	path

- The most basic usage of this skin is to set the background color of the lower part by /tcb/colbacklower→P. 164 and all other options like for the standard→P. 148 skin.

```
\begin{tcolorbox}[skin=bicolor,title=The title,
    colframe=FireBrick!75!black,colback=Salmon!50!white,colbacklower=Salmon]
  The upper part.
  \tcblower
  The lower part.
\end{tcolorbox}
```

The title

The upper part.

The lower part.

- The more advanced usage of this skin is to apply the /tcb/frame style→P. 101 and the /tcb/interior style→P. 102 like for the enhanced→P. 150 skin. Also, the /tcb/segmentation style→P. 104 can be used, but it is applied to the whole lower part.

```
\begin{tcolorbox}[skin=bicolor,title=The title,
    frame style={top color=FireBrick,
                 bottom color=FireBrick!15!white,draw=black},
    interior style={left color=Salmon,right color=Salmon!50!white},
    segmentation style={right color=Salmon,left color=Salmon!50!white}]
  The upper part.
  \tcblower
  The lower part.
\end{tcolorbox}
```

The title

The upper part.

The lower part.

/tcb/bicolor (style, no value)

This is an abbreviation for setting skin=bicolor.

```
\tcbset{bicolor,equal height group=bicolor,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/colbacklower=⟨*color*⟩ (no default, initially `black!15!white`)

Sets the background ⟨*color*⟩ of the lower part. It depends on the skin, if this value is used.

```
\tcbset{gitexample/.style={listing and comment,comment={#1},
  skin=bicolor,boxrule=1mm,fonttitle=\bfseries,coltitle=black,
  frame style={draw=black,left color=Gold,right color=Goldenrod!50!Gold},
  colback=black,colbacklower=Goldenrod!75!Gold,
  colupper=white,collower=black,
  listing options={language={bash},aboveskip=0pt,belowskip=0pt,nolol,
  basicstyle=\ttfamily\bfseries,extendedchars=true}}}

\begin{tcblisting}{title={Snapshot of the staging area},
  gitexample={The option '-a' automatically stages all tracked and modified
              files before the commit.\par
              This can be combined with the message option '-m'
              as seen in the third line.}}
git commit
git commit -a
git commit -am 'changes to my example'
\end{tcblisting}
```

Snapshot of the staging area

```
git commit
git commit -a
git commit -am 'changes to my example'
```

The option '-a' automatically stages all tracked and modified files before the commit.
This can be combined with the message option '-m' as seen in the third line.

/tcb/skin=bicolorfirst (skin)

This is a flavor of bicolor → P. 163 which is used as a *first* part in a break sequence for bicolor → P. 163. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolorfirst'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathfirst
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathfirst

```
\tcbset{skin=bicolorfirst,equal height group=bicolorfirst,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=bicolormiddle (skin)

This is a flavor of bicolor → P. 163 which is used as a *middle* part in a break sequence for bicolor → P. 163. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolormiddle'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathmiddle
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathmiddle

```
\tcbset{skin=bicolormiddle,equal height group=bicolormiddle,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

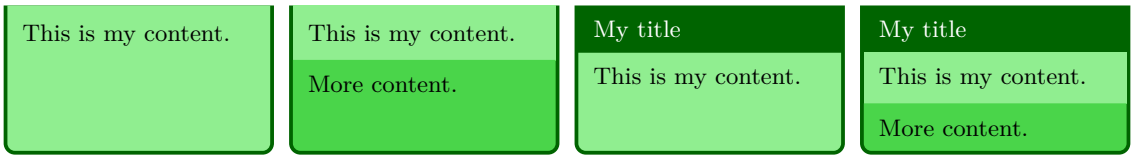
/tcb/skin=bicolorlast (skin)

This is a flavor of bicolor → P. 163 which is used as a *last* part in a break sequence for bicolor → P. 163. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'bicolorlast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathlast
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathlast

```
\tcbset{skin=bicolorlast,equal height group=bicolorlast,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



7.15 Skin Family 'beamer'

/tcb/skin=beamer (skin)

This skin resembles boxes known from the `beamer` class and therefore is called 'beamer'. It uses the normal colors from the core package but shades them a little bit. To use this skin, the `tikz` library `shadings` has to be included in the preamble by:

```
\usetikzlibrary{shadings}
```

The appearance of the skin can be controlled by /tcb/frame style → P. 101 and /tcb/interior style → P. 102, if needed. Here, the *segmentation* cannot be controlled by a style.

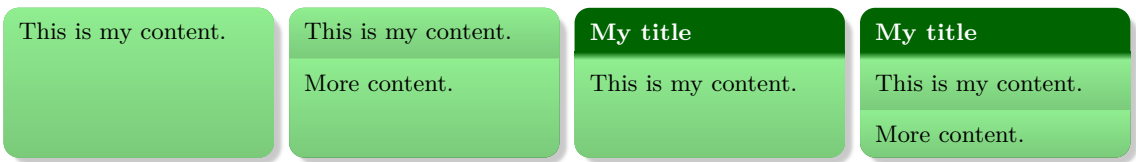
Environment and engines for the skin 'beamer'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	path
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	path

/tcb/beamer (style, no value)

This is an abbreviation for setting `skin=beamer`.
It also changes the geometry and some style options.

```
\tcbset{beamer,equal height group=beamer,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



```
\begin{tcolorbox}[beamer,colback=Salmon!50!white,colframe=FireBrick!75!black,
  adjusted title=A colored box with the 'beamer' skin]
This box looks like a box provided by the \texttt{beamer} class.
\end{tcolorbox}
```

A colored box with the 'beamer' skin

This box looks like a box provided by the `beamer` class.

```
\begin{tcolorbox}[beamer,colframe=blue,colback=black,
  watermark graphics=lichtspiel.jpg,
  coltext=white,watermark opacity=0.75,watermark stretch=1.0,
  title=Beamer Box with background picture]
\lipsum[1]
\end{tcolorbox}
```

Beamer Box with background picture

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
\newtcolorbox{myblock}[2][]{%
  beamer,breakable,colback=LightBlue,colframe=DarkBlue,#1,title=#2}%

\begin{myblock}{Beamerish \texttt{block}: \texttt{myblock}}
\lipsum[1]
\end{myblock}
```

Beamerish `block`: `myblock`

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

/tcb/skin=beamerfirst (skin)

This is a flavor of beamer → P. 168 which is used as a *first* part in a break sequence for beamer → P. 168. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamerfirst'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathfirst
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathfirst

```
\tcbset{beamer,skin=beamerfirst,equal height group=beamerfirst,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=beamermiddle (skin)

This is a flavor of beamer → P. 168 which is used as a *middle* part in a break sequence for beamer → P. 168. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamermiddle'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathmiddle
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathmiddle

```
\tcbset{beamer,skin=beamermiddle,equal height group=beamermiddle,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=beamerlast (skin)

This is a flavor of beamer → P. 168 which is used as a *last* part in a break sequence for beamer → P. 168. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'beamerlast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathlast
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	pathlast

```
\tcbset{beamer,skin=beamerlast,equal height group=beamerlast,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

7.16 Skin Family 'widget'

/tcb/skin=widget (skin)

This skin uses the normal colors from the core package but shades them a little bit. To use this skin, the `tikz` library `shadings` has to be included in the preamble by:

```
\usetikzlibrary{shadings}
```

The appearance of the skin can be controlled by /tcb/frame style→P. 101, /tcb/interior style→P. 102, and /tcb/segmentation style→P. 104, if needed.

Environment and engines for the skin 'widget'

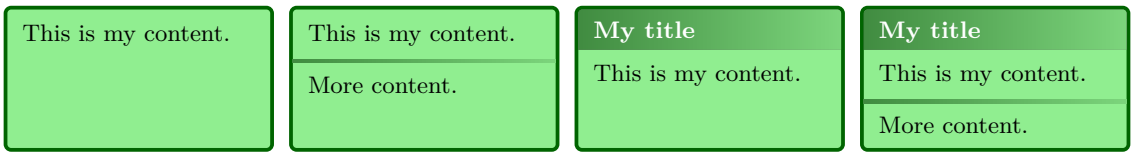
/tcb/graphical environment→P. 89:	tikzpicture
/tcb/frame engine→P. 90:	path
/tcb/interior titled engine→P. 90:	path
/tcb/interior engine→P. 91:	path
/tcb/segmentation engine→P. 91:	*special*
/tcb/title engine→P. 91:	*special*

/tcb/widget (style, no value)

This is an abbreviation for setting `skin=widget`.

It also changes the geometry and some style options.

```
\tcbset{widget,equal height group=widget,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



```
\begin{tcolorbox}[widget,colback=Salmon!50!white,colframe=FireBrick!75!black,
  adjusted title=A colored box with the 'widget' skin]
This is my content.
\end{tcolorbox}
```

A colored box with the 'widget' skin

This is my content.

/tcb/skin=widgetfirst (skin)

This is a flavor of widget → P. 173 which is used as a *first* part in a break sequence for widget → P. 173. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetfirst'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathfirst
/tcb/interior titled engine → P. 90:	pathfirst
/tcb/interior engine → P. 91:	pathfirst
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	*special*

```
\tcbset{widget,skin=widgetfirst,equal height group=widgetfirst,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=widgetmiddle (skin)

This is a flavor of widget → P. 173 which is used as a *middle* part in a break sequence for widget → P. 173. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetmiddle'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathmiddle
/tcb/interior titled engine → P. 90:	pathmiddle
/tcb/interior engine → P. 91:	pathmiddle
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	*special*

```
\tcbset{widget,skin=widgetmiddle,equal height group=widgetmiddle,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=widgetlast (skin)

This is a flavor of widget → P. 173 which is used as a *last* part in a break sequence for widget → P. 173. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'widgetlast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	pathlast
/tcb/interior titled engine → P. 90:	pathlast
/tcb/interior engine → P. 91:	pathlast
/tcb/segmentation engine → P. 91:	*special*
/tcb/title engine → P. 91:	*special*

```
\tcbset{widget,skin=widgetlast,equal height group=widgetlast,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

7.17 Skin Family 'empty'

/tcb/skin=empty (skin)

This skin sets all engines to `empty`, i. e. nothing is drawn at all. Therefore, this skin is a good starting point to create a complete new style by yourself.

Environment and engines for the skin 'empty'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	empty
/tcb/interior titled engine → P. 90:	empty
/tcb/interior engine → P. 91:	empty
/tcb/segmentation engine → P. 91:	empty
/tcb/title engine → P. 91:	empty

/tcb/empty (style, no value)

This is an abbreviation for setting `skin=empty`.

```
\tcbset{empty,equal height group=empty,
  coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/blanker (style, initially unset)

This style relies on the skin empty → P. 177. All engines are set to empty and all margins are set to 0pt. In contrast to /tcb/blank → P. 152, the graphical paths are not constructed with exception of the geometry nodes.

```
\begin{tcolorbox}[blanker,watermark text=A blank box]
\lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

A blank box

```
% \tcbuselibrary{fitting}
\newtcboxfit{\mybox}[1]{blanker,width=4cm,height=7cm,top=4pt,
  watermark text=#1}

\begin{tabular}{|c|c|c|}\hline
A & B & C\\\hline
\mybox{A}{\lipsum[1]} & \mybox{B}{\lipsum[2]} & \mybox{C}{\lipsum[3]}\\\hline
\end{tabular}
```

A	B	C
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.	Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.	Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

/tcb/skin=emptyfirst (skin)

This is a flavor of empty → P. 177 which is used as a *first* part in a break sequence for empty → P. 177. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptyfirst'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	empty
/tcb/interior titled engine → P. 90:	empty
/tcb/interior engine → P. 91:	empty
/tcb/segmentation engine → P. 91:	empty
/tcb/title engine → P. 91:	empty

```
\tcbset{empty,skin=emptyfirst,equal height group=emptyfirst,
  coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=emptymiddle (skin)

This is a flavor of empty → P. 177 which is used as a *middle* part in a break sequence for empty → P. 177. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptymiddle'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	empty
/tcb/interior titled engine → P. 90:	empty
/tcb/interior engine → P. 91:	empty
/tcb/segmentation engine → P. 91:	empty
/tcb/title engine → P. 91:	empty

```
\tcbset{empty,skin=emptymiddle,equal height group=emptymiddle,
  coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

/tcb/skin=emptylast (skin)

This is a flavor of empty → P. 177 which is used as a *last* part in a break sequence for empty → P. 177. Nevertheless, this skin can be applied independently.

Environment and engines for the skin 'emptylast'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	empty
/tcb/interior titled engine → P. 90:	empty
/tcb/interior engine → P. 91:	empty
/tcb/segmentation engine → P. 91:	empty
/tcb/title engine → P. 91:	empty

```
\tcbset{empty,skin=emptylast,equal height group=emptylast,
  coltitle=Navy,borderline={2pt}{0pt}{black!10!white},
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

This example demonstrates a breakable customized box. Here, we define an environment `freebox`. The first application of `freebox` produces an unbroken `tcolorbox`. The box is drawn by the code given by `/tcb/frame code` → P. 93 and `/tcb/interior code` → P. 94. The second application of `freebox` is broken into several parts which are drawn by the codes given by `/tcb/skin first is subskin of` → P. 96, `/tcb/skin middle is subskin of` → P. 96, and `/tcb/skin last is subskin of` → P. 96.

```
% Preamble:
%\usepackage{tikz,lipsum}
%\tcbuselibrary{skins,breakable}
\tikzset{coltria/.style={fill=red!15!white}}

\newtcolorbox{freebox}[1][]{empty,breakable,leftrule=5mm,left=2mm,
  frame style={fill,top color=red!75!black,bottom color=red!75!black,middle color=red},
  colback=yellow!50!white,
  watermark color=red!50!yellow!75!white,
  watermark text on=unbroken is unbroken box,
  watermark text on=first is first part,
  watermark text on=middle is middle part,
  watermark text on=last is last part,
  % code for unbroken boxes:
  frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
    --([xshift=-5mm]frame.north east)--([yshift=-5mm]frame.north east)
    --([yshift=5mm]frame.south east)--([xshift=-5mm]frame.south east)--cycle; },
  interior code={\path[tcb fill interior] (interior.south west)--(interior.north west)
    --([xshift=-4.8mm]interior.north east)--([yshift=-4.8mm]interior.north east)
    --([yshift=4.8mm]interior.south east)--([xshift=-4.8mm]interior.south east)
    --cycle; },
  % code for the first part of a break sequence:
  skin first is subskin of={emptyfirst}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --([xshift=-5mm]frame.north east)--([yshift=-5mm]frame.north east)
      --(frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=1mm]frame.south west) -- +(120:2mm)
      -- +(60:2mm)-- cycle; },
    interior code={\path[tcb fill interior] (interior.south west|-frame.south)
      --(interior.north west)--([xshift=-4.8mm]interior.north east)
      --([yshift=-4.8mm]interior.north east)--(interior.south east|-frame.south)
      --cycle; },
  },%
  % code for the middle part of a break sequence:
  skin middle is subskin of={emptymiddle}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --(frame.north east)--(frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=-1mm]frame.north west) -- +(240:2mm)
        -- +(300:2mm) -- cycle;
      \path[coltria] ([xshift=2.5mm,yshift=1mm]frame.south west) -- +(120:2mm)
        -- +(60:2mm) -- cycle;
      },
    interior code={\path[tcb fill interior] (interior.south west|-frame.south)
      --(interior.north west|-frame.north)--(interior.north east|-frame.north)
      --(interior.south east|-frame.south)--cycle; },
    },
  % code for the last part of a break sequence:
  skin last is subskin of={emptylast}{%
    frame code={\path[tcb fill frame] (frame.south west)--(frame.north west)
      --(frame.north east)--([yshift=5mm]frame.south east)
      --([xshift=-5mm]frame.south east)--cycle;
      \path[coltria] ([xshift=2.5mm,yshift=-1mm]frame.north west) -- +(240:2mm)
      -- +(300:2mm) -- cycle;
      },
    interior code={\path[tcb fill interior] (interior.south west)
```

```
      --(interior.north west|-frame.north)--(interior.north east|-frame.north)
      --([yshift=4.8mm]interior.south east)--([xshift=-4.8mm]interior.south east)
      --cycle; },
    },
  #1}

\begin{freebox}
\lipsum[1]
\end{freebox}

\begin{freebox}
\lipsum[1-12]
\end{freebox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

unbroken box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

first part

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim.

Vestibulum pellentesque felis eu massa.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.
Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.
Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.
Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus. Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et,

middle part

lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

last part

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

7.18 Skin 'spartan'

/tcb/skin=spartan (skin)

This skin is quite ... spartan. It supports no rounded corners, no overlays, no shadows, no borderlines, and no finishes. The only exception are underlays. One cannot do very fancy things with this skin, but it compiles very fast. Therefore, the `spartan` skin is used for the draft mode, see Section 7.11 on page 147. Nevertheless, it can be used as a normal skin.

Environment and engines for the skin 'spartan'

/tcb/graphical environment → P. 89:	tikzpicture
/tcb/frame engine → P. 90:	spartan
/tcb/interior titled engine → P. 90:	spartan
/tcb/interior engine → P. 91:	spartan
/tcb/segmentation engine → P. 91:	spartan
/tcb/title engine → P. 91:	spartan

/tcb/spartan (style, no value)

This is an abbreviation for setting `skin=spartan`.

```
\tcbset{spartan,equal height group=spartan,
  colback=LightGreen,colframe=DarkGreen,colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```

This is my content.

This is my content.

More content.

My title

This is my content.

My title

This is my content.

More content.

7.19 Skin 'draft'

/tcb/skin=draft (skin)

This skin is intended to be used while drafting new geometric settings for a `tcolorbox`.

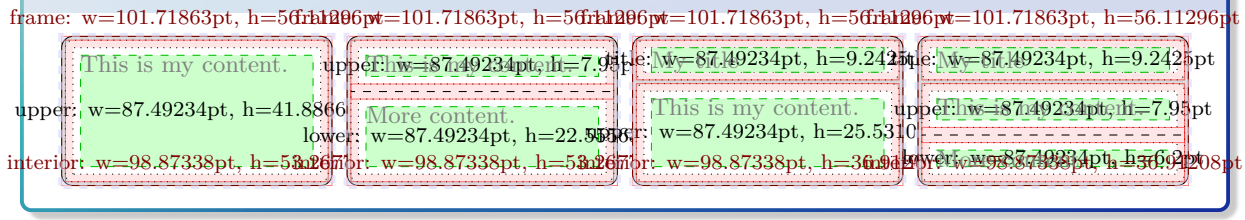
Environment and engines for the skin 'draft'

/tcb/graphical environment → P. 89:	`tikzpicture`
/tcb/frame engine → P. 90:	*special*
/tcb/interior titled engine → P. 90:	*special*
/tcb/interior engine → P. 91:	*special*
/tcb/segmentation engine → P. 91:	`path`
/tcb/title engine → P. 91:	`path`

/tcb/draft (style, no value)

This is an abbreviation for setting `skin=draft`.

```
\tcbset{draft,equal height group=draft,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/4,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}\hfill
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
```



```
\vspace*{3mm}
\begin{tcolorbox}[draft,title=A colored box with the 'draft' skin]
\lipsum[1-3]
\tcblower
\lipsum[4-6]
\end{tcolorbox}
```

frame: w=423.94617pt, h=506.20157pt

A colored box with the 'draft' skin

title: w=392.64822pt, h=6.2pt

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

upper: w=392.64822pt, h=249.95pt

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

lower: w=392.64822pt, h=205.95pt

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

interior: w=421.10092pt, h=490.04318pt

7.20 Skin Family 'freelance'

This skin family 'freelance' is deprecated with `tcolorbox` 3.00. It is not longer needed, because `/tcb/frame code` → P. 93, `/tcb/interior code` → P. 94, `/tcb/interior titled code` → P. 93, and `/tcb/title code` → P. 95 can be applied to every skin now. In this sense, everything has become *freelance* now.

For users of `/tcb/freelance`: Old code should continue to work. There may be exceptions for breakable freelance boxes under certain circumstances. For new code, use `/tcb/empty` → P. 177 or `/tcb/enhanced` → P. 150 where you would have used `/tcb/freelance` before.

`/tcb/skin=freelance` (skin)

This skin gives full freedom for the appearance of the `tcolorbox`. All drawing engines are set to type `freelance`; they use the `tikz` package and compute the `/tcb/geometry nodes` → P. 92.

Environment and engines for the skin 'freelance'

`/tcb/graphical environment` → P. 89:	`tikzpicture`
`/tcb/frame engine` → P. 90:	`freelance`
`/tcb/interior titled engine` → P. 90:	`freelance`
`/tcb/interior engine` → P. 91:	`freelance`
`/tcb/segmentation engine` → P. 91:	`freelance`
`/tcb/title engine` → P. 91:	`freelance`

`/tcb/freelance` (style, no value)

This is an abbreviation for setting `skin=freelance`.

`/tcb/skin=freelancefirst` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 11.6 on page 264.

`/tcb/skin=freelancemiddle` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 11.6 on page 264.

`/tcb/skin=freelancelast` (skin)

This skin equals `freelance` with exception of the break sequence, see Section 11.6 on page 264.

`/tcb/extend freelance=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelance`.

`/tcb/extend freelancefirst=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancefirst` which is used as first part of the break sequence of `freelance`. See `/tcb/skin first is subskin of` → P. 96 for a substitute of this key.

`/tcb/extend freelancemiddle=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancemiddle` which is used as middle part of the break sequence of `freelance`. See `/tcb/skin middle is subskin of` → P. 96 for a substitute of this key.

`/tcb/extend freelancelast=`⟨*options*⟩ (no default, initially empty)

The ⟨*options*⟩ are added to the skin definition of `freelancelast` which is used as last part of the break sequence of `freelance`. See `/tcb/skin last is subskin of` → P. 96 for a substitute of this key.

8 TikZ Image and Picture Fill Extensions; Auxiliary Macros

The LIB skins library adds some image and picture fill options to the vast option set of TikZ [20]. These options can be used in any `tikzpicture`. For the following options, the LIB skins library has to be loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

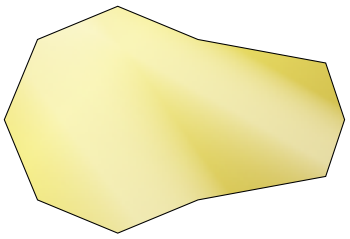
See Section 7 on page 101 for the documentation of all other options of the LIB skins library.

8.1 Fill Plain

`/tikz/fill plain image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area.

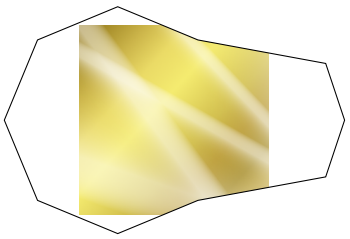
```
\begin{tikzpicture}
\path[draw,fill plain image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command.

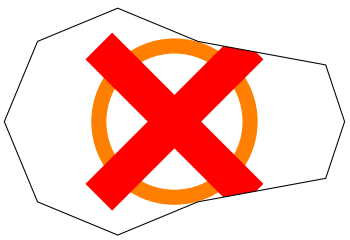
```
\begin{tikzpicture}
\path[draw,fill plain ↩
      ↪image*={width=2.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is put in the center of the path, but it is not resized to fit into the path area. Note that this is almost identical to the standard `path picture` option.

```
\begin{tikzpicture}
\path[draw,fill plain picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

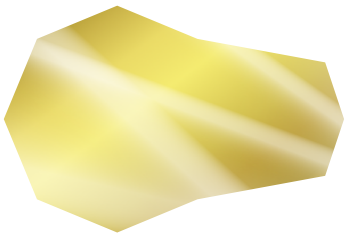


8.2 Fill Stretch

/tikz/fill stretch image=⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is stretched to fill the path area.

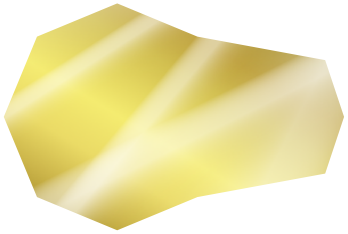
```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill stretch image*={⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying **\includegraphics** command. The image is stretched to fill the path area.

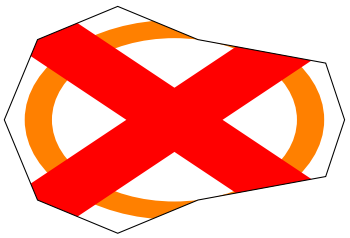
```
\begin{tikzpicture}
\path[fill stretch image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill stretch picture=⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is stretched to fill the path area.

```
\begin{tikzpicture}
\path[draw,fill stretch picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

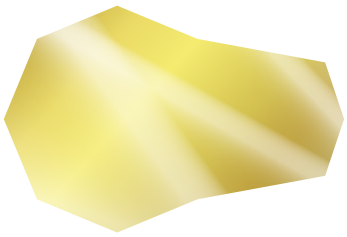


8.3 Fill Overzoom

`/tikz/fill overzoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that the path area fills the image.

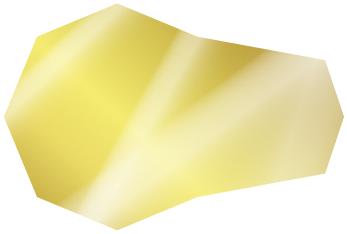
```
\begin{tikzpicture}
\path[fill overzoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that the path area fills the image.

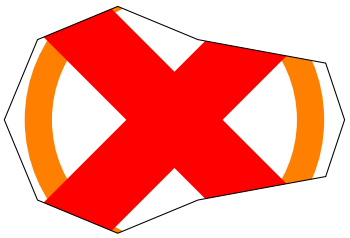
```
\begin{tikzpicture}
\path[fill overzoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that the path area fills the image.

```
\begin{tikzpicture}
\path[draw,fill overzoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

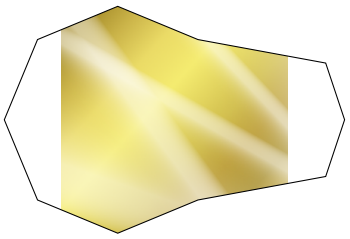


8.4 Fill Zoom

`/tikz/fill zoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

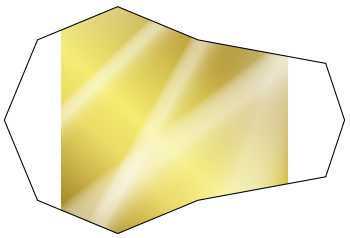
```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

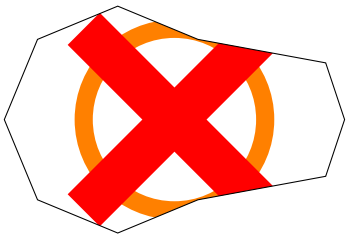
```
\begin{tikzpicture}
\path[draw,fill zoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill zoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

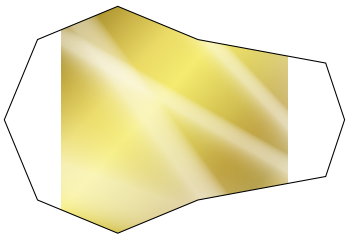


8.5 Fill Shrink

`/tikz/fill shrink image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

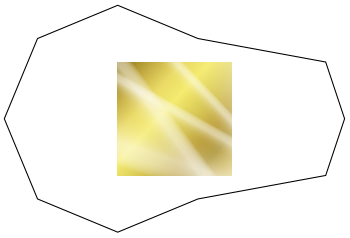
```
\begin{tikzpicture}
\path[draw,fill shrink image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

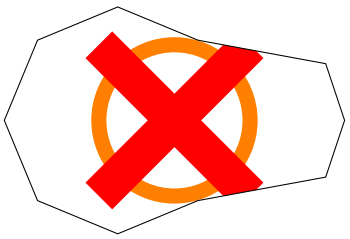
```
\begin{tikzpicture}
\path[draw,fill shrink ⤸
      ⤹image*={width=1.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill shrink picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



8.6 Fill Tile

/tikz/fill tile image=⟨*file name*⟩ (no default, initially unset)

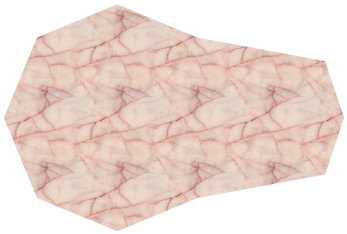
Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩.

```
\begin{tikzpicture}
\path[fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

/tikz/fill tile image*={⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying **\includegraphics** command.

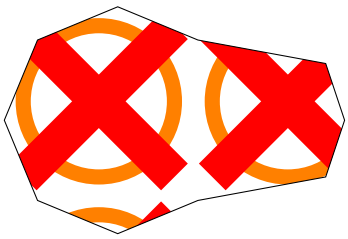
```
\begin{tikzpicture}
\path[fill tile image*={width=1cm}{pink_marble.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill tile picture=⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩.

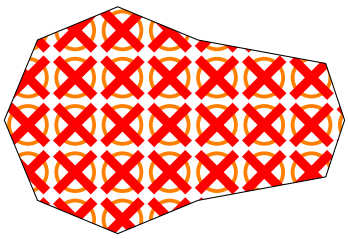
```
\begin{tikzpicture}
\path[draw,fill tile picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



/tikz/fill tile picture*={⟨*fraction*⟩}{⟨*graphical code*⟩} (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩. The graphic is resized by ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[draw,fill tile picture*={0.25}{%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (1cm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

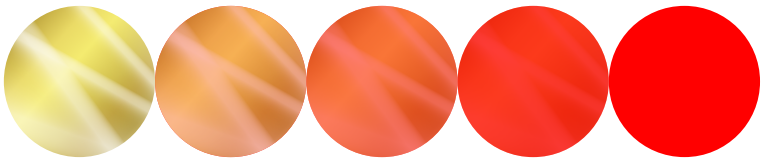


8.7 Filling Options

`/tikz/fill image opacity`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the fill opacity for the image or picture fill options to the given ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png] (0,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.75]
  (2,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.5]
  (4,0) circle (1cm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.25]
  (6,0) circle (1cm);
\path[fill=red] (8,0) circle (1cm);
\end{tikzpicture}
```



`/tikz/fill image scale`=⟨*fraction*⟩ (no default, initially `1.0`)

Stretches, zooms, overzooms or shrinks the image or picture to the given ⟨*fraction*⟩ of the width and height of the current path.

```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (0,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=0.75]
  (3,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=1.5]
  (6,0) rectangle +(2,2);
\end{tikzpicture}
```

`/tikz/fill image options`=⟨*graphics options*⟩ (no default, initially empty)

The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command for the image fill options. This can be just together with `/tikz/fill stretch image` → P. 191, `/tikz/fill overzoom image` → P. 192, `/tikz/fill zoom image` → P. 193, and `/tikz/fill tile image` → P. 195.

```
\begin{tikzpicture}
\path[fill image options={width=1cm},
  fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

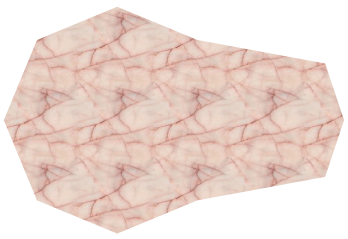
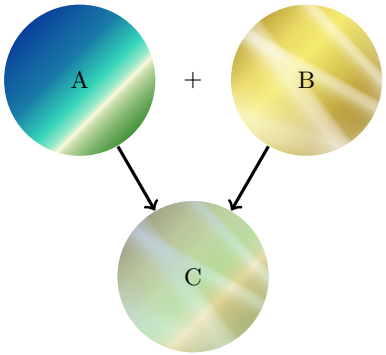


Image blending example

```
\begin{tikzpicture}[every node/.style=
  {circle,minimum width=2cm}]
\node[fill stretch image=blueshade.png]
  (A) at (120:3cm) {A};
\node[fill stretch image=goldshade.png]
  (B) at (60:3cm) {B};
\node[
  preaction={fill stretch image=blueshade.png},
  fill stretch image=goldshade.png,
  fill image opacity=0.5] (C) {C};
\path (A) -- node{$+$} (B);
\draw[->,very thick] (A)--(C);
\draw[->,very thick] (B)--(C);
\end{tikzpicture}
```



8.8 Extracting Node Dimensions

The following auxiliary macros are defined by the LIB skins library. They allow to determine the width and height of an arbitrary TikZ node. To be more specific, they determine the east-to-west and the north-to-south dimensions which may be not the maximal dimensions for a non-rectangular node.

N 2014-09-19 `\tcbsettowidthofnode{⟨register⟩}{⟨node⟩}`

Sets the east-to-west dimension of the given ⟨*node*⟩ to the TEX ⟨*register*⟩.

N 2014-09-19 `\tcbsetmacrotowidthofnode{⟨macro⟩}{⟨node⟩}`

Defines ⟨*macro*⟩ as the east-to-west dimension of the given ⟨*node*⟩.

N 2014-09-19 `\tcbsettoheightofnode{⟨register⟩}{⟨node⟩}`

Sets the north-to-south dimension of the given ⟨*node*⟩ to the TEX ⟨*register*⟩.

N 2014-09-19 `\tcbsetmacrotoheightofnode{⟨macro⟩}{⟨node⟩}`

Defines ⟨*macro*⟩ as the north-to-south dimension of the given ⟨*node*⟩.

```
\begin{tikzpicture}
  \node[align=center,draw=red,fill=yellow] (A) {This is my\\example node};
  \tcbsetmacrotowidthofnode\mywidth{A}
  \tcbsetmacrotoheightofnode\myheight{A}
  \path[draw=blue,fill=blue!25!white] ([xshift=2mm]A.south east)
    rectangle node{Copy} +(\mywidth,\myheight);
\end{tikzpicture}
```

9 Libraries LIB listings, LIB listingsutf8, and LIB minted

9.1 Loading the Libraries

In contrast to other `tcolorbox` libraries, the libraries LIB listings, LIB listingsutf8, and LIB minted are concurrent in the sense that they all do the same thing, i. e. displaying listings with or without typesetting the listing in LaTeX parallel. The difference is the underlying LaTeX package which does the core job for displaying a listing. So, typically, you need just *one* of these libraries. If you do not have a clue, which one of them you should use, you should take LIB listingsutf8.

The order in which the libraries are included influences the default settings and the `/tcb/reset` → P. 81 behavior. The settings of a later loaded library overwrite the settings of a previous loaded library. A library is never loaded twice.

9.1.1 Loading LIB listings

This library uses the package `listings` [6] to typeset listings. It is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{listings}
```

This also loads the package `listings` [6].

The `/tcb/listing engine` → P. 211 is set to `listings` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=listings}
```

9.1.2 Loading LIB listingsutf8

To extend `listings` for UTF-8 encoded sources, you can use the support from the package `listingsutf8` [10] by loading the library variant LIB listingsutf8.

```
\tcbuselibrary{listingsutf8}
\tcbset{listing utf8=latin1}% optional; 'latin1' is the default.
```

This also loads the library LIB listings and the packages `listings` [6] and `listingsutf8` [10].

The `/tcb/listing engine` → P. 211 is set to `listings` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=listings}
```

9.1.3 Loading LIB minted

This library uses the package `minted` [14] to typeset listings. It is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{minted}
```

This also loads the package `minted` [14].

The `minted` package uses the external tool `Pygments` [12] to apply syntax highlighting. It has to be installed and set up, before the library can be used, see [14] and [12]. The `tcolorbox` library LIB minted does not work, if the package `minted` [14] does not work.

The `/tcb/listing engine` → P. 211 is set to `minted` by the library. To reactivate this setting, if overwritten by other libraries, use

```
\tcbset{listing engine=minted}
```

9.2 Common Macros of the Libraries

`\begin{tcblisting}{`⟨*options*⟩`}`
⟨*environment content*⟩
`\end{tcblisting}`

Creates a colored box based on a `tcolorbox` → P. 9. Controlled by the given ⟨*options*⟩, the environment content is typeset normally and/or as a listing. Furthermore, the ⟨*options*⟩ control appearance and functions of the `tcolorbox`. By default, the listing is interpreted as a LaTeX listing.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{tcblisting}
```

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
% \usetikzlibrary{listings} /or/ \usetikzlibrary{listingsutf8}
\begin{tcblisting}{colback=yellow!5,colframe=yellow!50!black,listing only,
    title=This is source code in another language (XML), fonttitle=\bfseries,
  listing options={language=XML,columns=fullflexible,keywordstyle=\color{red}}}
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
\end{tcblisting}
```

This is source code in another language (XML)

```
<?xml version="1.0"?>
<project name="Package␣tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
```

```
% \usetikzlibrary{minted}
\begin{tcblisting}{colback=yellow!5,colframe=yellow!50!black,listing only,
    title=This is source code in another language (XML), fonttitle=\bfseries,
  listing engine=minted,minted language=xml}
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
\end{tcblisting}
```

This is source code in another language (XML)

```
<?xml version="1.0"?>
<project name="Package tcolorbox" default="documentation" basedir=".">
  <description>
    Apache Ant build file (http://ant.apache.org/)
  </description>
</project>
```

```
% This box is as wide as needed (listing only !!)
% \usetikzlibrary{skins}
\begin{tcblisting}{colback=green!5!white,colframe=green!50!black,listing only,
  hbox,enhanced,drop fuzzy shadow,before=\begin{center},after=\end{center}}
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
\end{tcblisting}
```

```
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\begin{tcboutputlisting}
⟨*environment content*⟩
\end{tcboutputlisting}

Saves the environment content to a file which is named by the key value of `listing file`. Later, this file can be loaded by `\tcbinputlisting` or `\tcbuselistingtext` or `\tcbuselistinglisting`.

```
\begin{tcboutputlisting}
This \textbf{text} is written to a standardized file for later usage.
\end{tcboutputlisting}
```

\tcbinputlisting{⟨*options*⟩}

Creates a colored boxed based on a `tcolorbox`. The text content is read from a file named by the key value of `listing file`. Apart from that, the function is equal to that of `tcblisting` → P. 200.

```
\tcbinputlisting{colback=red!5!white,colframe=red!75!black,text only}
\tcbinputlisting{colback=green!5,colframe=green!75!black,listing only}
```

```
\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\tcbuselistingtext

Loads text from a file named by the key value of `listing file`.

```
\tcbuselistingtext
```

\tcbuselistinglisting

Typesets text as listing from a file named by the key value of `listing file`.

```
\tcbuselistinglisting

\begin{tikzpicture}
\fill[red] (0,0) rectangle (1,1);
\end{tikzpicture}
```

\tcbusetemplisting

Typesets text as listing from a temporary file which was written by `tcbwritetemp` → P. 88.

See Section 14.4 on page 301 and Section 14.5 on page 303 for more elaborate methods to create new environments and commands.

If a new sort of `tcblisting` environments should be created with one optional argument only, one is highly recommended to use `\DeclareTCBListing` → P. 301 or `\NewTCBListing` → P. 301 instead of `\newtcblisting` to avoid content scanning problems.

`\newtcblisting[⟨init options⟩]{⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new environment ⟨*name*⟩ based on `tcblisting` → P. 200. Basically, `\newtcblisting` operates like `\newenvironment`. 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 `tcblisting`. Note that `/tcb/savedelimiter` → P. 19 is set to the given ⟨*name*⟩ automatically. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

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

\begin{mybox}
This is my \LaTeX\ box.
\end{mybox}
```

This is my \LaTeX\ box.

This is my LaTeX box.

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

\begin{mybox}{Listing Box}
This is my \LaTeX\ box.
\end{mybox}
```

Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

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

\begin{mybox}[listing only]
  {Listing Box}
This is my \LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}[listing side text]
  {Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
```

Listing Box

This is my \LaTeX\ box.

Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

Definition in the preamble:

```
\newtcblisting[auto counter]{mycbox}[1]{%
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  title=Listing \thetcbcounter: #1}
```

```
\begin{mycbox}{Listing Box}
This is my \LaTeX\ box.
\end{mycbox}
```

Listing 1: Listing Box

This is my \LaTeX\ box.

This is my LaTeX box.

`\renewtcblisting`[⟨*init options*⟩]{⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like `\newtcblisting` → P. 203, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

\newtcbinputlisting[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Creates a new macro \⟨*name*⟩ based on \tcbinputlisting → P. 202. Basically, \newtcbinputlisting operates like \newcommand. 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 tcbinputlisting. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

Listing (2) of tcolorbox.tcbtemp

This is the included file content:

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing engine=minted,minted language=latex,minted style=colorful,
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

Listing (3) of tcolorbox.tcbtemp

This is the included file content:

```
\newtcbinputlisting[use counter from=mycbox]{\mylisting}[2][]{%
  listing engine=minted,minted language=latex,minted style=colorful,
  listing file={#2},
  title=Listing (\thetcbcounter) of \texttt{#2},
  colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  listing only,breakable,#1}

\mylisting[before upper=\textit{This is the included file content:}]
          {\jobname.tcbtemp}
```

\renewtcbinputlisting[⟨*init options*⟩]{\⟨*name*⟩}[⟨*number*⟩][⟨*default*⟩]{⟨*options*⟩}

Operates like \newtcbinputlisting, but based on \renewcommand instead of \newcommand. An existing macro is redefined.

9.3 Option Keys of the LIB listings Library

/tcb/listing options=⟨*key list*⟩ (no default, initially `style=tcblatex`)

Sets the options from the package `listings` [6] which are used during typesetting of the listing. For LaTeX listings, there is a predefined `listings` style named `tcblatex` which can be used.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!25,left=6mm,
listing options={style=tcblatex,numbers=left,numberstyle=\tiny\color{red!75!black}}}
This is a \LaTeX\ example which displays the text as source code
and in compiled form. Additionally, we use line numbers here.
\end{tcblisting}
```

```
1 This is a \LaTeX\ example which displays the text as source code
2 and in compiled form. Additionally, we use line numbers here.
```

This is a LaTeX example which displays the text as source code and in compiled form. Additionally, we use line numbers here.

/tcb/no listing options (no value, initially unset)

Abbreviation for `listing options={}`. This removes all options for the `listings` package. This includes the `tcblisting` standard style `tcblatex` and the encoding presets. Use this option, if you want to set the `listings` options outside of `tcblisting`, e. g. globally in the preamble.

```
\begin{tcblisting}{no listing options}
All \textit{listings} options removed.
\end{tcblisting}
```

```
All \textit{listings} options removed.
```

All *listings* options removed.

/tcb/listing style=⟨*style*⟩ (no default, initially `tcblatex`)

Abbreviation for `listing options={style=...}`. This key sets a ⟨*style*⟩ for the `listings` package, see [6]. For LaTeX, there is a predefined style named `tcblatex`.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,
listing style=tcblatex}
Here, we use the predefined style.
\end{tcblisting}
```

```
Here, we use the predefined style.
```

Here, we use the predefined style.

`/tcb/listing inputencoding=`⟨*encoding*⟩ (no default, initially `\inputencodingname`)

Sets the input encoding value for the predefined listing style `tcblatex` and `tcbdocumentation` from the library LIB documentation. The initial value is derived from the package `inputenc` if used.

`/tcb/listing remove caption=true|false` (default `true`, initially `true`)

If set to `true`, some part of the caption building code of the `listings` package is silenced to prevent some unwanted interaction with the `hyperref` package resulting in additional vertical space. If set to `false`, the `listings` package code is kept unchanged. Note that listings outside `tcblisting` → P. 200 and `\tcbinputlisting` → P. 202 are always processed normally. Typically, a user is not expected to use this key at all.

`/tcb/every listing line=`⟨*text*⟩ (no default, initially unset/empty)

Inserts some ⟨*text*⟩ to the begin of every line of a listing. Note that this a hack of the `listings` package code. This may become unusable or superfluous in the future.

```
\newtcblisting{commandshell}{colback=black,colupper=white,colframe=yellow!75!black,
  listing only,listing options={style=tcblatex,language=sh},
  every listing line={\textcolor{red}{\small\ttfamily\bfseries root \$> }}}

\begin{commandshell}
ls -al
cd /usr/lib
\end{commandshell}
```

```
root $> ls -al
root $> cd /usr/lib
```

`/tcb/every listing line*=`⟨*text*⟩ (no default, initially unset/empty)

Identical to `/tcb/every listing line` plus additional enlargement of `/tcb/rightupper` → P. 27 by the width of ⟨*text*⟩. Therefore, this option has to be used after the geometry settings are done. This option is intended to be used in conjunction with `/tcb/hbox` → P. 76.

```
\newtcblisting{commandshell}{colback=black,colupper=white,colframe=yellow!75!black,
  listing only,listing options={style=tcblatex,language=sh},hbox,
  every listing line*={\textcolor{red}{\small\ttfamily\bfseries root \$> }}}

\begin{commandshell}
ls -al
cd /usr/lib
\end{commandshell}
```

```
root $> ls -al
root $> cd /usr/lib
```

See further options in Section 9.6 on page 211.

For an combined example of using `\lstinline` inside a `tcolorbox`, see `\DeclareTotalTCBox` → P. 299.

9.4 Option Keys of the LIB listingsutf8 Library

The LIB listingsutf8 library is an extension of the LIB listings library, so all options from Section 9.3 on page 206 are applicable.

`/tcb/listing utf8`=⟨*one-byte-encoding*⟩ (style, no default, initially `latin1`)
Abbreviation for using `/tcb/listing inputencoding` → P. 207 together with UTF-8 support from the package `listingsutf8` [10]. This option is available only for the library variant LIB listingsutf8. The ⟨*one-byte-encoding*⟩ is one of the applicable encodings from [10], e. g. `latin1`.

See further options in Section 9.6 on page 211.

9.5 Option Keys of the LIB minted Library

/tcb/minted language=⟨*programming language*⟩ (no default, initially `latex`)

Sets a ⟨*programming language*⟩ known to `Pygments` [12].

```
\begin{tcblisting}{listing engine=minted,minted style=trac,
  minted language=java,
  colback=red!5!white,colframe=red!75!black,listing only}
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
\end{tcblisting}
```

```
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

/tcb/minted options=⟨*key list*⟩ (no default, initially `tabsize=2,fontsize=\small`)

Sets the options from the package `minted` [14] which are used during typesetting of the listing.

```
% \tcbuselibrary{skins}
\newtcblisting{myjava}{listing engine=minted,minted style=colorful,
  minted language=java,minted options={fontsize=\small,linenos,numbersep=3mm},
  colback=blue!5!white,colframe=blue!75!black,listing only,
  left=5mm,enhanced,
  overlay={\begin{tcbclipinterior}\fill[red!20!blue!20!white] (frame.south west)
    rectangle ([xshift=5mm]frame.north west);\end{tcbclipinterior}}}

\begin{myjava}
public class HelloWorld {
  // A 'Hello World' in Java
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
\end{myjava}
```

```
1 public class HelloWorld {
2   // A 'Hello World' in Java
3   public static void main(String[] args) {
4     System.out.println("Hello World!");
5   }
6 }
```

`/tcb/minted style=`⟨*style*⟩ (no default, initially unset)

Sets a ⟨*style*⟩ known to `Pygments` [12]. This is independent from `/tcb/minted options` → P. 209. Note that styles are always applied globally; all following examples will be set in the given ⟨*style*⟩ until a new style is set. Also note that setting `\usemintedstyle{`⟨*style*⟩`}` only once per document is more economic, if all styles in a document are the same. For examples of different styles, see `/tcb/minted language` → P. 209 and `/tcb/minted options` → P. 209.

See further options in Section 9.6 on the following page.

9.6 Common Option Keys of all Libraries

For the ⟨*options*⟩ in `tcblisting` → P. 200 respectively `\tcbinputlisting` → P. 202 the following `pgf` keys can be applied. The key tree path `/tcb/` is not to be used inside these macros.

`/tcb/listing engine=`⟨*engine*⟩ (no default)

Sets the ⟨*engine*⟩ which typesets the listings. Feasible values are

- `listings`, if library LIB listings or LIB listingsutf8 is loaded.
- `minted`, if library LIB minted is loaded.

`/tcb/listing file=`⟨*file name*⟩ (no default, initially `\jobname.listing`)

Sets the ⟨*file name*⟩ of the file which is used to save listings.

`/tcb/listing and text` (no value, initially set)

Typesets the environment content as listing in the upper part and as compiled text in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing and text}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

`/tcb/text and listing` (no value)

Typesets the environment content as compiled text in the upper part and as listing in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text and listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

```
This is a \LaTeX\ example.
```

`/tcb/listing only` (no value)

Typesets the environment content as listing.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing only}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

/tcb/text only (no value)

Typesets the environment content as compiled text.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text only}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

/tcb/comment=⟨*text*⟩ (no default, initially empty)

Records a comment with ⟨*text*⟩ as content. The comment is displayed e. g. in conjunction with /tcb/listing and comment → P. 213 and /tcb/comment and listing → P. 213.

```
\begin{tcblisting}{comment={This comment is really only a comment},
  colback=red!5!white,colframe=red!75!black}
This is a \textbf{tcolorbox}.
\end{tcblisting}
```

This is a \textbf{tcolorbox}.

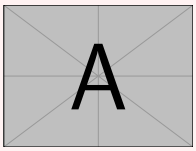
This is a **tcolorbox**.

/tcb/image comment={⟨*options*⟩**}{**⟨*filename*⟩**}** (style, no default, initially unset)

Uses an image denoted by ⟨*filename*⟩ as *comment* for the listing. The image is included by the standard **\includegraphics** macro with given ⟨*options*⟩.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side comment,
  image comment={width=2.5cm}{example-image-a.pdf},center lower}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.



/tcb/listing and comment (no value)

Typesets the environment content as listing in the upper part and a given comment in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing and comment,
  comment={This is my comment. It may contain line breaks.\par
    It can even use the environment content
    \flqq\ignorespaces\tcbuselistingtext\unskip\frqq}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

This is my comment. It may contain line breaks.
It can even use the environment content «This is a LaTeX example.»

/tcb/comment and listing (no value)

Typesets a given comment in the upper part and the environment content as listing in the lower part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment and listing,
comment={This is my comment.}}
This is a \LaTeX\ example.
\end{tcblisting}
```

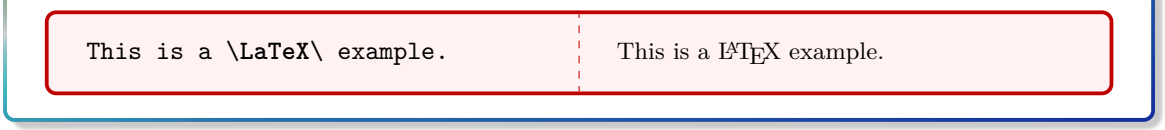
This is my comment.

This is a \LaTeX\ example.

/tcb/listing side text (no value)

Typesets the environment content side by side as listing in the left (upper) part and as compiled text in the right (lower) part.

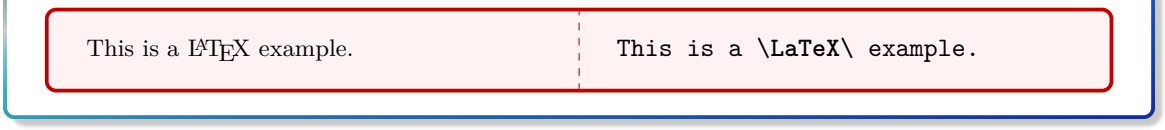
```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side text}
This is a \LaTeX\ example.
\end{tcblisting}
```



/tcb/text side listing (no value)

Typesets the environment content side by side as compiled text in the left (upper) part and as listing in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text side listing}
This is a \LaTeX\ example.
\end{tcblisting}
```



/tcb/listing outside text (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and as compiled text outside the box in the right part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 3.12, see page 62.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing outside text}
This is a \LaTeX\ example.
\end{tcblisting}
```

/tcb/text outside listing (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and as compiled text outside the box in the left part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 3.12, see page 62.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text outside listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

/tcb/listing side comment (no value)

Typesets the environment content side by side as listing in the left (upper) part and a given comment in the right (lower) part.

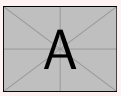
```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing side comment,
  righthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

/tcb/comment side listing (no value)

Typesets the environment content side by side with a given comment in the left (upper) part and as listing in the right (lower) part.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment side listing,
  lefthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```



This is a \LaTeX\ example.

/tcb/listing outside comment (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and a given comment outside the box in the right part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 3.12, see page 62.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing outside comment,
  righthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

/tcb/comment outside listing (no value)

Typesets the environment content side by side as listing in a `tcolorbox` and a given comment outside the box in the left part of the page. Nevertheless, the outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The space partitioning is done with the side by side options from Section 3.12, see page 62.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment outside listing,
  lefthand width=1.5cm,image comment={width=1.5cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a \LaTeX\ example.

/tcb/listing above text (no value)

Typesets the environment content as listing in a `tcolorbox` and as compiled text outside and below the box. The outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The distance between box and text is controlled by `/tcb/middle` → P. 29.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing above text}
This is a \LaTeX\ example.
\end{tcblisting}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

/tcb/text above listing (no value)

Typesets the environment content as listing in a `tcolorbox` and as compiled text outside and above the box. The outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The distance between box and text is controlled by `/tcb/middle` → P. 29.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,text above listing}
This is a \LaTeX\ example.
\end{tcblisting}
```

This is a LaTeX example.

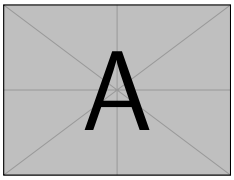
```
This is a \LaTeX\ example.
```

/tcb/listing above comment (no value)

Typesets the environment content as listing in a `tcolorbox` and a given comment outside and below the box. The outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The distance between box and comment is controlled by `/tcb/middle` → P. 29.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,listing above comment,
  center lower,image comment={width=3cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```

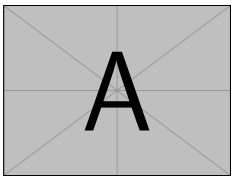
This is a \LaTeX\ example.



/tcb/comment above listing (no value)

Typesets the environment content as listing in a `tcolorbox` and a given comment outside and above the box. The outside text is treated as *lower* part of the `tcolorbox` and can be formatted with all lower part options. The distance between box and comment is controlled by `/tcb/middle` → P. 29.

```
\begin{tcblisting}{colback=red!5!white,colframe=red!75!black,comment above listing,
  center lower,image comment={width=3cm}{example-image-a.pdf}}
This is a \LaTeX\ example.
\end{tcblisting}
```



This is a \LaTeX\ example.

9.7 Creation of LaTeX Tutorials

The following source code gives a guideline for the creation of LaTeX tutorials. In the next section, a framework for LaTeX exercises is described. All examples shall be numbered optionally.

Firstly, some additional `tcb` keys are defined for the appearance. For the examples, three environments `texexp`, `texexptitled`, and `texexptitledspec` are defined with automatic numbering.

- `texexp` is used for untitled examles,
- `texexptitled` is used for titled examles,
- `texexptitledspec` is used for titled examles with special treatment.

Definition in the preamble:

```
\tcbset{
  texexp/.style={colframe=red!50!yellow!50!black, colback=red!50!yellow!5!white,
      coltitle=red!50!yellow!3!white,
      fonttitle=\small\sffamily\bfseries, fontupper=\small, fontlower=\small},
  example/.style 2 args={texexp,
      title={Example \thetcbcounter: #1},label={#2}},
}

\newtcblisting{texexp}[1]{texexp,#1}
\newtcblisting[auto counter,number within=section]{texexptitled}[3][]{%
  example={#2}{#3},#1}
\newtcolorbox[use counter from=texexptitled]{texexptitledspec}[3][]{%
  example={#2}{#3},#1}
```

```
\begin{tcblisting}{texexp}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{tcblisting}
```

```
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
```

This is a LaTeX example which displays the text as source code and in compiled form.

```
\begin{texexptitled}{First example with a title line}{firstExample}
Here, we use Example \ref{firstExample} with a title line.
\end{texexptitled}
```

Example 9.1: First example with a title line

```
Here, we use Example \ref{firstExample} with a title line.
```

Here, we use Example 9.1 with a title line.

```
\begin{texexp}{}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{texexp}
```

This is a \LaTeX\ example which displays the text as source code and in compiled form.

This is a LaTeX example which displays the text as source code and in compiled form.

```
\begin{texexp}{text and listing}
This is a \LaTeX\ example which displays the text as source code
and in compiled form.
\end{texexp}
```

This is a LaTeX example which displays the text as source code and in compiled form.

This is a \LaTeX\ example which displays the text as source code and in compiled form.

```
\begin{texexp}{listing only}
This is a \LaTeX\ example which displays the text as source code only.
\end{texexp}
```

This is a \LaTeX\ example which displays the text as source code only.

```
\begin{texexp}{text only}
This is a \LaTeX\ example which displays the text in compiled form only.
\end{texexp}
```

This is a LaTeX example which displays the text in compiled form only.

```
\begin{texexptitled}{An Example with a Heading}{heading1}
This is a \LaTeX\ example with a numbered heading line
which can be referred to.
\end{texexptitled}
Here, we see Example \ref{heading1}.
```

Example 9.2: An Example with a Heading

```
This is a \LaTeX\ example with a numbered heading line
which can be referred to.
```

This is a LaTeX example with a numbered heading line which can be referred to.

Here, we see Example 9.2.

```
\begin{texexptitled}[listing only]{Another Example with a Heading}{heading2}
The keys can be used in combination. Here, an example with a heading line
and source code only is given.
\end{texexptitled}
Here, we see Example \ref{heading2}.
```

Example 9.3: Another Example with a Heading

```
The keys can be used in combination. Here, an example with a heading line
and source code only is given.
```

Here, we see Example 9.3.

```
\begin{texexptitled}[float]{A floating Example with a Heading}{heading3}
This is another \LaTeX\ example with numbered heading line.
But now, the box is a floating object.
\end{texexptitled}
```

Example 9.4: A floating Example with a Heading

```
This is another \LaTeX\ example with numbered heading line.
But now, the box is a floating object.
```

This is another LaTeX example with numbered heading line. But now, the box is a floating object.

```
The floating box of the last example is seen as Example \ref{heading3}
on page \pageref{heading3}.
```

The floating box of the last example is seen as Example 9.4 on page 220.

```
\begin{texexptitledspec}{Special application}{texexpbox1}
\begin{lstlisting}[style=tcblatex]
Some \LaTeX\ source code.
\end{lstlisting}
\tcblower
For special cases, the environment |texexptitledspec| with style
|example| can be used directly. As one can see, the upper and the lower
part of the box can be used uncoupled also.
\end{texexptitledspec}
```

Example 9.5: Special application

```
Some \LaTeX\ source code.
```

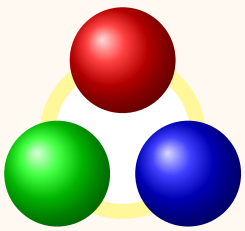
For special cases, the environment `texexptitledspec` with style `example` can be used directly. As one can see, the upper and the lower part of the box can be used uncoupled also.

The following series of examples demonstrate the application of tcolorbox → P. 9 options for diversification.

```
\begin{texexptitled}{How to use options (1):\par The basic example}{options1}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 9.6: How to use options (1): The basic example

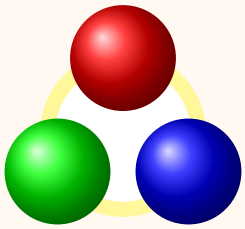
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



```
\begin{texexptitled}[center lower,enhanced,segmentation hidden,middle=0mm]
  {How to use options (2):\par The text output is centered and the
    segmentation line has vanished.}{options2}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 9.7: How to use options (2):
The text output is centered and the segmentation line has vanished.

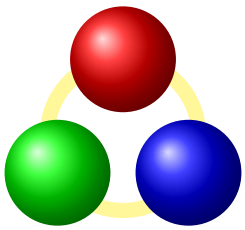
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



```
\begin{texexptitled}[tikz lower,bicolor,colbacklower=white]
  {How to use options (3):\par Here, the |tikzpicture| is totally hidden.
    The |bicolor| skin highlights the output.}{options3}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
\end{texexptitled}
```

Example 9.8: How to use options (3):
Here, the `tikzpicture` is totally hidden. The `bicolor` skin highlights the output.

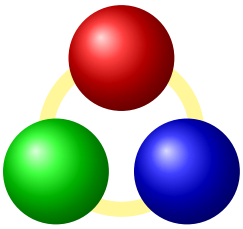
```
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c] (\w:1cm) circle (7mm);}
```



```
\begin{texexptitled}[center lower,listing side text,righthand width=3.5cm,
  bicolor,colbacklower=white]
  {How to use options (4):\par The |bicolor| skin also works with side
   by side mode}{options4}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 9.9: How to use options (4):
The `bicolor` skin also works with side by side mode

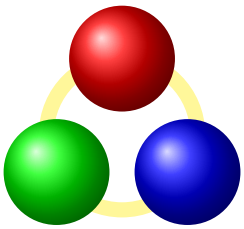
```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



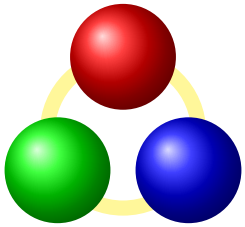
```
\begin{texexptitled}[center lower,listing outside text,righthand width=3.5cm]
  {How to use options (5):\par Putting our picture outside is just
    a matter of one word.}{options5}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```

Example 9.10: How to use options (5):
Putting our picture outside is just a matter of one word.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```



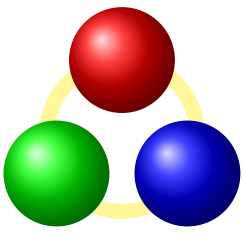
```
\begin{texexptitled}[center lower,text above listing]
  {How to use options (6):\par The picture may also be put above
    the listing box.}{options6}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```



Example 9.11: How to use options (6):
The picture may also be put above the listing box.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```

```
\begin{texexptitled}[beamer,center lower,text outside listing,lefthand width=3.5cm]
  {How to use options (7):\par Our style is easily transformed into
    a beamerish one.}{options7}
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
\end{texexptitled}
```



Example 9.12: How to use options (7):
Our style is easily transformed into a beamerish one.

```
\begin{tikzpicture}
\path[fill=yellow!50!white] (0,0) circle (11mm);
\path[fill=white] (0,0) circle (9mm);
\foreach \w/\c in {90/red,210/green,330/blue}
{\path[shading=ball,ball color=\c]
    (\w:1cm) circle (7mm);}
\end{tikzpicture}
```

9.8 Creation of LaTeX Exercises

In the following, a guideline is given for the creation of LaTeX exercises with solutions. These solutions are saved to disk for application at a place of choice. Therefore, all used exercises are logged to a file `\jobname.sol` for automatic processing. The solution contents themselves are saved to a subdirectory named `solutions`.

```
%\newcounter{texercise}  % preamble

\newwrite\solout
\def\openoutsol{\immediate\openout\solout\jobname.sol}
\def\solfile#1{solutions/texercise#1.tex}
\def\writesol#1{\immediate\write\solout{\noexpand\processsol{\thetcbcounter}{#1}}}%
\def\closeoutsol{\immediate\closeout\solout}
\def\inputsol{\IfFileExists{\jobname.sol}{\input{\jobname.sol}}{}}
```

- Before the first exercise is given, `\openoutsol` has to be called to start logging.
- The solution is given as content of a `tcboutputlisting` → P. 202 environment. Note, that you can use this content also inside the exercise with `\tcbuselistingtext` → P. 202 in compiled form.
- After the last exercise is given (and before using the solutions), `\closeoutsol` has to be called to stop logging.
- The solutions are loaded by `\inputsol`.

Inside the exercise text, there may be text parts which are needed as LaTeX source code and as compiled text as well. These parts can be saved by `tcbwritetemp` → P. 88 and used in compiled form by `\tcbusetemp` → P. 88 or as source code by `\tcbusetemplisting` → P. 202.

At first, we generate some a common style for the exercises and the solutions. Further, since exercises and solutions should be numbered, we force to use a label ⟨*marker*⟩. Automatically, the label `exe:`⟨*marker*⟩ is used to mark the exercise and the label `sol:`⟨*marker*⟩ is used to mark the solution.

```
\tcbset{texercisestyle/.style={arc=0.5mm, colframe=blue!25!yellow!90!white,
  colback=blue!25!yellow!5!white, coltitle=blue!25!yellow!40!black,
  fonttitle=\small\sffamily\bfseries, fontupper=\small, fontlower=\small}}
```

With these preparations, the kernel environment `texercise` for our exercises is created quickly:

Definition in the preamble:

```
\newtcolorbox[auto counter,number within=section,list inside=exam]%
  {texercise}[2][]{texercisestyle,
  listing file={\solfile\thetcbcounter},label={exe:#2},
  phantom={\writesol{#2}},
  title={Exercise \thetcbcounter\hfill\mdseries Solution on page \pageref{sol:#2}},
  list text={Exercise with solution on page \pageref{sol:#2}},#1}
```

```
\newtcolorbox{texercise}[2][]{texercisestyle,
  listing file={\solfile\thetexercise},
  phantom={\refstepcounter{texercise}\label{exe:#2}\writesol{#2}},
  title={Exercise \arabic{texercise}%
    \hfill\mdseries Solution on page \pageref{sol:#2}},#1}
```

The following examples demonstrate the application.

```
\begin{texercise}{tabular_example}
\textit{Create the following table:}\par\smallskip%
\begin{tcboutputlisting}
\begin{tabular}{|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}\hline
\multicolumn{4}{|c|}{\bfseries\itshape Das alte Italien}\\\hline
\multicolumn{2}{|c|}{\bfseries Antike} &
\multicolumn{2}{c|}{\bfseries Mittelalter}\\\hline
\multicolumn{1}{|c|}{\itshape Republik}&
\multicolumn{1}{c|}{\itshape Kaiserreich}&
\multicolumn{1}{c|}{\itshape Franken}&
\multicolumn{1}{c|}{\itshape Teilstaaten}\\\hline
In den Zeiten der r\"{o}mischen Republik standen dem Staat jeweils zwei
Konsuln vor, deren Machtbefugnisse identisch waren. &
Das r\"{o}mische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser,
regiert.
& In der V\"{o}lkerwanderungszeit \"{u}bernahmen die Goten und sp\"{a}ter die
Franken die Vorherrschaft.
& Im sp\"{a}teren Mittelalter regierten F\"{u}rsten einen Fleckenteppich
von Einzelstaaten.\\\hline
\end{tabular}
\end{tcboutputlisting}
\tcbuselistingtext%
\end{texercise}
```

Exercise 9.1 Solution on page 229

Create the following table:

Das alte Italien			
Antike		**Mittelalter**	
Republik	*Kaiserreich*	*Franken*	*Teilstaaten*
In den Zeiten der römischen Republik standen dem Staat jeweils zwei Konsuln vor, deren Machtbefugnisse identisch waren.	Das römische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser, regiert.	In der Völkerwanderungszeit übernahmen die Goten und später die Franken die Vorherrschaft.	Im späteren Mittelalter regierten Fürsten einen Fleckenteppich von Einzelstaaten.

```
\begin{texercise}{macro_oneparam}
\begin{tcboutputlisting}
\newcommand{\headingline}[1]{%
  \begin{center}\Large\bfseries #1\end{center}}
\end{tcboutputlisting}
\tcbuselistingtext%

Create a new macro \verb+\headingline+ which produces the
following output:\par\smallskip
\begin{tcbwritetemp}
\headingline{Very important heading}
\end{tcbwritetemp}
\tcbusetemplisting\tcbusetemp%
\end{texercise}
```

Exercise 9.2 Solution on page 229

Create a new macro `\headingline` which produces the following output:

```
\headingline{Very important heading}
```

Very important heading

```
\begin{texercise}{macro_twoparam}
\begin{tcboutputlisting}
\newcommand{\minitable}[2]{%
  \begin{center}\begin{tabular}{p{10cm}}\hline%
  \multicolumn{1}{c}{\bfseries#1}\\\hline%
  #2\\\hline%
  \end{tabular}\end{center}}
\end{tcboutputlisting}
\tcbuselistingtext%
Create a new macro \verb+\minitable+ which produces the
following output:\par\smallskip
\begin{tcbwritetemp}
\minitable{My heading}{In this tiny tabular, there is only a heading
  and some text below which has a width of ten centimeters.}
\end{tcbwritetemp}
\tcbusetemplisting\par\smallskip\tcbusetemp%
\end{texercise}
```

Exercise 9.3 Solution on page 229

Create a new macro `\minitable` which produces the following output:

```
\minitable{My heading}{In this tiny tabular, there is only a heading
  and some text below which has a width of ten centimeters.}
```

My heading
In this tiny tabular, there is only a heading and some text below which has a width of ten centimeters.

```
\begin{texercise}{macro_threeparam}
\begin{tcboutputlisting}
\newcommand{\synop}[3]{%
  \begin{tabular}{@{}p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}|%
    p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}@{}}\hline
  \multicolumn{2}{c}{\bfseries #1}\\\hline
  \multicolumn{1}{c|}{\itshape English}&
  \multicolumn{1}{c}{\itshape German}\\\hline
  #2 & #3
  \end{tabular}}
\end{tcboutputlisting}
\tcbuselistingtext%
Create a new macro \verb+\synop+ which typesets a synoptic text according
to the following example. Base your macro on a tabular which takes the
total line width.\par\smallskip
\begin{tcbwritetemp}
\synop{Neil Armstrong}%
{That's one small step for a man, one giant leap for mankind.}%
{Das ist ein kleiner Schritt f\"{u}r einen Mann,
   ein riesiger Sprung f\"{u}r die Menschheit.}
\end{tcbwritetemp}
\tcbusetemplisting\par\smallskip\tcbusetemp%
\end{texercise}
```

Exercise 9.4 Solution on page 230

Create a new macro `\synop` which typesets a synoptic text according to the following example. Base your macro on a tabular which takes the total line width.

```
\synop{Neil Armstrong}%
{That's one small step for a man, one giant leap for mankind.}%
{Das ist ein kleiner Schritt f\"{u}r einen Mann,
   ein riesiger Sprung f\"{u}r die Menschheit.}
```

Neil Armstrong	
English	*German*
That's one small step for a man, one giant leap for mankind.	Das ist ein kleiner Schritt für einen Mann, ein riesiger Sprung für die Menschheit.

Now, we give a list of all exercises with:

```
\tcblistof[\subsection]{exam}{List of Exercises%
  \label{listofexercises}}
```

9.9 List of Exercises

9.1 Exercise with solution on page 229 226
9.2 Exercise with solution on page 229 227
9.3 Exercise with solution on page 229 227
9.4 Exercise with solution on page 230 228

9.10 Solutions for the given LaTeX Exercises

For all solutions, a macro `\processsol` was written to the file `\jobname.sol`. Now, we need a definition for this macro to use the solutions.

```
% \usepackage{hyperref} % for \phantomsection
\newcommand{\processsol}[2]{%
  \tcbinputlisting{texercisestyle,listing only,
    phantom={\phantomsection\label{sol:#2}},%
    title={Solution for Exercise \ref{exe:#2} on page \pageref{exe:#2}},
    listing file={\solfile{#1}}}}
```

The loading of all solutions is done by:

```
\inputsol
```

With this, we get:

Solution for Exercise 9.1 on page 226

```
\begin{tabular}{|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}\hline
\multicolumn{4}{|c|}{\bfseries\itshape Das alte Italien}\\\hline
\multicolumn{2}{|c|}{\bfseries Antike} &
\multicolumn{2}{c|}{\bfseries Mittelalter}\\\hline
\multicolumn{1}{|c|}{\itshape Republik}&
\multicolumn{1}{c|}{\itshape Kaiserreich}&
\multicolumn{1}{c|}{\itshape Franken}&
\multicolumn{1}{c|}{\itshape Teilstaaten}\\\hline
In den Zeiten der r\"{o}mischen Republik standen dem Staat jeweils zwei
Konsuln vor, deren Machtbefugnisse identisch waren. &
Das r\"{o}mische Kaiserreich wurde von einem Alleinherrscher, dem Kaiser,
regiert.
& In der V\"{o}lkerwanderungszeit \"{u}bernahmen die Goten und sp\"{a}ter die
Franken die Vorherrschaft.
& Im sp\"{a}teren Mittelalter regierten F\"{u}rsten einen Fleckenteppich
von Einzelstaaten.\\\hline
\end{tabular}
```

Solution for Exercise 9.2 on page 227

```
\newcommand{\headingline}[1]{%
  \begin{center}\Large\bfseries #1\end{center}}
```

Solution for Exercise 9.3 on page 227

```
\newcommand{\minitable}[2]{%
  \begin{center}\begin{tabular}{p{10cm}}\hline%
  \multicolumn{1}{c}{\bfseries#1}\\\hline%
  #2\\\hline%
  \end{tabular}\end{center}}
```

Solution for Exercise 9.4 on page 228

```
\newcommand{\synop}[3]{%
  \begin{tabular}{@{}p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}|%
    p{(\linewidth-\tabcolsep*2-\arrayrulewidth)/2}@{}}\hline
  \multicolumn{2}{c}{\bfseries #1}\\\hline
  \multicolumn{1}{c|}{\itshape English}&
  \multicolumn{1}{c}{\itshape German}\\\hline
  #2 & #3
  \end{tabular}}
```

10 Library LIB theorems

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{theorems}
```

This also loads the package `amsmath`.

10.1 Macros of the Library

`\newtcbtheorem`[⟨*init options*⟩]{⟨*name*⟩}{⟨*display name*⟩}{⟨*options*⟩}{⟨*prefix*⟩}

Creates new environments ⟨*name*⟩ and ⟨*name*⟩`*` based on `tcolorbox` to frame a (mathematical) theorem. The ⟨*display name*⟩ is used in the title line with a number, e. g. «Theorem 5.1». The ⟨*options*⟩ are given to the underlying `tcolorbox` to control the appearance. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 on page 82. The new environment ⟨*name*⟩ takes one optional and two mandatory parameters. The optional parameter supplements the options and should be used only in rare cases. The first mandatory parameter is the title text for the theorem and the second mandatory parameter is a ⟨*marker*⟩. The theorem is automatically labeled with ⟨*prefix*⟩`:`⟨*marker*⟩. The new environment ⟨*name*⟩`*` takes one optional and one mandatory parameter and represents an unnumbered variant of the environment ⟨*name*⟩. This variant is not labeled and not listed in lists of theorems.

Definition in the preamble:

```
\newtcbtheorem[number within=section]{mytheo}{My Theorem}%
  {colback=green!5,colframe=green!35!black,fonttitle=\bfseries}{th}
```

```
\begin{mytheo}{This is my title}{theoexample}
  This is the text of the theorem. The counter is automatically assigned and,
  in this example, prefixed with the section number. This theorem is numbered with
  \ref{th:theoexample} and is given on page \pageref{th:theoexample}.
\end{mytheo}
```

My Theorem 10.1: This is my title

This is the text of the theorem. The counter is automatically assigned and, in this example, prefixed with the section number. This theorem is numbered with 10.1 and is given on page 231.

```
\begin{mytheo}[label=myownlabel]{This is my title}{}
  The label parameter can be left empty without \LaTeX\ error.
  Or you may use an own label to reference Theorem \ref{myownlabel}.
\end{mytheo}
```

My Theorem 10.2: This is my title

The label parameter can be left empty without LaTeX error. Or you may use an own label to reference Theorem 10.2.

```
\begin{mytheo}{}{}
  The title can also be left empty without problem. Note that the ':'
  vanished magically.
\end{mytheo}
```

My Theorem 10.3

The title can also be left empty without problem. Note that the ':' vanished magically.

```
\begin{mytheo*}{Unnumbered Theorem}
  This theorem is not numbered.
\end{mytheo*}
```

My Theorem: Unnumbered Theorem

This theorem is not numbered.

```
\begin{mytheo*}{}
  This theorem has no number and no title.
\end{mytheo*}
```

My Theorem

This theorem has no number and no title.

`\renewtcbtheorem[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*display name*⟩`}{`⟨*options*⟩`}{`⟨*prefix*⟩`}`

Operates like `\newtcbtheorem` → P. 231, but based on `\renewenvironment` instead of `\newenvironment`. An existing environment is redefined.

`\tcbmaketheorem{`⟨*name*⟩`}{`⟨*display name*⟩`}{`⟨*options*⟩`}{`⟨*counter*⟩`}{`⟨*prefix*⟩`}`

`\newtcbtheorem` → P. 231 supersedes this macro.

Creates a new environment ⟨*name*⟩ based on `tcolorbox` to frame a (mathematical) theorem. The ⟨*display name*⟩ is used in the title line with a number, e. g. «Theorem 5.1». The ⟨*options*⟩ are given to the underlying `tcolorbox` to control the appearance. The ⟨*counter*⟩ is used for automatic numbering. The new environment ⟨*name*⟩ takes one optional and two mandatory parameters. The optional parameter supplements the options and should be used only in rare cases. The first mandatory parameter is the title text for the theorem and the second mandatory parameter is a ⟨*marker*⟩. The theorem is automatically labeled with ⟨*prefix*⟩`:`⟨*marker*⟩.

deprecated

`\tcboxmath`[⟨*options*⟩]{⟨*mathematical box content*⟩}

Creates a `tcolorbox` → P. 9 which is fitted to the width of the given ⟨*mathematical box content*⟩. This box is intended to be applied as part of a larger formula and may be used as replacement for the `\boxed` macro of `amsmath`.

```
\begin{equation}
\tcbset{fonttitle=\scriptsize}
\tcboxmath[colback=LightBlue!25!white,colframe=blue]{ a^2 = 16 }
\quad \Rightarrow \quad
\tcboxmath[colback=Salmon!25!white,colframe=red,title=Implication]%
  { a = 4 ~\vee~ a=-4. }
\end{equation}
```

$$\boxed{a^2 = 16} \quad \Rightarrow \quad \boxed{\begin{array}{l}\text{Implication}\\ a = 4 \ \vee \ a = -4.\end{array}} \tag{3}$$

`\tcbhighmath`[⟨*options*⟩]{⟨*mathematical box content*⟩}

This is a special case of the `\tcboxmath` macro which uses the style `/tcb/highlight math` → P. 240. It is intended to provide context sensitive highlighting of formula parts. The color settings via `/tcb/highlight math style` → P. 240 may be different inside theorems or other colored areas and outside.

```
\tcbset{myformula/.style={colback=yellow!10!white,colframe=red!50!black,
  every box/.style={highlight math style={colback=LightBlue!50!white,colframe=Navy}}
  }}

\begin{align}
  \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} &= \infty.\\
  \int x^2 ~\text{d}x                                  &= \frac13 x^3 + c.
\end{align}

\begin{tcolorbox}[ams align,myformula]
  \tcbhighmath{\sum\limits_{n=1}^{\infty} \frac{1}{n}} &= \infty.\\
  \int x^2 ~\text{d}x                                  &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\boxed{\sum_{n=1}^{\infty} \frac{1}{n}} = \infty. \tag{4}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{5}$$

$$\boxed{\sum_{n=1}^{\infty} \frac{1}{n}} = \infty. \tag{6}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{7}$$

\tcbhighmath → P. 233 can be used in symbiosis with the empheq package which allows to specify own boxing commands to mark multiline formulas.

```
% \usepackage{empheq}
\begin{empheq}[box=\tcbhighmath]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}

\tcbset{highlight math style={enhanced,
  colframe=red!60!black,colback=yellow!50!white,arc=4pt,boxrule=1pt,
  drop fuzzy shadow}}

\begin{empheq}[box=\tcbhighmath]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}
```

$$a = \sin(z) \tag{8}$$
$$E = mc^2 + \int_a^b x\,dx \tag{9}$$

$$a = \sin(z) \tag{10}$$
$$E = mc^2 + \int_a^b x\,dx \tag{11}$$

Besides \tcbhighmath → P. 233, one can easily define an independent new box based on \tcbox → P. 11 which acts like \tcbhighmath → P. 233:

```
% \usepackage{empheq}
\newtcbox{\otherbox}[1][]{nobeforeafter,math upper,tcbox raise base,
  enhanced,frame hidden,boxrule=0pt,interior style={top color=green!10!white,
  bottom color=green!10!white,middle color=green!50!yellow},
  fuzzy halo=1pt with green,#1}

\begin{empheq}[box=\otherbox]{align}
a&=\sin(z)\\
E&=mc^2 + \int_a^b x\, dx
\end{empheq}

\begin{equation}
\tcbhighmath{E} = \otherbox{mc^2}
\end{equation}
```

$$a = \sin(z) \tag{12}$$
$$E = mc^2 + \int_a^b x\,dx \tag{13}$$

$$E = mc^2 \tag{14}$$

10.2 Option Keys of the Library

/tcb/separator sign=⟨*sign*⟩ (no default, initially :)

The given ⟨*sign*⟩ is used inside the title text of a theorem as separater between display name combined with number and the specific title text. It is omitted, if there is no specific title text.

```
% \usepackage{amssymb}
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign={\ $\blacktriangleright$}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.4 ▶ My example

My theorem text.

/tcb/separator sign colon (style, no value, initially set)

Sets /tcb/separator sign to the default colon : sign.

/tcb/separator sign dash (style, no value)

Sets /tcb/separator sign to an en-dash sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign dash}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.5 – My example

My theorem text.

/tcb/separator sign none (style, no value)

Sets /tcb/separator sign to empty.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign none}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.6 My example

My theorem text.

`/tcb/description delimiters={⟨left⟩}{⟨right⟩}` (no default, initially empty)

The given ⟨*left*⟩ and ⟨*right*⟩ delimiter signs are used to frame the descriptive title text of a theorem.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters={\flqq}{\frqq}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.7: «My example»

My theorem text.

`/tcb/description delimiters parenthesis` (style, no value)

Sets `/tcb/description delimiters` to (and).

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters parenthesis}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.8: (My example)

My theorem text.

`/tcb/description delimiters none` (style, no value, initially set)

Sets `/tcb/description delimiters` to the default empty texts.

`/tcb/description color=`⟨*color*⟩ (default empty, initially empty)

Sets the ⟨*color*⟩ of the descriptive title text deviating from `/tcb/coltitle` → P. 21. The color is reset to `/tcb/coltitle` → P. 21, if `description color` is used without value.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description color=red!25!yellow}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.9: My example

My theorem text.

`/tcb/description font=`⟨*text*⟩ (default empty, initially empty)

Sets ⟨*text*⟩ (e. g. font settings) before the descriptive title text deviating from `/tcb/fonttitle` → P. 22. The ⟨*text*⟩ is removed, if `description font` is used without value.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description delimiters={\glqq}{\grqq},
   description font=\mdseries\itshape}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.10: *„My example“*

My theorem text.

`/tcb/description formatter=`⟨*macro*⟩ (default empty, initially empty)

Sets ⟨*macro*⟩ as formatter for the descriptive title text. The ⟨*macro*⟩ has to take one mandatory argument (the description text).

Note that `/tcb/description delimiters` → P. 236, `/tcb/description color` → P. 236, and `/tcb/description font` are ignored, if this option is used.

If `description formatter` is used without value, the formatter is reset to its standard behavior.

```
\newtcbox{\formbox}{enhanced,frame empty,size=minimal,boxsep=2pt,arc=1pt,
  on line,interior style image=goldshade.png}

\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   description formatter=\formbox}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.11: My example

My theorem text.

`/tcb/terminator sign=`⟨*sign*⟩ (no default, initially empty)

The given ⟨*sign*⟩ is used as terminator at the end of the title text of a theorem.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   terminator sign={.}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.12: My example.

My theorem text.

/tcb/terminator sign colon (style, no value, initially set)

Sets /tcb/terminator sign → P. 237 to the colon : sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   separator sign dash,terminator sign colon}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.13 – My example:

My theorem text.

/tcb/terminator sign dash (style, no value)

Sets /tcb/terminator sign → P. 237 to an en-dash sign.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   terminator sign dash}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.14: My example –

My theorem text.

/tcb/terminator sign none (style, no value)

Sets /tcb/terminator sign → P. 237 to the default empty text.

/tcb/theorem name and number (style, no value, initially set)

Prints theorem name followed by theorem number inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem name and number}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem 10.15: My example

My theorem text.

/tcb/theorem number and name (style, no value)

Prints theorem number followed by theorem name inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem number and name}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

10.16 Theorem: My example

My theorem text.

/tcb/theorem name (style, no value)

Prints theorem name without number inside the title.

```
\newtcbtheorem[use counter from=mytheo]{sometheorem}{Theorem}%
  {colback=white,colframe=red!50!black,fonttitle=\bfseries,
   theorem name,enhanced,watermark text={\thetcbcounter}}{theo}
\begin{sometheorem}{My example}{}
My theorem text.
\end{sometheorem}
```

Theorem: My example

My theorem text. 10.17

/tcb/theorem={⟨*display name*⟩}{⟨*counter*⟩}{⟨*title*⟩}{⟨*marker*⟩} (no default)

This key is internally used by `\tcbmaketheorem` → P. 232, but can be used directly in a `tcolorbox` for a more flexible approach. The ⟨*display name*⟩ is used together with the increased ⟨*counter*⟩ value and the ⟨*title*⟩ for the title line of the box. Additionally, a `\label` with the given ⟨*marker*⟩ is created.

```
% \newcounter{texercise}%  preamble
\begin{tcolorbox}[colback=green!10,colframe=green!50!black,arc=4mm,
                  theorem={Test}{texercise}{Direct usage}{myMarker}]
Here, we see the test \ref{myMarker}.
\end{tcolorbox}
```

Test 1: Direct usage

Here, we see the test 1.

For a common appearance inside the document, the key `theorem` should not be used directly as in the example above, but as part of a new environment created by hand or using `\tcbmaketheorem` → P. 232 or using its successor `\newtcbtheorem` → P. 231.

/tcb/highlight math (style, no value)

Predefined style which is used for \tcbhighmath → P. 233. It can be changed comfortable with /tcb/highlight math style.

/tcb/highlight math style=⟨*style definition*⟩ (style, no default)

Changes the definition for /tcb/highlight math to the given ⟨*style definition*⟩. See \tcbhighmath → P. 233 for another example.

```
% \tcbuselibrary{skins}
\tcbset{highlight math style={enhanced,%<-- needed for the 'remember' options
  colframe=red,colback=red!10!white,boxsep=0pt}}
\begin{align*}
\tcbhighmath[remember as=fx]{f(x)}
     &= \int\limits_{1}^{x} \frac{1}{t^2}~dt
      = \left[ -\frac{1}{t} \right]_{1}^{x}\\
     &= -\frac{1}{x} + \frac{1}{1}\\
     &=
\tcbhighmath[remember,overlay={%
    \draw[blue,very thick,->] (fx.south) to[bend right] ([yshift=2mm]frame.west);}]
  {1-\frac{1}{x}.}
\end{align*}
```

$$\boxed{f(x)} = \int_1^x \frac{1}{t^2}\,dt = \left[-\frac{1}{t}\right]_1^x$$
$$= -\frac{1}{x} + \frac{1}{1}$$
$$= \boxed{1 - \frac{1}{x}.}$$

/tcb/math upper (style, no value)

Sets the upper part to mathematical mode with font \displaystyle.

/tcb/math lower (style, no value)

Sets the lower part to mathematical mode with font \displaystyle.

/tcb/math (style, no value)

Sets the upper part *and* lower part to mathematical mode with font \displaystyle.

```
\begin{tcolorbox}[math,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

The following styles are only tested to work with the original amsmath environments. If e.g. the equation environment is redefined as gather, then /tcb/ams equation should / could not be used. Obviously, you are encouraged to use /tcb/ams gather → P. 243 in this case.

/tcb/ams equation upper (style, no value)

Adds an amsmath equation environment to the start and end of the upper part.

/tcb/ams equation lower (style, no value)

Adds an amsmath equation environment to the start and end of the lower part.

/tcb/ams equation (style, no value)

Adds an amsmath equation environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams equation,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{15}$$

/tcb/ams equation* upper (style, no value)

Adds an amsmath equation* environment to the start and end of the upper part.

/tcb/ams equation* lower (style, no value)

Adds an amsmath equation* environment to the start and end of the lower part.

/tcb/ams equation* (style, no value)

Adds an amsmath equation* environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams equation*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

/tcb/ams align upper (style, no value)

Adds an `amsmath` `align` environment to the start and end of the upper part.

/tcb/ams align lower (style, no value)

Adds an `amsmath` `align` environment to the start and end of the lower part.

/tcb/ams align (style, no value)

Adds an `amsmath` `align` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams align,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x                    &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{16}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{17}$$

/tcb/ams align* upper (style, no value)

Adds an `amsmath` `align*` environment to the start and end of the upper part.

/tcb/ams align* lower (style, no value)

Adds an `amsmath` `align*` environment to the start and end of the lower part.

/tcb/ams align* (style, no value)

Adds an `amsmath` `align*` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams align*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x                    &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$

`/tcb/ams gather upper` (style, no value)

Adds an `amsmath gather` environment to the start and end of the upper part.

`/tcb/ams gather lower` (style, no value)

Adds an `amsmath gather` environment to the start and end of the lower part.

`/tcb/ams gather` (style, no value)

Adds an `amsmath gather` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams gather,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{18}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{19}$$

`/tcb/ams gather* upper` (style, no value)

Adds an `amsmath gather*` environment to the start and end of the upper part.

`/tcb/ams gather* lower` (style, no value)

Adds an `amsmath gather*` environment to the start and end of the lower part.

`/tcb/ams gather*` (style, no value)

Adds an `amsmath gather*` environment to the start and end of the upper *and* lower part.

```
\begin{tcolorbox}[ams gather*,colback=yellow!10!white,colframe=red!50!black]
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$

`/tcb/ams nodisplayskip upper` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the upper part. Note that the text content has to start with such a formula.

`/tcb/ams nodisplayskip lower` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the lower part. Note that the text content has to start with such a formula.

`/tcb/ams nodisplayskip` (style, no value)

Neutralizes the `\abovedisplayskip` of a following `align` or `gather` environment for the upper part *and* lower part. Note that the text content has to start with such a formula.

```
\begin{tcolorbox}[ams nodisplayskip,colback=yellow!10!white,colframe=red!50!black]
  \begin{gather}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
  \end{gather}
  And now for something completely different.
\end{tcolorbox}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{20}$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c. \tag{21}$$

And now for something completely different.

New colored mathematical environments are easily created using `\newtcolorbox` → P. 12:

```
\newtcolorbox{mymath}{ams gather*,colback=yellow!10!white,colframe=red!50!black}

\begin{mymath}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} = \infty.\\
  \int x^2 ~\text{d}x = \frac13 x^3 + c.
\end{mymath}
```

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty.$$

$$\int x^2 \,\mathrm{d}x = \frac{1}{3}x^3 + c.$$

All described options like `/tcb/ams gather upper` → P. 243, `/tcb/ams gather lower` → P. 243, `/tcb/ams gather` → P. 243 are (partially) setting (overwritting) the keys `/tcb/before upper` → P. 46, `/tcb/after upper` → P. 47, `/tcb/before lower` → P. 47, `/tcb/after lower` → P. 47.

Therefore, e. g. `\tcbset{ams gather,before upper={\text{Pythagoras:}}}` produces an invalid result. For this case, you are invited to use
`\tcbset{ams gather,before upper app={\text{Pythagoras:}}}`,
see `/tcb/before upper app` → P. 284.

`/tcb/theorem style=`⟨*name*⟩ (no default, initially `standard`)

Applies a predefined style ⟨*name*⟩ to the theorem environment. Some of the feasible ⟨*name*⟩ values resemble style names from the packages `theorem` and `ntheorem` to give convenient access to known patterns.

The styles alter `/tcb/separator sign`→ P. 235, `/tcb/description delimiters`→ P. 236, `/tcb/terminator sign`→ P. 237, and more. Therefore, one should apply such keys *after* a theorem style.

For the following examples, we use:

Definition in the preamble:

```
\newtcbtheorem[use counter from=mytheo]{theorem}{Theorem}{%
  fonttitle=\bfseries\upshape,fontupper=\itshape,
  colframe=green!50!black,colback=green!10!white,
  colbacktitle=green!20!white,coltitle=blue!75!black}{theo}
```

The predefined styles are:

- `standard`: This is the initial value.

```
\begin{theorem}[theorem style=standard]{standard}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 10.18: standard

This is my theorem.

$$a^2 + b^2 = c^2.$$

- `change standard`

```
\begin{theorem}[theorem style=change standard]{change standard}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.19 Theorem: change standard

This is my theorem.

$$a^2 + b^2 = c^2.$$

- `plain`

```
\begin{theorem}[theorem style=plain]{plain}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 10.20 (plain): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **break**

```
\begin{theorem}[theorem style=break]{break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 10.21 (break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **plain apart**

```
\begin{theorem}[theorem style=plain apart]{plain apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

Theorem 10.22 (plain apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

- **change**

```
\begin{theorem}[theorem style=change]{change}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.23 Theorem (change): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **change break**

```
\begin{theorem}[theorem style=change break]{change break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.24 Theorem (change break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **change apart**

```
\begin{theorem}[theorem style=change apart]{change apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.25 Theorem (change apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

- **margin**

```
\begin{theorem}[theorem style=margin,left=10mm]{margin}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin,left=10mm,oversize]{margin}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.26 Theorem (margin): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

10.27 Theorem (margin): *This is my theorem.*

$$a^2 + b^2 = c^2.$$

- **margin break**

```
\begin{theorem}[theorem style=margin break,left=10mm]{margin break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin break,left=10mm,oversize]{margin break}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.28 Theorem (margin break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

10.29 Theorem (margin break):
This is my theorem.

$$a^2 + b^2 = c^2.$$

- **margin apart**

```
\begin{theorem}[theorem style=margin apart,left=10mm]{margin apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
\begin{theorem}[theorem style=margin apart,left=10mm,oversize]{margin apart}{}
This is my theorem. \begin{equation*} a^2 + b^2 = c^2. \end{equation*}
\end{theorem}
```

10.30 Theorem (margin apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

10.31 Theorem (margin apart)

This is my theorem.

$$a^2 + b^2 = c^2.$$

10.3 Examples for Definitions and Theorems

In the following, the application of \tcbmaketheorem → P. 232 to highlight mathematical definitions, theorems, or the like is demonstrated.

At first, additional tcb keys are created for the appearance of the colored boxes. It is assumed that theorems and corollaries should be identically colored. All following environments are numbered with a common counter, but this can be changed easily. Here, the counter output is supplemented by the subsection number.

Definition in the preamble:

```
\tcbset{
  defstyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
              arc=0mm, colback=blue!5!white,colframe=blue!75!black},
  theostyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
               colback=red!10!white,colframe=red!75!black},
}
\newtcbtheorem[number within=subsection]{Definition}{Definition}{defstyle}{def}
\newtcbtheorem[use counter from=Definition]{Theorem}{Theorem}{theostyle}{theo}
\newtcbtheorem[use counter from=Definition]{Corollary}{Corollary}{theostyle}{cor}
```

By \newtcbtheorem → P. 231, commonly numbered theorem environments are created now. defstyle and theostyle are used for the appearance.

Now, everything is prepared for the following examples.

```
The following theorem is numbered as Theorem \ref{theo:diffbarstetig} and
referenced with the marker \texttt{theo:diffbarstetig}.\bigskip

\begin{Theorem}{Differenzierbarkeit bedingt Stetigkeit, wobei diese Benennung
  zu Testzwecken ungew\"{o}hnlich lang ist}{diffbarstetig}%
  Eine Funktion $f:I\to\mathbb{R}$ ist in $x_0\in I$ stetig, wenn $f$ in
  $x_0$ differenzierbar ist.
\end{Theorem}
```

The following theorem is numbered as Theorem 10.3.1 and referenced with the marker theo:diffbarstetig.

Theorem 10.3.1: Differenzierbarkeit bedingt Stetigkeit, wobei diese Benennung zu Testzwecken ungewöhnlich lang ist

Eine Funktion $f : I \to \mathbb{R}$ ist in $x_0 \in I$ stetig, wenn f in x_0 differenzierbar ist.

```
The following definition is numbered as Definition \ref{def:diffbarkeit} and
referenced with the marker \texttt{def:diffbarkeit}.\bigskip

\begin{Definition}{Differenzierbarkeit}{diffbarkeit}
  Eine Funktion $f:~I\to\mathbb{R}$ auf einem Intervall $I$ hei\ss{}t in
  $x_0\in I$ differenzierbar oder linear approximierbar,
  wenn der Grenzwert
  \begin{equation*}
  \lim\limits_{x\to x_0}\frac{f(x)-f(x_0)}{x-x_0}=
  \lim\limits_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h}
  \end{equation*}
  existiert. Bei Existenz hei\ss{}t dieser Grenzwert Ableitung
  oder Differentialquotient von $f$ in $x_0$ und man
  schreibt f\"{u}r ihn
  \begin{equation*}
  f'(x_0)\quad\text{oder}\quad\frac{df}{dx}(x_0).
  \end{equation*}
\end{Definition}
```

The following definition is numbered as Definition 10.3.2 and referenced with the marker `def:diffbarkeit`.

Definition 10.3.2: Differenzierbarkeit

Eine Funktion $f: I \to \mathbb{R}$ *auf einem Intervall* I *heißt in* $x_0 \in I$ *differenzierbar oder linear approximierbar, wenn der Grenzwert*

$$\lim_{x\to x_0} \frac{f(x)-f(x_0)}{x-x_0} = \lim_{h\to 0} \frac{f(x_0+h)-f(x_0)}{h}$$

existiert. Bei Existenz heißt dieser Grenzwert Ableitung oder Differentialquotient von f *in* x_0 *und man schreibt für ihn*

$$f'(x_0) \quad \text{oder} \quad \frac{df}{dx}(x_0).$$

```
The following corollary is numbered as Corollary \ref{cor:nullstellen} and
referenced with the marker \texttt{cor:nullstellen}.\bigskip

\begin{Corollary}{Nullstellenexistenz}{nullstellen}
  Ist $f:[a,b]\to\mathbb{R}$ stetig und haben $f(a)$ und $f(b)$ entgegengesetzte
  Vorzeichen, also $f(a)f(b)<0$, so besitzt $f$ eine Nullstelle $x_0\in]a,b[$,
  also $f(x_0)=0$.
\end{Corollary}
```

The following corollary is numbered as Corollary 10.3.3 and referenced with the marker `cor:nullstellen`.

Corollary 10.3.3: Nullstellenexistenz

Ist $f:[a,b]\to\mathbb{R}$ *stetig und haben* $f(a)$ *und* $f(b)$ *entgegengesetzte Vorzeichen, also* $f(a)f(b)<0$*, so besitzt* f *eine Nullstelle* $x_0 \in]a,b[$*, also* $f(x_0)=0$.

```
\begin{Theorem}[boxrule=2mm,toptitle=-1.5mm,bottomtitle=-1.5mm]{%
    Hinreichende Bedingung f\"{u}r Wendepunkte}{wendehinreichend}%
  $f$ sei eine auf einem Intervall $]a,b[$ dreimal stetig differenzierbare Funktion.
  Ist $f''(x_0)=0$ in $x_0\in]a,b[$ und $f'''(x_0)\ne 0$, so ist
  $(x_0,f(x_0))$ ein Wendepunkt von $f$.
\end{Theorem}
```

Theorem 10.3.4: Hinreichende Bedingung für Wendepunkte

f sei eine auf einem Intervall $]a,b[$ dreimal stetig differenzierbare Funktion. Ist $f''(x_0) = 0$ in $x_0 \in]a,b[$ und $f'''(x_0) \neq 0$, so ist $(x_0, f(x_0))$ ein Wendepunkt von f.

```
% \usepackage{varioref}
% \usepackage{cleveref}
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=plain apart,label type=theorem,enhanced,frame hidden,
  boxrule=2mm,titlerule=0mm,toptitle=1mm,bottomtitle=1mm,
  fonttitle=\bfseries\large,fontupper=\normalsize,
  coltitle=green!35!black,colbacktitle=green!15!white,
  colback=green!50!yellow!15!white,borderline={1pt}{0pt}{green!25!blue},
  }{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{meanvaluetheorem}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}

\medskip
Here, |cleveref| support is used to reference \Cref{theo:meanvaluetheorem}
on \Cpageref{theo:meanvaluetheorem}. This \namecref{theo:meanvaluetheorem}
can also be referenced by |\Vref| resulting in \Vref{theo:meanvaluetheorem}.
```

Theorem 10.3.5 (Mittelwertsatz für n Variable)

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^{\top}(x - x_0)$$

Here, `cleveref` support is used to reference Theorem 10.3.5 on Page 250. This theorem can also be referenced by `\Vref` resulting in Theorem 10.3.5.

Note that `/tcb/label type` → P. 79 was used in the example above to feed `cleveref` [5] with the needed name information.

```
Here, using |\Vref| resulting in \Vref{theo:meanvaluetheorem} is more interesting\ldots
```

Here, using \Vref resulting in Theorem 10.3.5 on page 250 is more interesting...

```
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=change apart,enhanced,arc=0mm,outer arc=0mm,
  boxrule=0mm,toprule=1mm,bottomrule=1mm,left=1mm,right=1mm,
  titlerule=0mm,toptitle=0mm,bottomtitle=1mm,top=0mm,
  colframe=red!50!black,colback=red!5!white,coltitle=red!50!black,
  title style={top color=yellow!50!white,bottom color=red!5!white,
    middle color=yellow!50!white},
  fonttitle=\bfseries\sffamily\normalsize,fontupper=\normalsize\itshape,
  }{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n2}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

10.3.6 Theorem (Mittelwertsatz für n Variable)

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

```
% \usepackage{varwidth}   \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
{enhanced,frame empty,interior empty,colframe=ForestGreen!50!white,
  coltitle=ForestGreen!50!black,fonttitle=\bfseries,colbacktitle=ForestGreen!15!white,
  borderline={0.5mm}{0mm}{ForestGreen!15!white},
  borderline={0.5mm}{0mm}{ForestGreen!50!white,dashed},
  attach boxed title to top center={yshift=-2mm},
  boxed title style={boxrule=0.4pt},varwidth boxed title}{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n3}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

Theorem 10.3.7: Mittelwertsatz für n Variable

Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

You need more attention for your theorems? Here, you are ...

```
% tcbuselibrary{skins}  % preamble
\begin{Theorem}[enhanced,
    fuzzy halo=3mm with yellow,
    fuzzy halo=2mm with red,
    fuzzy halo=1mm with yellow,
    watermark color=red!35!white,
    watermark text={Overacting\\Fundamental Theorem}]%
  {Fundamental Theorem of Theorems}{fundamental}%
  \lipsum[1-2]
\end{Theorem}
```

Theorem 10.3.8: Fundamental Theorem of Theorems

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Overacting
Fundamental Theorem

Let's try a more conservative approach:

```
% \tcbuselibrary{skins}
\newtcbtheorem[use counter from=Definition]{YetAnotherTheorem}{Theorem}%
  {theorem style=plain,enhanced,colframe=blue!50!black,colback=yellow!20!white,
   coltitle=red!50!black,fonttitle=\upshape\bfseries,fontupper=\itshape,
   drop fuzzy shadow=blue!50!black!50!white,boxrule=0.4pt}{theo}

\begin{YetAnotherTheorem}{Mittelwertsatz f\"{u}r $n$ Variable}{mittelwertsatz_n4}%
  Es sei $n\in\mathbb{N}$, $D\subseteq\mathbb{R}^n$ eine offene Menge und
  $f\in C^{1}(D,\mathbb{R})$. Dann gibt es auf jeder Strecke
  $[x_0,x]\subset D$ einen Punkt $\xi\in[x_0,x]$, so dass gilt
  \begin{equation*}
  f(x)-f(x_0) = \operatorname{grad} f(\xi)^{\top}(x-x_0)
  \end{equation*}
\end{YetAnotherTheorem}
```

Theorem 10.3.9 (Mittelwertsatz für n Variable): *Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt*

$$f(x) - f(x_0) = \operatorname{grad} f(\xi)^\top (x - x_0)$$

11 Library LIB breakable

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{breakable}
```

11.1 Technical Overview

The library LIB breakable supports the automatic breaking of a `tcolorbox`. This feature is enabled by `/tcb/breakable` → P. 255 and disabled by `/tcb/unbreakable` → P. 256.

If a `tcolorbox` is set to be `/tcb/breakable` → P. 255, then the following algorithm is executed:

1. The box content is read to a box register similar but not identical to the unbreakable case.
2. If the total box fits into the current page, it is shipped out visibly unbroken and the algorithm stops.

   Unbroken Box

   The box. unbroken

3. Otherwise, it is checked if at least `/tcb/lines before break` → P. 256 of the upper box can be placed on the current page. If not, a page break is inserted and the algorithm goes back to Step 2.
4. Now, the *break sequence* starts. The upper box part or the lower box part is split such that it fits into the current page. The fitting part is named *first part* of the *break sequence* and shipped out.

   Broken Box

   The box. first

5. If the remaining content of the total box fits into the current page, the algorithm continues with Step 7, else with Step 6.
6. The upper box part or the lower box part is split such that it fits into the current page. The fitting part is named *middle part* of the *break sequence* and shipped out. Then, the algorithm goes back to Step 5.

   The box. middle

7. The remaining part is named *last part* of the *break sequence* and shipped out. The algorithm stops.

   The box. last

The algorithm takes care that the optional segmentation line never appears at the end of a box. The optional lower box part is also checked to have at least `/tcb/lines before break` → P. 256.

In principal, all boxes of the *break sequence* share the same geometric parameters. The differences are:

- The given `/tcb/before` [→ P. 65] and `/tcb/after` [→ P. 65] values are used only before the *first* and after the *last* part of the *break sequence*.
- A special behavior between the parts of the *break sequence* can be given by `/tcb/toprule at break` [→ P. 259], `/tcb/bottomrule at break` [→ P. 259], `/tcb/enlarge top at break by` [→ P. 68], and `/tcb/enlarge bottom at break by` [→ P. 68].
- The `/tcb/skin` [→ P. 89] decides *how* the *first*, *middle*, and *last* part look like. Actually, every part type has its own skin given by the options `/tcb/skin first` [→ P. 89], `/tcb/skin middle` [→ P. 89], and `/tcb/skin last` [→ P. 89]. Typically, these options are set automatically by the main skin, see Subsection 11.6 from page 264.

11.2 Limitations and Known Bugs

- The maximal total height of the upper and of the lower part of normal breakable `tcolorbox`es is about 65536pt (ca. 2300cm) apiece. If such a part gets longer, the output will get buggy without warning. For very oversized boxes which are longer than 65536pt, use the `unlimited` value for `/tcb/breakable` [→ P. 255]. With the `unlimited` setting, the applied algorithm has (virtually) no height limit for boxes, but very likely the compiler memory will have to be increased for boxes longer than 300 pages (depending on compiler settings and box content). But it is recommended to use `unlimited` for critical large boxes only.
- You can nest an unbreakable `tcolorbox` inside another `tcolorbox`, even inside a breakable one. But you cannot not nest a breakable box inside a breakable box. The `/tcb/breakable` [→ P. 255] key for a nested box is ignored automatically[4], i. e. inner boxes are always unbreakable.

  After all, in the unlikely case you really want to have the nested box to be breakable, use `/tcb/enforce breakable` [→ P. 256] for the nested box[5]. **But, a breakable box inside a breakable box will usually give a mess.**
- If your text content contains some text color changing commands, your color will not survive the break to the next box.

[4] Until `tcolorbox` 3.04, the `/tcb/breakable` [→ P. 255] key was not ignored for nested boxes.
[5] `/tcb/enforce breakable` [→ P. 256] acts like `/tcb/breakable` [→ P. 255] until `tcolorbox` 3.04.

11.3 Main Option Keys

`/tcb/breakable=true|false|unlimited` (default `true`, initially `false`)

Allows the `tcolorbox` to be breakable. If the box is larger than the available space at the current page, the box is automatically broken and continued to the next next page. All sorts of `tcolorbox` can be made breakable. It depends on the skin how the breaking looks like. If you do not know better, use `/tcb/enhanced` → P. 150 for breaking a box. The parts of the *break sequence* are numbered by the counter `tcbbreakpart`.

- `false`: Sets the `tcolorbox` to be unbreakable.
- `true`: Breaks the `tcolorbox` from one page to another. The maximal total height of the upper and of the lower part is about 65536pt (ca. 2300cm or ca. 90 pages) apiece.
- `unlimited`: Experimental code for unlimited total height of breakable boxes. For boxes longer than 300 pages (or even shorter ones) the compiler memory will have to be increased.

```
% \usepackage{lipsum}  % preamble
\tcbset{enhanced jigsaw,colback=red!5!white,colframe=red!75!black,
  watermark color=yellow!25!white,watermark text=\arabic{tcbbreakpart},
  fonttitle=\bfseries}

\begin{tcolorbox}[breakable,title=My breakable box]
\lipsum[1-6]
\end{tcolorbox}
```

My breakable box

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea

dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.
Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

`/tcb/unbreakable` (no value, initially set)

Sets the `tcolorbox` to be unbreakable.

`/tcb/enforce breakable` (no value)

A `tcolorbox` inside a `tcolorbox` is automatically set to be unbreakable. Using `/tcb/breakable` → P. 255 on such an inner box has no effect. If one *really* wants the inner box to be breakable, use `/tcb/enforce breakable`. **This will usually give a mess of shattered boxes. You are advised to not use this option.**

Note that `/tcb/enforce breakable` has the functionality that `/tcb/breakable` → P. 255 had until package version 3.04 and exists for backward compatibility.

`/tcb/title after break=`⟨*text*⟩ (no default, initially empty)

The `/tcb/title` → P. 14 is used only for the *first* part of a *break sequence*. Use `title after break` to create a heading line with ⟨*text*⟩ as content for all following parts.

`/tcb/notitle after break` (no value, initially set)

Removes the title line or following parts in a *break sequence* if set before.

`/tcb/adjusted title after break=`⟨*text*⟩ (style, no default, initially unset)

Works like `/tcb/adjusted title` → P. 14 but applied to `/tcb/title after break`.

`/tcb/lines before break=`⟨*number*⟩ (no default, initially 2)

Assures that the given ⟨*number*⟩ of lines of the upper box part or the lower box part are placed before a break happens.

/tcb/break at=⟨*length*⟩/⟨*length*⟩/.../⟨*length*⟩ (no default, initially `0pt`)

Defines break points at the given ⟨*length*⟩ values. The first ⟨*length*⟩ defines the (maximal) height of the first partial box, the second ⟨*length*⟩ defines the (maximal) height of the second partial box, and so on. The last ⟨*length*⟩ value is applied to all following partial boxes if any. Setting a length to `0pt` means that the naturally available space is used for breaking.

```
% \usepackage{multicol,lipsum}
\begin{multicols}{3}\footnotesize
Breakable boxes inside a |multicols| environment need special attendance.
They are broken by default at |\textheight|.
The |break at| option can be used to insert better break points by hand.
\begin{tcolorbox}[enhanced jigsaw,size=small,vfill before first,
  colframe=red,colback=yellow!10!white,before title=\raggedright,
  title={Broken box inside a |multicols| environment},fonttitle=\bfseries,
  enforce breakable,% use only breakable in the real world!
  pad at break=1mm,break at=3cm/6.3cm ]
\lipsum[1]
\end{tcolorbox}
\refKey{/tcb/height fixed for} may also be considered for |multicols| environments.
\end{multicols}
```

Breakable boxes inside a `multicols` environment need special attendance. They are broken by default at `\textheight`. The `break at` option can be used to insert better break points by hand.

Broken box inside a `multicols` environment

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`/tcb/height fixed for` → P. 260 may also be considered for `multicols` environments.

/tcb/enlargepage=⟨*length*⟩/⟨*length*⟩/.../⟨*length*⟩ (no default, initially `0pt`)

Inserts a `\enlargethispage{`⟨*length*⟩`}` to the pages of the break sequence, i. e. allows one to enlarge (or shrink) partial boxes. The first ⟨*length*⟩ is applied to the first partial box, the second ⟨*length*⟩ is applied to the second partial box, and so on. The last ⟨*length*⟩ value is applied to all following partial boxes if any. Note that floating boxes will not be enlarged.

```
\begin{tcolorbox}[breakable,enlargepage=0mm/\baselineskip/2\baselineskip/0mm,...
```

The example code enlarged the second partial box by one line, the third partial box by two lines, and all following parts are not enlarged.

If an automated page break occures before the first partial box, the page enlargement is applied to the page before the first partial box *and* again to the page of the first partial box. Insert a manual break to prevent this.

In general, `enlargepage` should be used at the final stage of a document for fine-tuning only.

`/tcb/enlargepage flexible`=⟨*length*⟩ (no default, initially `0pt`)

This allows an automated page enlargement for up to ⟨*length*⟩. The algorithm can use this to avoid breaking a box, if there is anough room after enlargement. Also, the *last* partial box of a break sequence may be enlarged to avoid further breaking.

Note that this potential enlargement is *additive* to settings of `/tcb/enlargepage` → P. 257. But `/tcb/enlargepage flexible` overwrites settings of `/tcb/pad before break*` → P. 259 or `/tcb/pad at break*` → P. 260.

```
% The following setting hinders orphan lines for the last partial box
\tcbset{enlargepage flexible=\baselineskip}
```

`/tcb/shrink break goal`=⟨*length*⟩ (no default, initially `0pt`)

This is an emergency parameter if the break algorithm produces unpleasant breaks. It shrinks the goal height of the current box part by ⟨*length*⟩ which may result in smaller boxes. Never use negative values. *Usually, this option will never be needed at all.*

11.4 Option Keys for the Break Appearance

/tcb/toprule at break=⟨*length*⟩ (no default, initially 0.5mm)
Sets the line width of the top rule to ⟨*length*⟩ *if* the box is /tcb/breakable → P. 255. In this case, it is applied to *middle* and *last* parts in a break sequence. Note that /tcb/toprule → P. 23 overwrites this value if used afterwards.

/tcb/bottomrule at break=⟨*length*⟩ (no default, initially 0.5mm)
Sets the line width of the bottom rule to ⟨*length*⟩ *if* the box is /tcb/breakable → P. 255. In this case, it is applied to *first* and *middle* parts in a break sequence. Note that /tcb/bottomrule → P. 23 overwrites this value if used afterwards.

/tcb/topsep at break=⟨*length*⟩ (no default, initially 0mm)
Additional vertical space of ⟨*length*⟩ which is added at the top of *middle* and *last* parts in a break sequence. In general, it is not advisable to change this value if these parts start with a rule or a title.

/tcb/bottomsep at break=⟨*length*⟩ (no default, initially 0mm)
Additional vertical space of ⟨*length*⟩ which is added at the bottom of *first* and *middle* parts in a break sequence. In general, it is not advisable to change this value if these parts end with a rule.

/tcb/pad before break=⟨*length*⟩ (style, no default, initially 3.5mm)
Sets the total amount of vertical space after the text content and before the break point to ⟨*length*⟩. This style sets /tcb/toprule at break to 0pt and changes /tcb/topsep at break as required. In general, it is not advisable to change this value if the *middle* and *last* parts in a break sequence start with a rule or a title.

/tcb/pad before break*=⟨*length*⟩ (style, no default)
Sets /tcb/pad before break to ⟨*length*⟩ and /tcb/enlargepage flexible → P. 258 to an appropriate value such that empty closing frames are avoided.

/tcb/pad after break=⟨*length*⟩ (style, no default, initially 3.5mm)
Sets the total amount of vertical space after the break point and before the text content to ⟨*length*⟩. This style sets /tcb/bottomrule at break to 0pt and changes /tcb/bottomsep at break as required. In general, it is not advisable to change this value if the *first* and *middle* parts in a break sequence end with a rule.

/tcb/pad at break=⟨*length*⟩ (style, no default, initially 3.5mm)
Abbreviation for setting ⟨*length*⟩ to /tcb/pad before break and /tcb/pad after break.

```
% \usepackage{lipsum}  % preamble
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries}

\begin{tcolorbox}[enhanced jigsaw,breakable,pad at break=0mm,
  title={For this box, the pad space at the break point is set to 0mm}]
  \lipsum[1-2]
\end{tcolorbox}
```

For this box, the pad space at the break point is set to 0mm

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est,

iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/pad at break*=`⟨*length*⟩ (style, no default)

Sets `/tcb/pad at break` → P. 259 to ⟨*length*⟩ and `/tcb/enlargepage flexible` → P. 258 to an appropriate value such that empty closing frames are avoided.

`/tcb/pad at break` → P. 259 or `/tcb/pad at break*` should be used as very last option in an option list, because they adapt other settings.

Also see `/tcb/enlarge top at break by` → P. 68 and `/tcb/enlarge bottom at break by` → P. 68.

`/tcb/height fixed for=`⟨*part*⟩ (no default, initially `none`)

When certain amount of space is available for a partial box of a break sequence, the partial box typically is smaller than this space (depending on the box content). For given ⟨*part*⟩(s), the height can be set to all available space.

- `none`: Every partial `tcolorbox` is set with its natural height.
- `first`: The *first* partial box is set to a height which matches the available space.
- `middle`: All *middle* partial boxes are set to a height which matches the available space.
- `last`: The *last* partial box is set to a height which matches the available space.
- `first and middle`: The *first* and all *middle* partial boxes are set to a height which matches the available space.
- `middle and last`: All *middle* partial boxes and the *last* partial box are set to a height which matches the available space.
- `all`: All partial boxes are set to a height which matches the available space.

If the box keeps unbroken, this option is not applied. See `/tcb/height` → P. 39 for setting a fixed height for unbroken boxes. See `/tcb/height fill` → P. 41 for giving unbroken boxes maximum height.

`/tcb/vfill before first=true|false` (default `true`, initially `false`)

Inserts a `\vfill` at the begin of the *first* partial box to move this partial box to the end of the current page. This may be used as an alternative to `/tcb/height fixed for=first` to get justified columns or pages. The `\vfill` is not inserted, if the box gets not actually broken.

11.5 Breakable boxes and the `multicol` package

Unbreakable `tcolorboxes` can be used without special care inside a `multicols` environment from the `multicol` package [9].
Since version 3.10, a breakable `tcolorbox` detects, if it is used inside a `multicols` environment. But choosing break points for a breakable box cannot be done by the balancing routine of `multicols`. By default, boxes will break at `\textheight`. To get pleasant results, use the `/tcb/break at` → P. 257 and `/tcb/height fixed for` → P. 260 options.

```
% \usepackage{lipsum,multicol}  % preamble
\small
\begin{multicols}{2}
  \lipsum[1]
  \begin{tcolorbox}[enhanced jigsaw,breakable,size=title,
    colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
    title=My breakable box,pad at break=1mm, break at=7.5cm/0pt ]
  \lipsum[2-4]
  \end{tcolorbox}
  \lipsum[4]
\end{multicols}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My breakable box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

This example is already set inside a `multicols` environment. This time, a *middle* part has full `\textheight`. `/tcb/height fixed for` → P. 260 is used to spread this box part over the full height toi align with neighboring columns.

```
% \usepackage{lipsum,multicol}
\lipsum[1]
\begin{tcolorbox}[enhanced ↩
      ↪jigsaw,breakable,
  size=title,
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  title=My breakable box,
  pad at break=2mm,
  break at=8.2cm/0pt,
  height fixed for=middle ]
\lipsum[2-7]
\end{tcolorbox}
\lipsum[8]
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

My breakable box

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi

quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

The following example has a `\tcolorbox` which fills the `\multicols` environment completely. Here, `/tcb/height fixed for` → P. 260 is used to give all three columns the full height. Note that the appropriate `/tcb/break at` → P. 257 value is not computed automatically but set manually.

```
% \usepackage{lipsum,multicol}  % preamble
\small
\begin{multicols}{3}
  \begin{tcolorbox}[enhanced jigsaw,breakable,size=small,
    colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
    title=My breakable box,pad at break=2mm,drop fuzzy shadow,
    height fixed for=all, break at=11.4cm ]
  \lipsum[1-3]
  \end{tcolorbox}
\end{multicols}
```

My breakable box

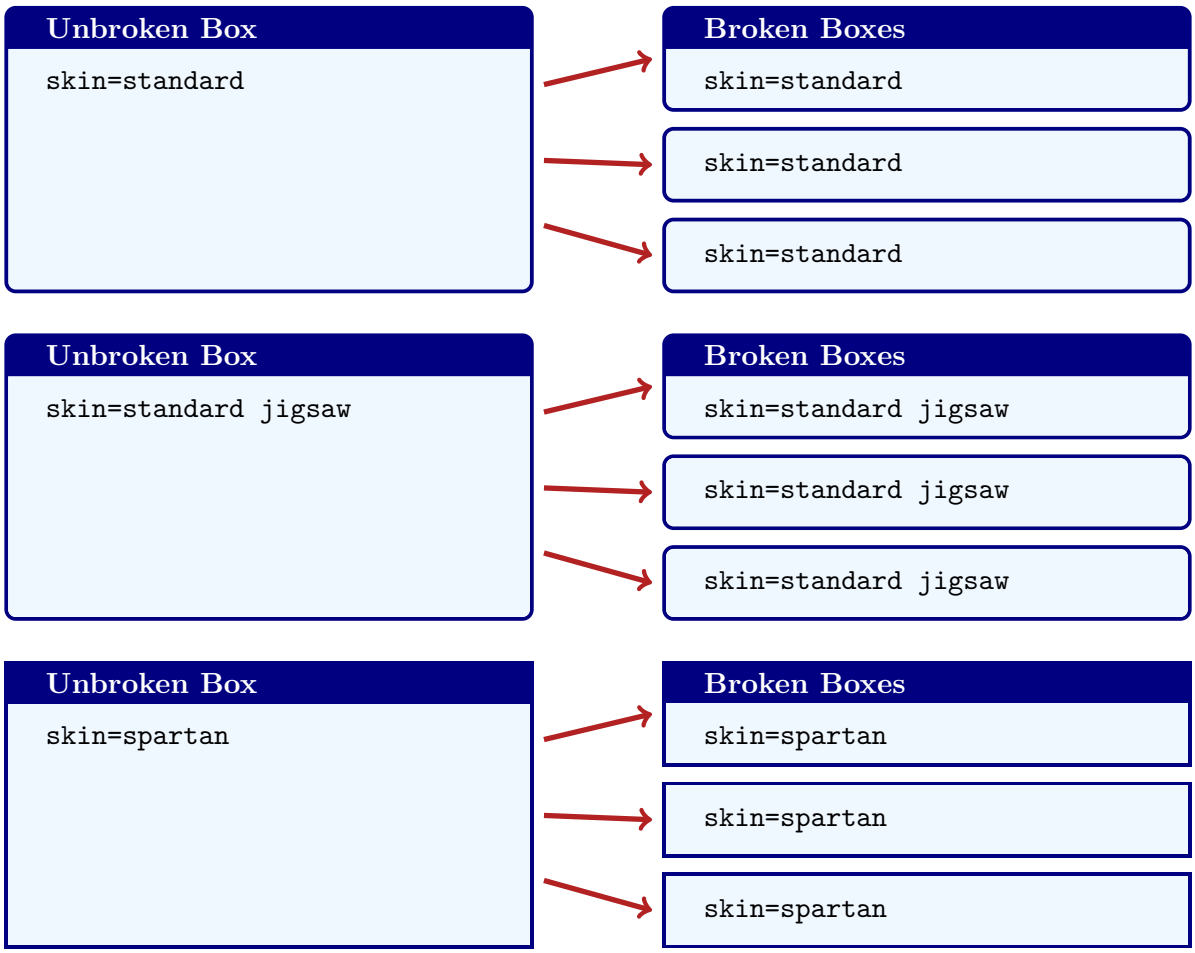
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

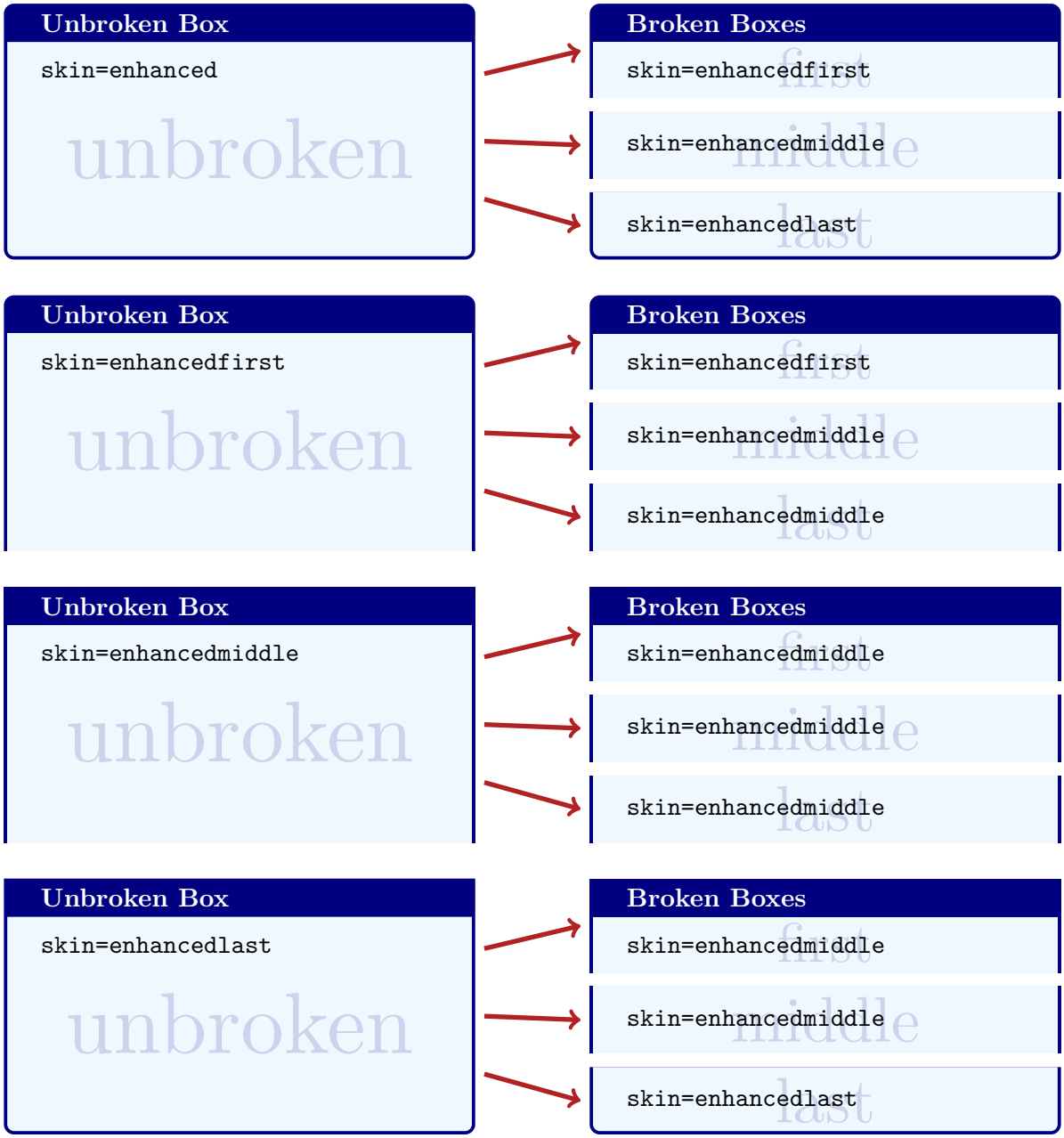
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

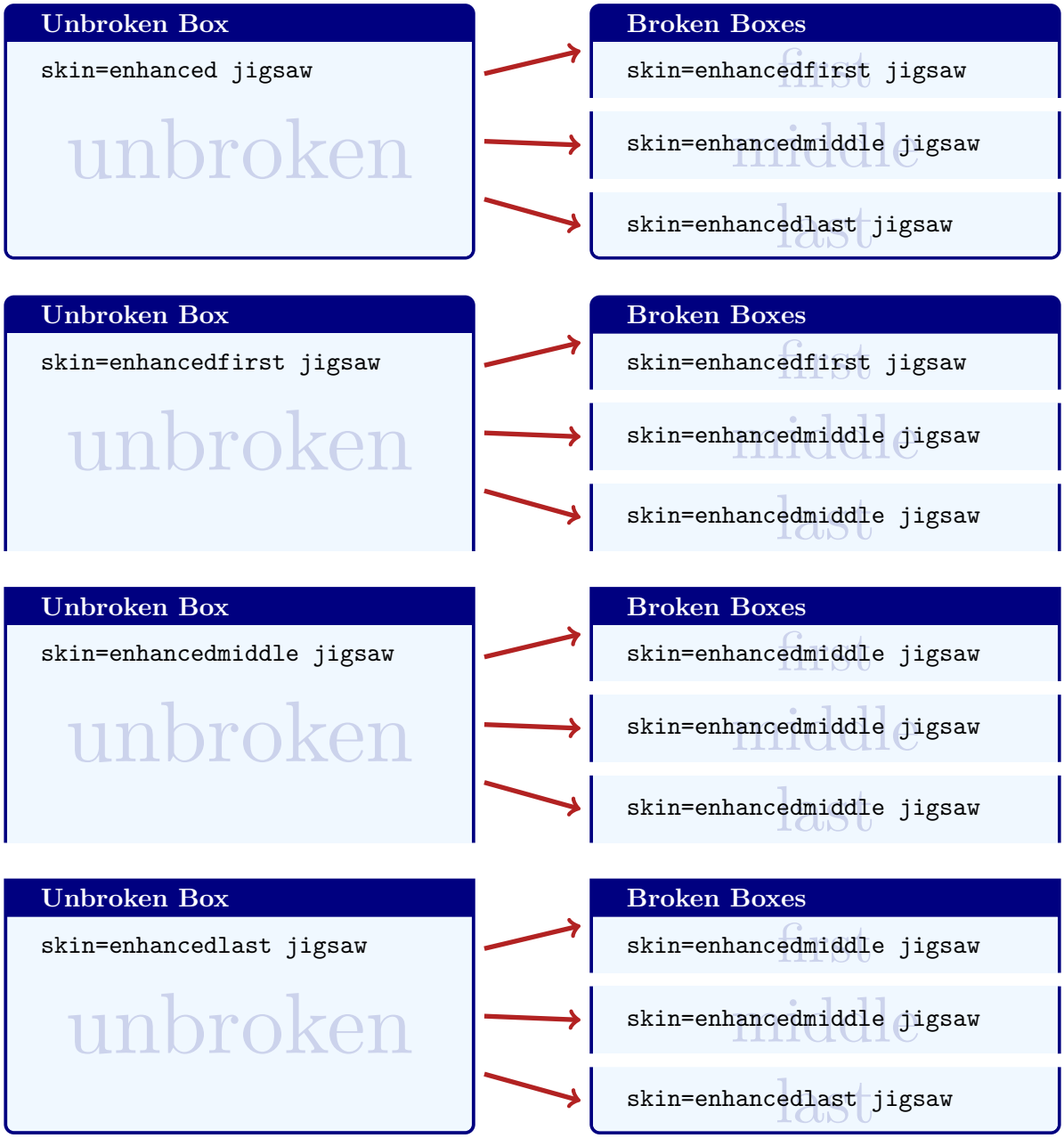
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

11.6 Break Sequence for the Skins

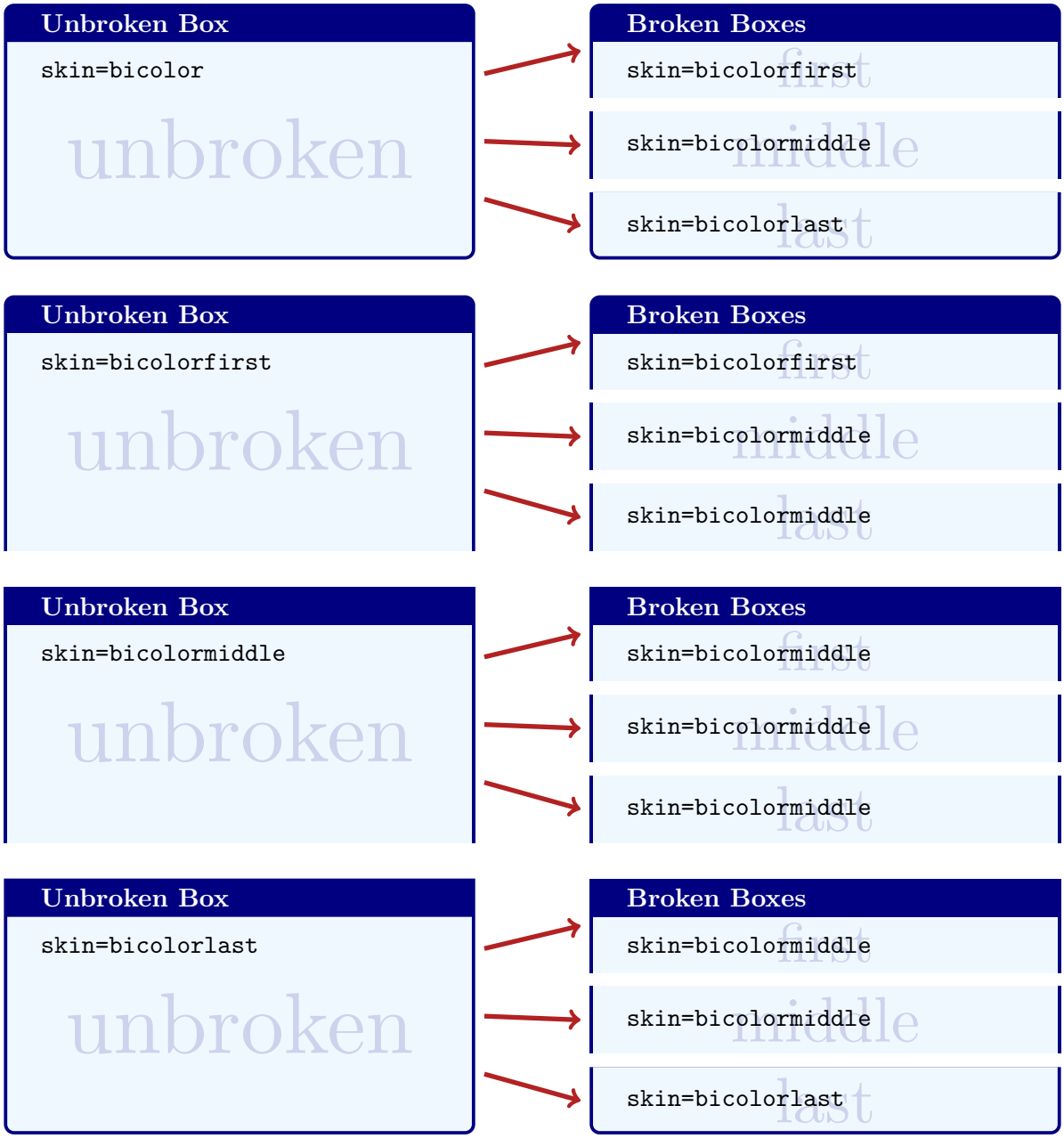
The following diagrams document the *break sequence* for different skins. Depending on the main skin of a `tcolorbox`, the actual skins of the *break sequence* parts are displayed.

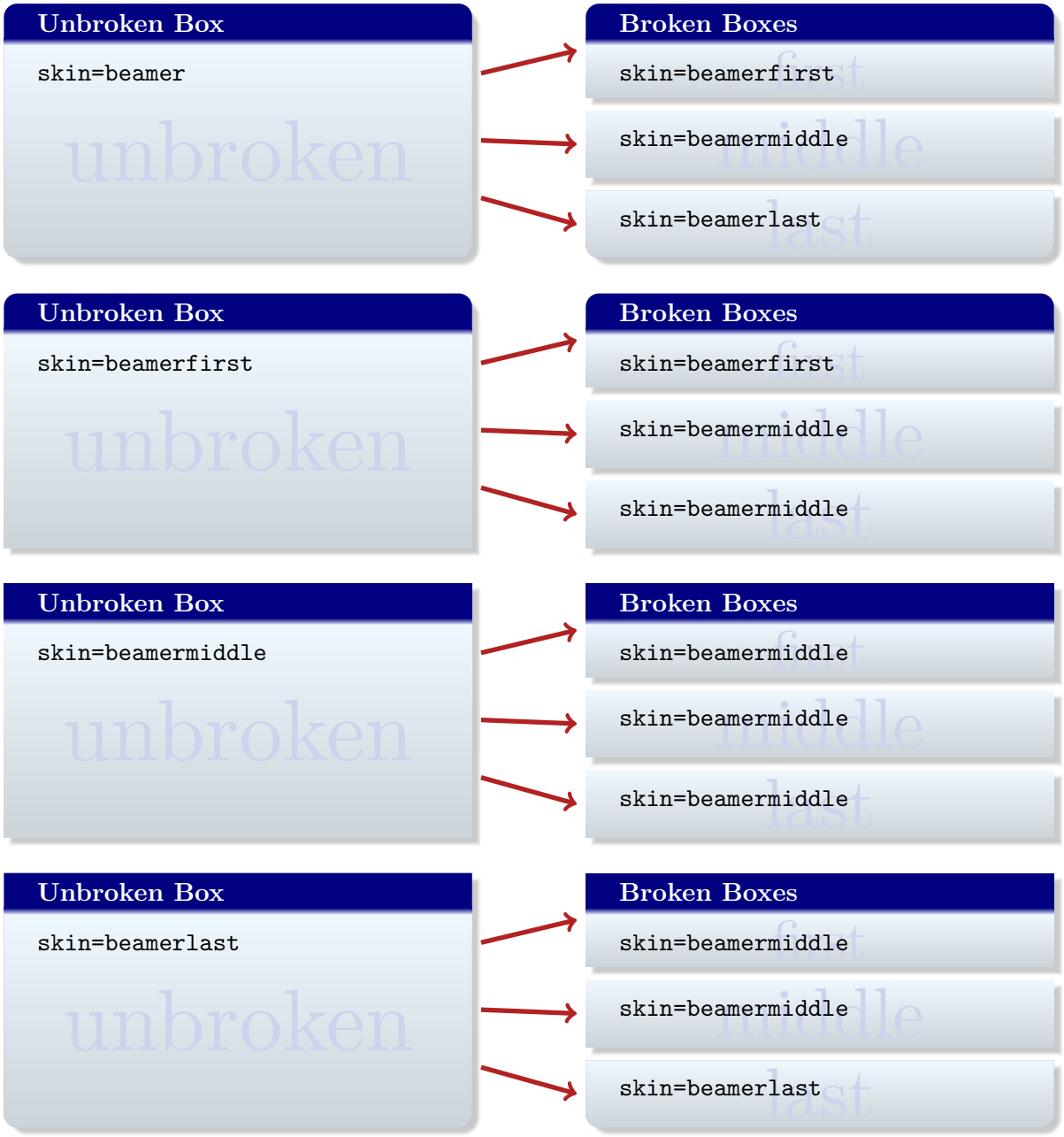


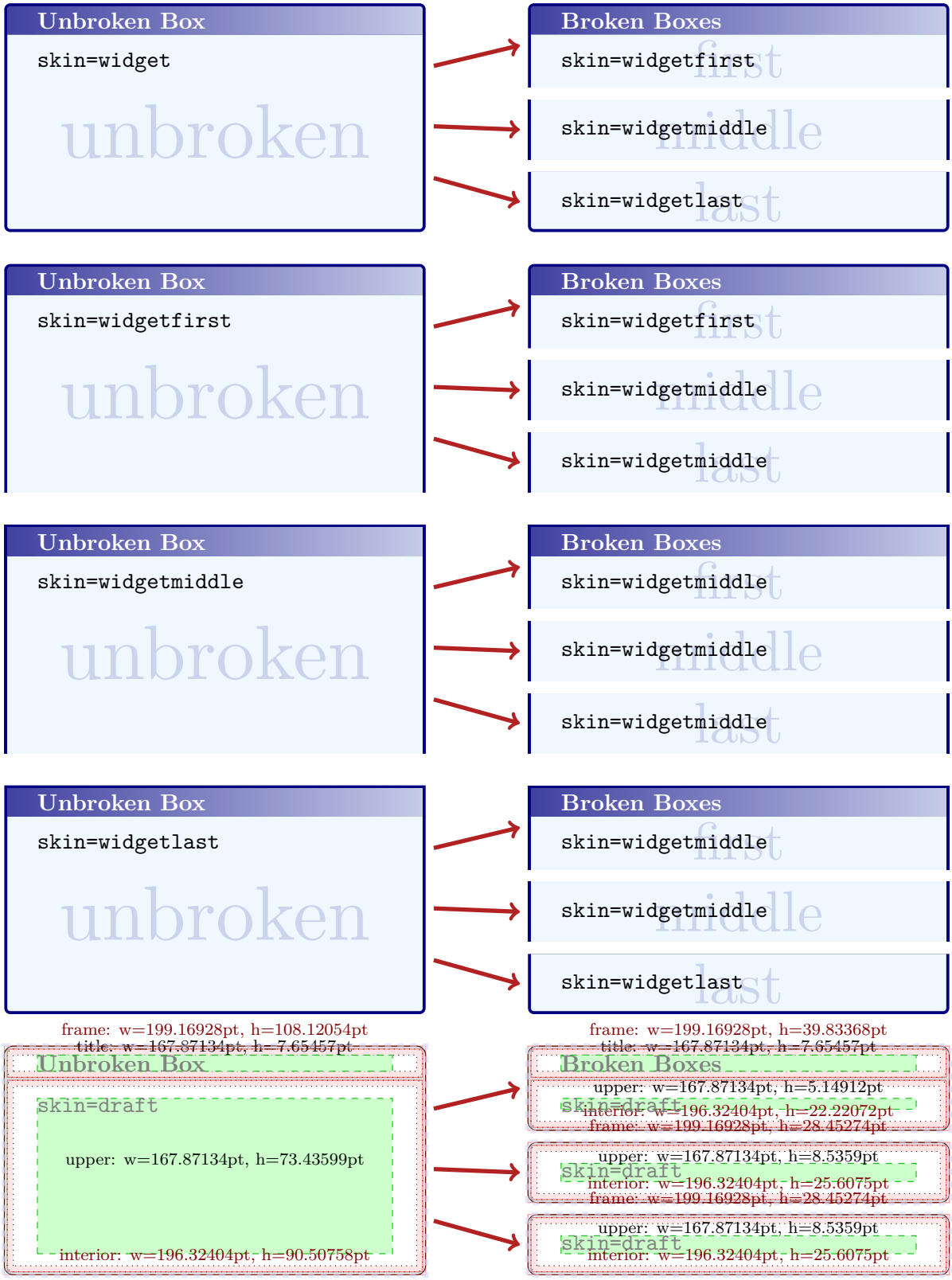

Unbroken Box
skin=enhanced
unbroken
Broken Boxes
skin=enhancedfirst
first
skin=enhancedmiddle
middle
skin=enhancedlast
last
Unbroken Box
skin=enhancedfirst
unbroken
Broken Boxes
skin=enhancedfirst
first
skin=enhancedmiddle
middle
skin=enhancedmiddle
last
Unbroken Box
skin=enhancedmiddle
unbroken
Broken Boxes
skin=enhancedmiddle
first
skin=enhancedmiddle
middle
skin=enhancedmiddle
last
Unbroken Box
skin=enhancedlast
unbroken
Broken Boxes
skin=enhancedmiddle
first
skin=enhancedmiddle
middle
skin=enhancedlast
last

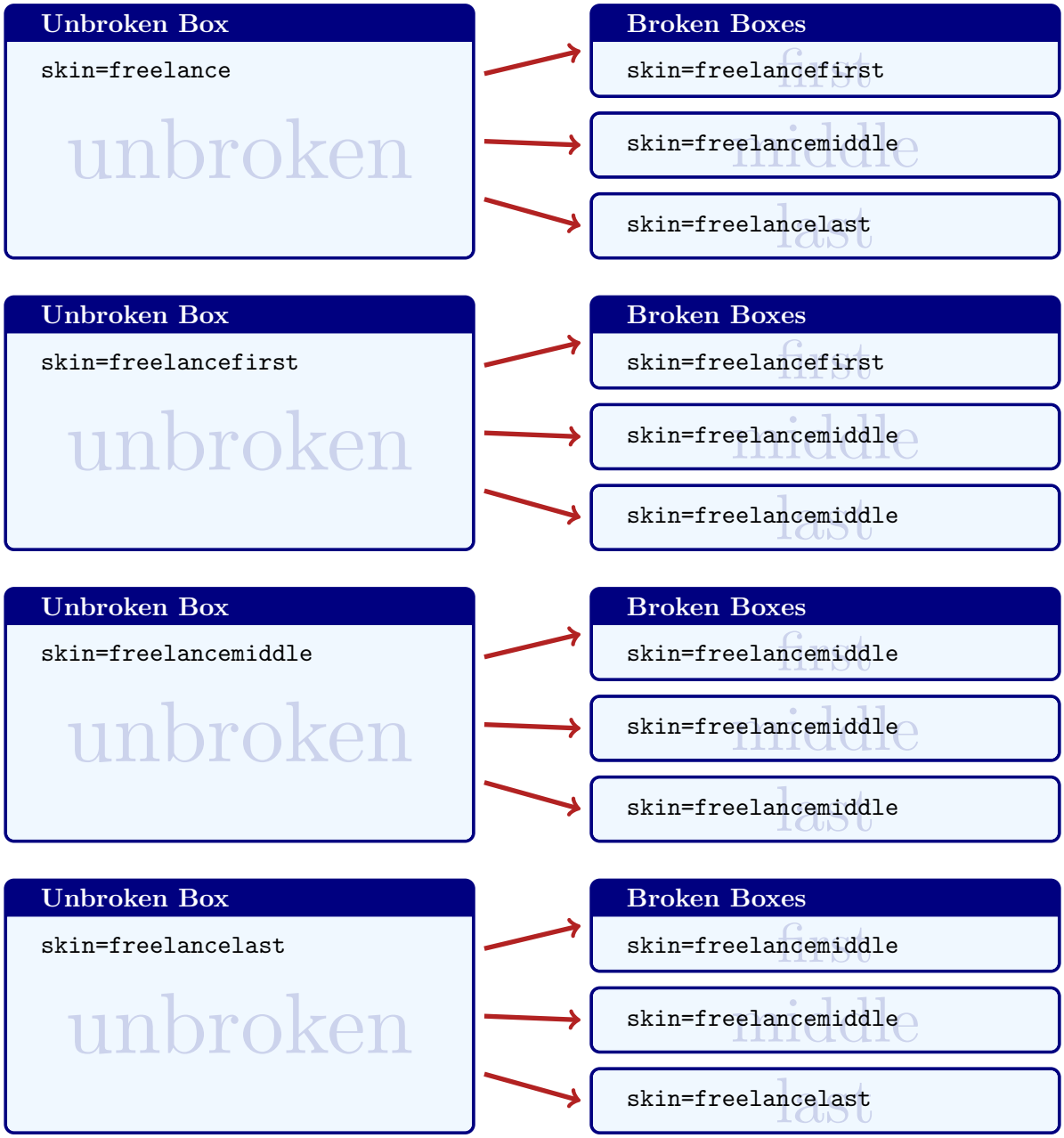

Unbroken Box
skin=enhanced jigsaw
unbroken
Broken Boxes
skin=enhancedfirst jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedlast jigsaw
Unbroken Box
skin=enhancedfirst jigsaw
unbroken
Broken Boxes
skin=enhancedfirst jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
Unbroken Box
skin=enhancedmiddle jigsaw
unbroken
Broken Boxes
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
Unbroken Box
skin=enhancedlast jigsaw
unbroken
Broken Boxes
skin=enhancedmiddle jigsaw
skin=enhancedmiddle jigsaw
skin=enhancedlast jigsaw

skin=empty
unbroken
skin=emptyfirst
first
skin=emptymiddle
middle
skin=emptylast
last
skin=emptyfirst
unbroken
skin=emptyfirst
first
skin=emptymiddle
middle
skin=emptymiddle
last
skin=emptymiddle
unbroken
skin=emptymiddle
first
skin=emptymiddle
middle
skin=emptymiddle
last
skin=emptylast
unbroken
skin=emptymiddle
first
skin=emptymiddle
middle
skin=emptylast
last


Unbroken Box
skin=bicolor
unbroken
Broken Boxes
skin=bicolorfirst
first
skin=bicolormiddle
middle
skin=bicolorlast
last
Unbroken Box
skin=bicolorfirst
unbroken
Broken Boxes
skin=bicolorfirst
first
skin=bicolormiddle
middle
skin=bicolormiddle
last
Unbroken Box
skin=bicolormiddle
unbroken
Broken Boxes
skin=bicolormiddle
first
skin=bicolormiddle
middle
skin=bicolormiddle
last
Unbroken Box
skin=bicolorlast
unbroken
Broken Boxes
skin=bicolormiddle
first
skin=bicolormiddle
middle
skin=bicolorlast
last


Unbroken Box
skin=beamer
unbroken
Broken Boxes
skin=beamerfirst
skin=beamermiddle
skin=beamerlast
Unbroken Box
skin=beamerfirst
unbroken
Broken Boxes
skin=beamerfirst
skin=beamermiddle
skin=beamermiddle
Unbroken Box
skin=beamermiddle
unbroken
Broken Boxes
skin=beamermiddle
skin=beamermiddle
skin=beamermiddle
Unbroken Box
skin=beamerlast
unbroken
Broken Boxes
skin=beamermiddle
skin=beamermiddle
skin=beamerlast


Unbroken Box
skin=widget
unbroken
Broken Boxes
skin=widgetfirst
skin=widgetmiddle
skin=widgetlast
first
middle
last
Unbroken Box
skin=widgetfirst
unbroken
Broken Boxes
skin=widgetfirst
skin=widgetmiddle
skin=widgetmiddle
first
middle
last
Unbroken Box
skin=widgetmiddle
unbroken
Broken Boxes
skin=widgetmiddle
skin=widgetmiddle
skin=widgetmiddle
first
middle
last
Unbroken Box
skin=widgetlast
unbroken
Broken Boxes
skin=widgetmiddle
skin=widgetmiddle
skin=widgetlast
first
middle
last
frame: w=199.16928pt, h=108.12054pt
title: w=167.87134pt, h=7.65457pt
Unbroken Box
skin=draft
upper: w=167.87134pt, h=73.43599pt
interior: w=196.32404pt, h=90.50758pt
frame: w=199.16928pt, h=39.83368pt
title: w=167.87134pt, h=7.65457pt
Broken Boxes
upper: w=167.87134pt, h=5.14912pt
skin=draft
interior: w=196.32404pt, h=22.22072pt
frame: w=199.16928pt, h=28.45274pt
upper: w=167.87134pt, h=8.5359pt
skin=draft
interior: w=196.32404pt, h=25.6075pt
frame: w=199.16928pt, h=28.45274pt
upper: w=167.87134pt, h=8.5359pt
skin=draft
interior: w=196.32404pt, h=25.6075pt

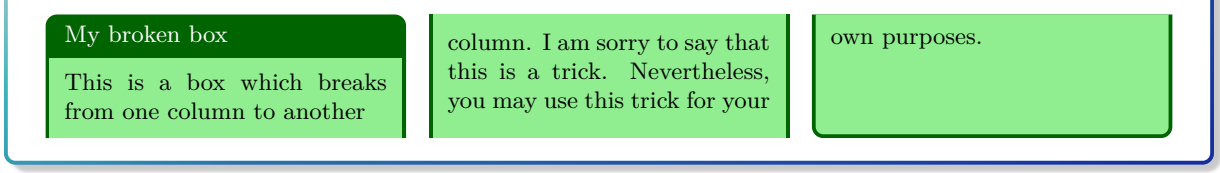

Unbroken Box
skin=freelance
unbroken
Broken Boxes
skin=freelancefirst
first
skin=freelancemiddle
middle
skin=freelancelast
last
Unbroken Box
skin=freelancefirst
unbroken
Broken Boxes
skin=freelancefirst
first
skin=freelancemiddle
middle
skin=freelancemiddle
last
Unbroken Box
skin=freelancemiddle
unbroken
Broken Boxes
skin=freelancemiddle
first
skin=freelancemiddle
middle
skin=freelancemiddle
last
Unbroken Box
skin=freelancelast
unbroken
Broken Boxes
skin=freelancemiddle
first
skin=freelancemiddle
middle
skin=freelancelast
last

11.7 Break by Hand (Faked Break)

See Section 11.5 on page 261 for *real* column breaks.

Since the appearance of broken boxes is done by skins, it is quite easy to 'fake a break'. For this, you actually don't need the LIB breakable library at all.

```
\tcbset{enhanced,equal height group=fakedbreak,
  colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-6mm)/3,nobeforeafter,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}
%
\begin{tcolorbox}[title=My broken box,skin=enhancedfirst]
This is a box which breaks from one column to another
\end{tcolorbox}\hfill
\begin{tcolorbox}[skin=enhancedmiddle]
column. I am sorry to say that this is a trick.
Nevertheless, you may use this trick for your
\end{tcolorbox}\hfill
\begin{tcolorbox}[skin=enhancedlast]
own purposes.
\end{tcolorbox}
```



12 Library LIB fitting

The library is loaded by a package option or inside the preamble by:

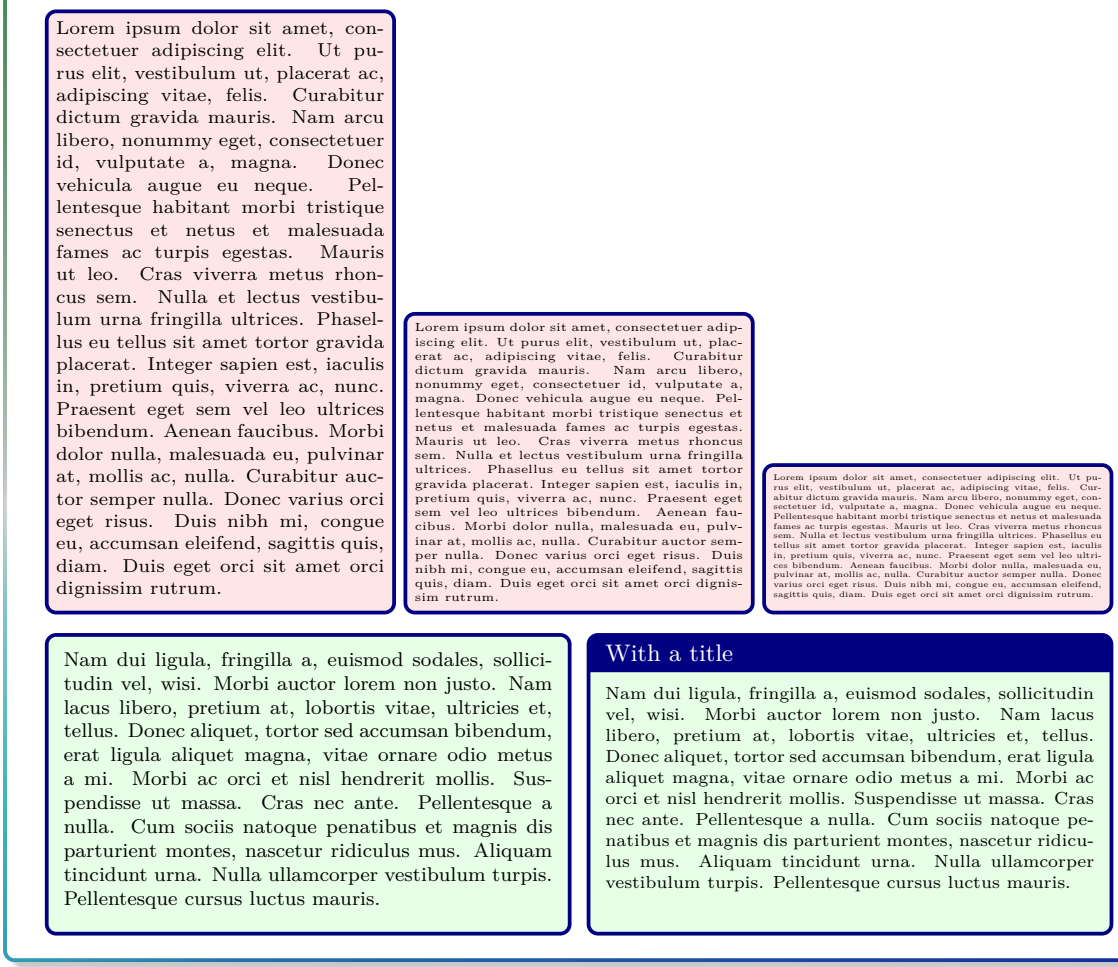
```
\tcbuselibrary{fitting}
```

12.1 Macros of the Library

`\tcboxfit[⟨options⟩]{⟨box content⟩}`

Creates a colored box where the given ⟨*box content*⟩ is fitted to the width and height of the box. A `tcboxfit` has to have a fixed height. If no fixed height is given, a square box is constructed. In principal, most ⟨*options*⟩ for a `tcolorbox` → P. 9 can be used for `\tcboxfit` with some restrictions. A `\tcboxfit` cannot have a lower part and cannot be broken.

```
% \usepackage{lipsum}
\tcbset{colframe=blue!50!black,colback=red!10!white,
        boxsep=0pt,top=1mm,bottom=1mm,left=1mm,right=1mm,
        nobeforeafter,width=(\linewidth-2mm)/3}

\tcboxfit[height=8cm]{\lipsum[1]}\hfill
\tcboxfit[height=4cm]{\lipsum[1]}\hfill
\tcboxfit[height=2cm]{\lipsum[1]}

\medskip
\tcbset{width=(\linewidth-2mm)/2,colback=green!10!white,boxsep=1mm}
\tcboxfit[height=4cm]{\lipsum[2]}\hfill
\tcboxfit[height=4cm,title=With a title]{\lipsum[2]}
```

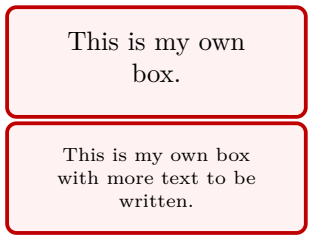


See Section 14.6 on page 304 for more elaborate methods to create new commands.

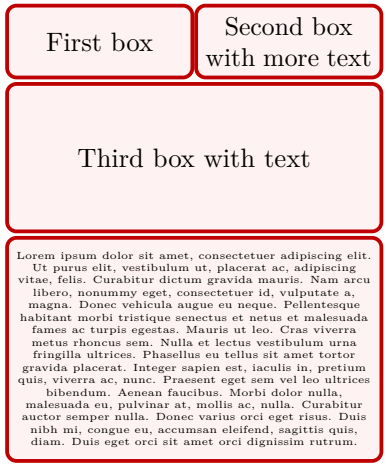
`\newtcboxfit[⟨init options⟩]{\⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Creates a new macro `\⟨name⟩` based on `\tcboxfit` → P. 273. Basically, `\newtcboxfit` operates like `\newcommand`. The new macro `\⟨name⟩` optionally takes ⟨*number*⟩+1 arguments, where ⟨*default*⟩ is the default value for the optional first argument. The ⟨*options*⟩ are given to the underlying `tcboxfit`. The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

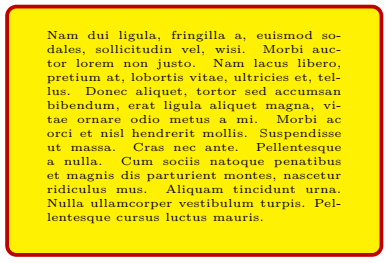
```
\newtcboxfit{\mybox}{colback=red!5!white,
  colframe=red!75!black,width=4cm,
  height=1.5cm,center upper}

\mybox{This is my own box.}\par
\mybox{This is my own box with more text
       to be written.}
```




```
% \usepackage{lipsum}
\newtcboxfit{\mybox}[2]{colback=red!5!white,
  colframe=red!75!black,fonttitle=\bfseries,
  boxsep=1mm,left=0mm,right=0mm,top=0mm,
  bottom=0mm,center upper,valign=center,
  nobeforeafter,width=#1,height=#2}

\mybox{2.5cm}{1cm}{First box}%
\mybox{2.5cm}{1cm}{Second box with more text}\\
\mybox{5cm}{2cm}{Third box with text}\\
\mybox{5cm}{3cm}{\lipsum[1]}
```




```
% \usepackage{lipsum}
\newtcboxfit{\mybox}[2][]{colback=red!5!white,
  colframe=red!75!black,
  width=#2,height=#2/3*2,#1}

\mybox[colback=yellow]{5cm}%
  {\lipsum[2]}
```



`\renewtcboxfit[⟨init options⟩]{\⟨name⟩}[⟨number⟩][⟨default⟩]{⟨options⟩}`

Operates like `\newtcboxfit`, but based on `\renewcommand` instead of `\newcommand`. An existing macro is redefined.

`\tcbfontsize{⟨factor⟩}`

Selects a font size inside a tcolorbox which is scaled with the given ⟨*factor*⟩ relative to `\tcbfitdim`.

```
\tcbset{colback=red!5!white,colframe=red!75!black}
\begin{tcolorbox}[fit basedim=10pt]
  {\tcbfontsize{0.25} Very tiny,}\\
  {\tcbfontsize{0.5} Small,}\\
  {\tcbfontsize{1} Normal,}\\
  {\tcbfontsize{2} Large,}\\
  {\tcbfontsize{4} Huge.}
\end{tcolorbox}
```

12.2 Option Keys of the Library

The font size for the content of a box with fixed width and fixed height can be adjusted automatically. This is called the *fitbox capture mode*. Note that the fit control algorithm constructs a series of versions for the box and selects the 'best'. Therefore, the compilation time is quite longer than for a normal box. The algorithm will fail, if a different selected font size does not change the overall size of the box content. The `\tcboxfit` → P. 273 macro uses this algorithm by default.

The fit control keys are only applicable to unbreakable boxes without a lower part. The box content should not change counters.

`/tcb/fit` (style, initially unset)

Sets the `/tcb/capture` → P. 76 mode to `fitbox`, i. e. enables the font size adjustment algorithm. Thereby, a `tcolorbox` → P. 9 acts like `\tcboxfit` → P. 273 where the given ⟨*box content*⟩ is fitted to the width and height of the box. Therefore, the box has to have a fixed height. If no fixed height is given, a square box is constructed. The font dimension `\tcbfitdim` can also be used to adjust the margins of the box since a box with a tiny font may not need large margins. The number of constructed boxes is saved to the macro `\tcbfitsteps` for analysis.

```
% \usepackage{lipsum}
% \tcbuselibrary{skins}
\newtcolorbox{fitting}[2][]{fit,height=#2,boxsep=1pt,valign=center,opacityupper=0.5,
  top=0.4\tcbfitdim,bottom=0.4\tcbfitdim,left=0.75\tcbfitdim,right=0.75\tcbfitdim,
  enhanced,watermark text={\tcbfitsteps},colframe=blue!75!black,colback=white,#1}

\begin{fitting}{4cm}
\lipsum[1]
\end{fitting}

\begin{fitting}{2cm}
\lipsum[2]
\end{fitting}

\begin{fitting}{1cm}
\lipsum[3]
\end{fitting}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

/tcb/fit to=⟨*width*⟩ and ⟨*height*⟩ (style, initially unset)

Shortcut for using /tcb/fit → P. 275 and setting the ⟨*width*⟩ and ⟨*height*⟩ values separately.

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

\begin{tcolorbox}[fit to=3cm and 2cm]
  This box content is fitted to the given
  dimensions.
\end{tcolorbox}
```

This box content is fitted to the given dimensions.

/tcb/fit to height=⟨*height*⟩ (style, initially unset)

Shortcut for using /tcb/fit → P. 275 and setting the ⟨*height*⟩ value separately.

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

\begin{tcolorbox}[fit to height=2cm]
  This box content is fitted to the given
  height.
\end{tcolorbox}
```

This box content is fitted to the given height.

/tcb/fit basedim=⟨*length*⟩ (no default, initially 10pt)

Sets the starting font dimension for the font size adjustment algorithm to ⟨*length*⟩. The algorithm never enlarges this dimension.

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

\begin{tcolorbox}[fit to=4cm and 2cm]
  Too few words for the box.
\end{tcolorbox}

\begin{tcolorbox}[fit to=4cm and 2cm,
  fit basedim=50pt]
  Enough words for the box.
\end{tcolorbox}
```

Too few words for the box.

Enough words for the box.

/tcb/fit skip=⟨*real value*⟩ (no default, initially 1.2)

Sets the skip value of the selected font to ⟨*real value*⟩ times \tcbfitdim.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\begin{tcolorbox}[fit to=5cm and 4cm,
  fit skip=1.0  ]
  \lipsum[1]
\end{tcolorbox}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

/tcb/fit fontsize macros (style, initially unset)

Redefines the standard LaTeX font size macros `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, and `\Huge`, to set font sizes relative to the current `\tcbfitdim`. Note that the display skip values for mathematical formulas are respected by the redefined macros.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\begin{tcolorbox}[fit to height=4cm]
  {\Large\bfseries This text is
              not adapted:\par}
  \lipsum[2]
\end{tcolorbox}

\begin{tcolorbox}[fit to height=4cm,
  fit fontsize macros ]
  {\Large\bfseries This text is adapted:\par}
  \lipsum[2]
\end{tcolorbox}
```

This text is not adapted:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This text is adapted:

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\tcbset{colback=red!5!white,
  colframe=red!75!black,left=1mm,
  right=1mm,boxsep=0mm}

\let\realHuge=\Huge

\begin{tcolorbox}[fit basedim=7pt,
  fontupper=\normalsize,
  fit fontsize macros]
The relative relative font size macros
are also usable without the
\textit{fit} algorithm.\par
{\Huge Adapted Huge} ---
{\realHuge Original Huge}
\end{tcolorbox}
```

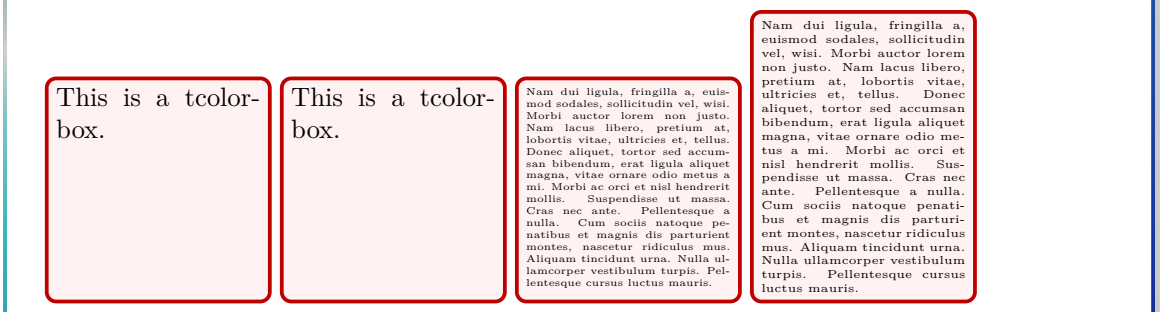
The relative relative font size macros are also usable without the *fit* algorithm.

Adapted Huge – Original Huge

`/tcb/fit height plus=`⟨*dimension*⟩ (no default, initially `0pt`)

The box is allowed to enlarge the fixed height up to the given ⟨*dimension*⟩, before a font size fit is applied. An optional `/tcb/fit width plus` is tried after the height adaption.

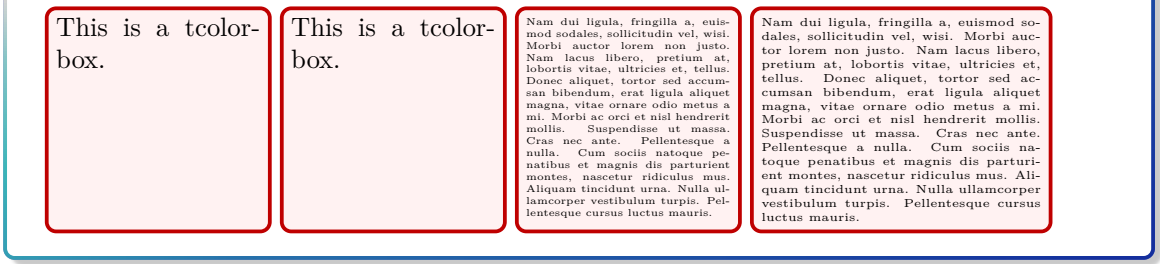
```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,height=3cm,nobeforeafter}

\begin{tcolorbox}[fit]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit,fit height plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit]
\lipsum[2]
\end{tcolorbox}
\begin{tcolorbox}[fit,fit height plus=1cm]
\lipsum[2]
\end{tcolorbox}
```



`/tcb/fit width plus=`⟨*dimension*⟩ (no default, initially `0pt`)

The box is allowed to enlarge the fixed width up to the given ⟨*dimension*⟩, before a font size fit is applied. An optional `/tcb/fit height plus` is tried before the width adaption.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,width=3cm,height=3cm,nobeforeafter}

\begin{tcolorbox}[fit]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit,fit width plus=1cm]
This is a tcolorbox.
\end{tcolorbox}
\begin{tcolorbox}[fit]
\lipsum[2]
\end{tcolorbox}
\begin{tcolorbox}[fit,fit width plus=1cm]
\lipsum[2]
\end{tcolorbox}
```



Typically, the optional title of a `tcolorbox` is not part of the fit operation. If a `/tcb/fit width plus` is applied, the title is also adapted to the new width. If counters are increased inside the title text, they may be increased more than one time. To avoid this, you are encouraged to use `/tcb/phantom` → P. 79 or `/tcb/step and label` → P. 79 to set counters or use automatic numbering, see Subsection 4.1 from page 82.

`/tcb/fit width from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box width to ⟨*min*⟩ and allows the width to grow up to ⟨*max*⟩.

```
% \usepackage{lipsum}
\tcbset{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,bottom=1mm,
  right=1mm,boxsep=0mm,height=4cm}

\begin{tcolorbox}[fit,width=\linewidth/2]
\lipsum[2]
\end{tcolorbox}\par
\begin{tcolorbox}[fit width from=\linewidth/2 to \linewidth]
\lipsum[2]
\end{tcolorbox}\par
```

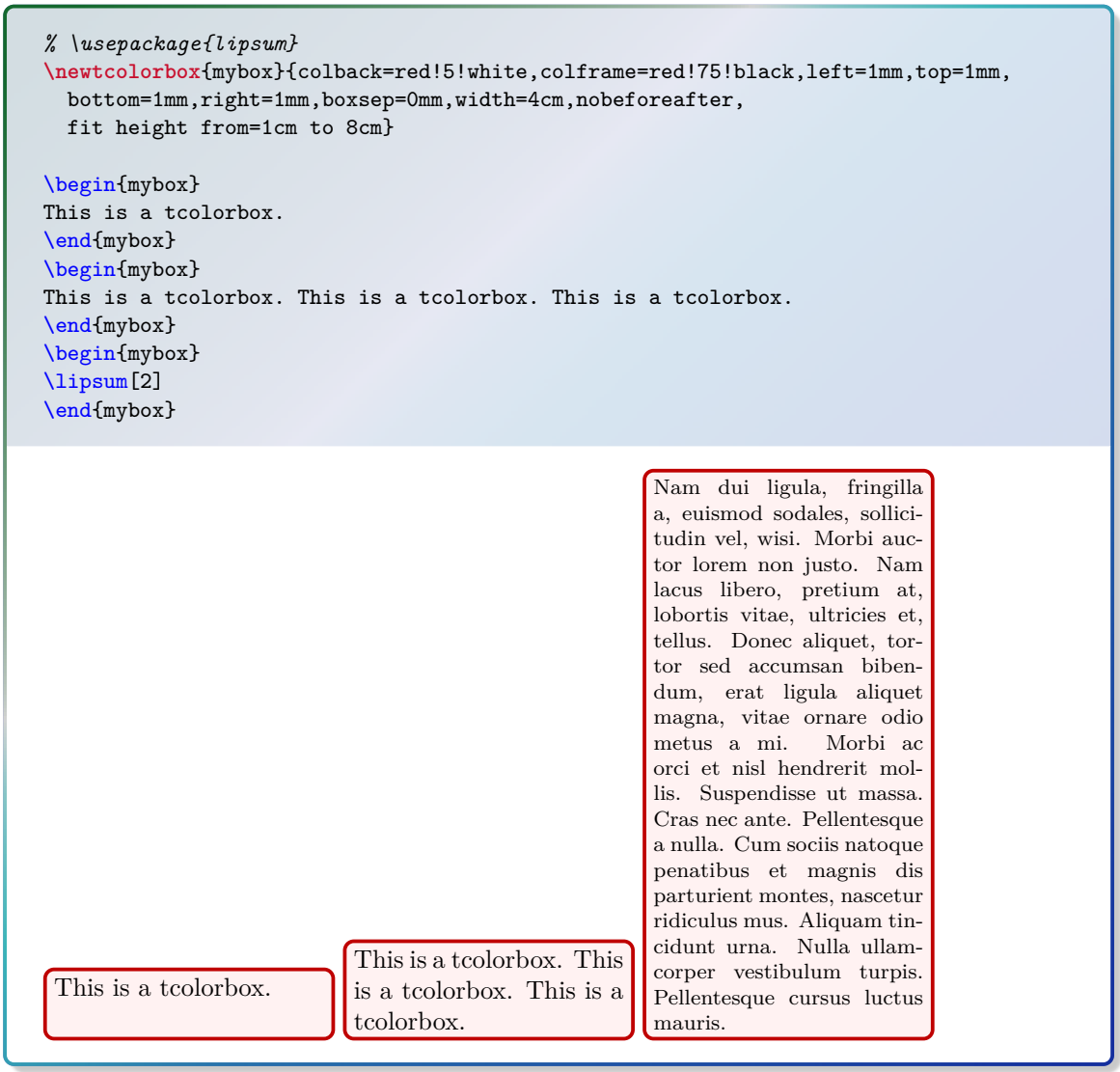
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/fit height from=`⟨*min*⟩ `to` ⟨*max*⟩ (style, no default)

Sets the box height to ⟨*min*⟩ and allows the height to grow up to ⟨*max*⟩.

```
% \usepackage{lipsum}
\newtcolorbox{mybox}{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=4cm,nobeforeafter,
  fit height from=1cm to 8cm}

\begin{mybox}
This is a tcolorbox.
\end{mybox}
\begin{mybox}
This is a tcolorbox. This is a tcolorbox. This is a tcolorbox.
\end{mybox}
\begin{mybox}
\lipsum[2]
\end{mybox}
```



`/tcb/fit algorithm=`⟨*name*⟩ (no default, initially `fontsize`)

Sets the algorithm for the fitting process *after* optionally width and height are adapted. Feasible values for ⟨*name*⟩ are:

- `fontsize` (initial): The algorithm is a bisection method that adapts the font size until certain stop conditions are fulfilled. This is the most time-consuming method but it is robust and gives pleasant results.

  The used font has to be freely scalable for this method! Other content than text is not scaled down. The aspect ratio is fully garanteed.

- `areasize`: The algorithm calculates the area size for the text without scaling the font. The text box is shaped for the needed aspect ratio in one or two steps. Finally, it is scaled down with a standard `\resizebox` macro.

  The used font has not to be scalable. Every box content is scaled down. The aspect ratio cannot be fully garanteed.

- `hybrid`: First, this algorithm estimates the needed font size in one or two steps. Then an `areasize` fitting as above is a applied.

  The used font has to be freely scalable for this method! Other content than text may be slightly rescaled. The aspect ratio cannot be fully garanteed.

- `squeeze`: The text box is brutally scaled down to fit.

  The aspect ratio is very likely to be horrible. You should not use this method for final documents.

```
% \usepackage{lipsum}
\newtcboxfit{mybox}[1]{colback=red!5!white,colframe=red!75!black,left=1mm,top=1mm,
  bottom=1mm,right=1mm,boxsep=0mm,width=3.5cm,height=7cm,nobeforeafter,
  before upper=\textcolor{blue}{\rule{5mm}{5mm}}\ ,
  enhanced,watermark text={\tcbfitsteps},
  fonttitle=\bfseries,adjusted title=#1,fit algorithm=#1}

\mybox{fontsize}{\lipsum[2]}\hfill
\mybox{hybrid}{\lipsum[2]}\hfill
\mybox{areasize}{\lipsum[2]}\hfill
\mybox{squeeze}{\lipsum[2]}

Quality \dotfill versus \dotfill Speed
```

fontsize

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

hybrid

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

areasize

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

squeeze

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Quality . . . . . versus . . . . . Speed

The following options set control parameters for the fit algorithm. Mainly, they apply to the `fontsize` variant, see `/tcb/fit algorithm` → P. 281. The options should be seen as experimental and are likely to change in future versions, if necessary.

`/tcb/fit maxstep`=⟨*number*⟩ (no default, initially `20`)
Sets the maximal step size for the font size adjustment algorithm. In normal situations, the algorithm stops before reaching the intial value of 20 steps. If the box content does not shrink, this value prevents an endless loop.

`/tcb/fit maxfontdiff`=⟨*dimension*⟩ (no default, initially `0.1pt`)
The algorithm stops, if the font size is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxfontdiffgap`=⟨*dimension*⟩ (no default, initially `1pt`)
The algorithm stops, if the number of lines is determined and the font size is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxwidthdiff`=⟨*dimension*⟩ (no default, initially `1pt`)
The algorithm stops, if the (optionally) flexible box width is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit maxwidthdiffgap`=⟨*dimension*⟩ (no default, initially `10pt`)
The algorithm stops, if the number of lines is determined and the (optionally) flexible box width is determined within a deviation of ⟨*dimension*⟩.

`/tcb/fit warning`=⟨*value*⟩ (no default, initially `off`)
Typically, the fit control algorithm constructs several auxiliary boxes to determine the optimal one. If not switched off, the construction of the auxiliary boxes may produce many `hbox` warnings. This option key changes the `\hbadness` value.

- `off`: Most of `'Underfull \hbox'` and `'Overfull \hbox'` warnings are switched off (including the ones for the finally used box).
- `on`: All warnings for all auxiliary boxes are displayed.
- `final`: Only warnings for the finally used box are displayed. Note that an additional box has to be contructed for theses messages.

13 Library LIB hooks

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{hooks}
```

For the skin related options, the library LIB skins has to be loaded separately.

13.1 Concept of Hooks

A hook is a placeholder in some LaTeX code where additional code can be added. For example, the LaTeX macro `\AtBeginDocument` adds code to a hook which is placed at the beginning of every document.

Several option keys of `tcolorbox` allow providing some code which is added to specific places of a colored box. For example, `/tcb/before upper` → P. 46 places code before the content of the upper part. A following usage of this key overwrites any prior settings.

The library LIB hooks extends `/tcb/before upper` → P. 46 and several more existing keys to 'hookable' versions, e. g. `/tcb/before upper app` → P. 284 and `/tcb/before upper pre` → P. 284. The 'hookable' keys don't overwrite prior settings but either *app*end or *pre*pend the newly given code to the existing code.

The general naming convention (with some small exceptions) is:

- ⟨*option key*⟩ `app`: works like ⟨*option key*⟩ but *app*ends its code to the existing code.
- ⟨*option key*⟩ `pre`: works like ⟨*option key*⟩ but *pre*pends its code to the existing code.

If the original ⟨*option key*⟩ is used (again), all code will be overwritten. Therefore, the order of the option key usage is crucial.

```
% \usepackage{array,tabularx}
\newcolumntype{Y}{>{\raggedleft\arraybackslash}X}% see tabularx
\tcbset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
  colback=yellow!10!white,colframe=red!50!black,colbacktitle=Salmon!30!white,
  coltitle=black,center title,
  tabularx={X||Y|Y|Y|Y||Y},% this sets 'before upper' and 'after upper'
  before upper app={Group & One & Two & Three & Four & Sum\\\hline\hline}  }

\begin{tcolorbox}[title=My table]
Red   & 1000.00 & 2000.00 &  3000.00 &  4000.00 & 10000.00\\\hline
Green & 2000.00 & 3000.00 &  4000.00 &  5000.00 & 14000.00\\\hline
Blue  & 3000.00 & 4000.00 &  5000.00 &  6000.00 & 18000.00\\\hline\hline
Sum   & 6000.00 & 9000.00 & 12000.00 & 15000.00 & 42000.00
\end{tcolorbox}
```

My table

Group	One	Two	Three	Four	Sum
Red	1000.00	2000.00	3000.00	4000.00	10000.00
Green	2000.00	3000.00	4000.00	5000.00	14000.00
Blue	3000.00	4000.00	5000.00	6000.00	18000.00
Sum	6000.00	9000.00	12000.00	15000.00	42000.00

13.2 Box Content Additions

The following option keys extend the options given in Subsection 3.9 from page 46.

/tcb/before title app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/before title→ P. 46 *after* the color and font settings and *before* the content of the title.

/tcb/before title pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/before title→ P. 46 *after* the color and font settings and *before* the content of the title.

/tcb/after title app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/after title→ P. 46 *after* the content of the title.

/tcb/after title pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/after title→ P. 46 *after* the content of the title.

/tcb/before upper app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/before upper→ P. 46 *after* the color and font settings and *before* the content of the upper part.

/tcb/before upper pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/before upper→ P. 46 *after* the color and font settings and *before* the content of the upper part.

/tcb/after upper app=⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to /tcb/after upper→ P. 47 *after* the content of the upper part.

/tcb/after upper pre=⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to /tcb/after upper→ P. 47 *after* the content of the upper part.

```
% \tcbuselibrary{theorems}
\tcbset{ams align,%  this sets 'before upper' and 'after upper'
  colback=yellow!10!white,colframe=red!50!black,
  before upper app={\frac{2}{\sqrt{2}}&=\sqrt{2}.\\},
  after upper pre={\\\sin\left(\frac{\pi}{2}\right)&=1.},
}

\begin{tcolorbox}
  \sum\limits_{n=1}^{\infty} \frac{1}{n} &= \infty.\\
  \int x^2 ~\text{d}x &= \frac13 x^3 + c.
\end{tcolorbox}
```

$$\frac{2}{\sqrt{2}} = \sqrt{2}. \tag{22}$$

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{23}$$

$$\int x^2 \, \mathrm{d}x = \frac{1}{3}x^3 + c. \tag{24}$$

$$\sin\left(\frac{\pi}{2}\right) = 1. \tag{25}$$

`/tcb/before lower app=`⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to `/tcb/before lower` → P. 47 *after* the color and font settings and *before* the content of the lower part.

`/tcb/before lower pre=`⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to `/tcb/before lower` → P. 47 *after* the color and font settings and *before* the content of the lower part.

`/tcb/after lower app=`⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to `/tcb/after lower` → P. 47 *after* the content of the lower part.

`/tcb/after lower pre=`⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to `/tcb/after lower` → P. 47 *after* the content of the lower part.

13.3 Embedding into the Surroundings

The following option keys extend the options given in Subsection 3.13 from page 65.

The 'hookable' versions are usable inside the document. In the preamble, they can only be used after explicit setting of `/tcb/before` → P. 65 and `/tcb/after` → P. 65 or by e. g. `/tcb/parskip` → P. 65.

`/tcb/before app=`⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to `/tcb/before` → P. 65 before the colored box.

`/tcb/before pre=`⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to `/tcb/before` → P. 65 before the colored box.

`/tcb/after app=`⟨*code*⟩ (no default)
Appends the given ⟨*code*⟩ to `/tcb/after` → P. 65 after the colored box.

`/tcb/after pre=`⟨*code*⟩ (no default)
Prepends the given ⟨*code*⟩ to `/tcb/after` → P. 65 after the colored box.

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

\begin{tcolorbox}[title=My title,before app={The box follows:\\[4pt]},
  after app={This is the end.}]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

The box follows:

My title

This is a **tcolorbox**.

This is the end.

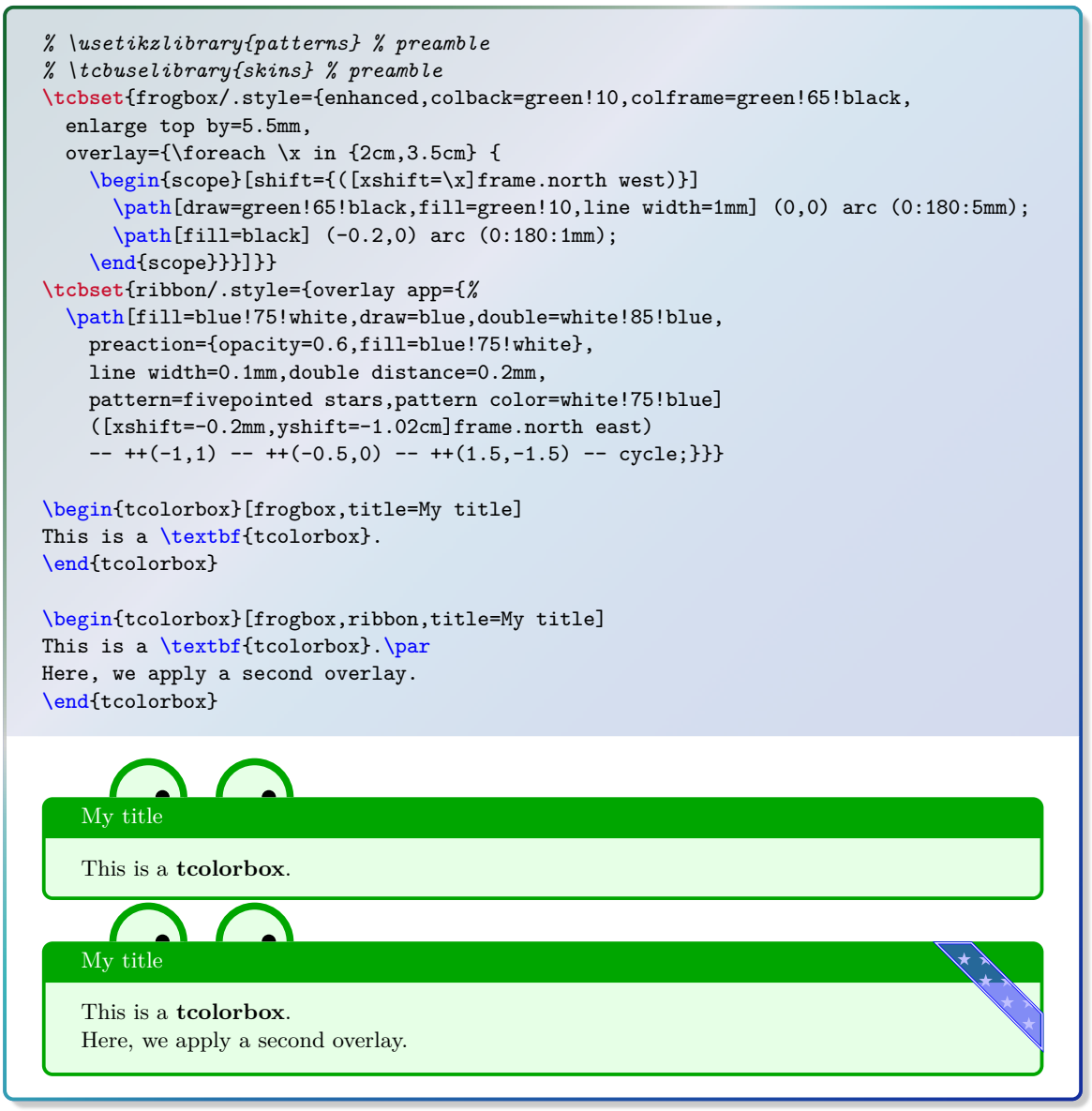
13.4 Overlays

The following option keys extend the options given in Subsection 3.10 from page 55.

`/tcb/overlay app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay`→ P. 55.

```
% \usetikzlibrary{patterns} % preamble
% \tcbuselibrary{skins} % preamble
\tcbset{frogbox/.style={enhanced,colback=green!10,colframe=green!65!black,
  enlarge top by=5.5mm,
  overlay={\foreach \x in {2cm,3.5cm} {
    \begin{scope}[shift={([xshift=\x]frame.north west)}]
      \path[draw=green!65!black,fill=green!10,line width=1mm] (0,0) arc (0:180:5mm);
      \path[fill=black] (-0.2,0) arc (0:180:1mm);
    \end{scope}}}]}}
\tcbset{ribbon/.style={overlay app={%
  \path[fill=blue!75!white,draw=blue,double=white!85!blue,
    preaction={opacity=0.6,fill=blue!75!white},
    line width=0.1mm,double distance=0.2mm,
    pattern=fivepointed stars,pattern color=white!75!blue]
    ([xshift=-0.2mm,yshift=-1.02cm]frame.north east)
    -- ++(-1,1) -- ++(-0.5,0) -- ++(1.5,-1.5) -- cycle;}}}

\begin{tcolorbox}[frogbox,title=My title]
This is a \textbf{tcolorbox}.
\end{tcolorbox}

\begin{tcolorbox}[frogbox,ribbon,title=My title]
This is a \textbf{tcolorbox}.\par
Here, we apply a second overlay.
\end{tcolorbox}
```



`/tcb/overlay pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay`→ P. 55.

`/tcb/overlay unbroken app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken`→ P. 56.

`/tcb/overlay unbroken pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken`→ P. 56.

`/tcb/overlay first app=`⟨*graphical code*⟩ (no default)

Appends the given ⟨*graphical code*⟩ to `/tcb/overlay first`→ P. 56.

`/tcb/overlay first pre=`⟨*graphical code*⟩ (no default)

Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay first`→ P. 56.

`/tcb/overlay middle app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay middle` → P. 56.

`/tcb/overlay middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay middle` → P. 56.

`/tcb/overlay last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay last` → P. 56.

`/tcb/overlay last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay last` → P. 56.

`/tcb/overlay broken app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay broken` → P. 56.

`/tcb/overlay broken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay broken` → P. 56.

`/tcb/overlay unbroken and first app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and first` → P. 56.

`/tcb/overlay unbroken and first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and first` → P. 56.

`/tcb/overlay middle and last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay middle and last` → P. 56.

`/tcb/overlay middle and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay middle and last` → P. 56.

`/tcb/overlay unbroken and last app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and last` → P. 56.

`/tcb/overlay unbroken and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay unbroken and last` → P. 56.

N 2014-09-19 `/tcb/overlay first and middle app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/overlay first and middle` → P. 56.

N 2014-09-19 `/tcb/overlay first and middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/overlay first and middle` → P. 56.

13.5 Watermarks

The following option keys extend the options given in Subsection 7.3 from page 114.

Watermarks are special overlays. The LIB hooks library allows the combination of several watermarks and overlays.

`/tcb/watermark text app=`⟨*text*⟩ (no default)

Appends a `/tcb/watermark text` →P. 114 to the colored box.

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

\begin{tcolorbox}[enhanced,title=My title,watermark graphics=Basilica_5.png,
  watermark opacity=0.25,
  watermark text app=Basilica,watermark color=Navy
  ]
\lipsum[1-2]
\tcblower
This example uses a public domain picture from\\
\url{http://commons.wikimedia.org/wiki/File:Basilica_5.png}
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

This example uses a public domain picture from
http://commons.wikimedia.org/wiki/File:Basilica_5.png

`/tcb/watermark text pre=`⟨*text*⟩ (no default)

Prepends a `/tcb/watermark text` →P. 114 to the colored box.

`/tcb/watermark text app on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default)

Appends a `/tcb/watermark text on` →P. 114 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark text pre on=`⟨*part*⟩ `is` ⟨*text*⟩ (no default)

Prepends a `/tcb/watermark text on` →P. 114 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark graphics app=`⟨*file name*⟩ (no default)
Appends a `/tcb/watermark graphics` →P. 115 referenced by ⟨*file name*⟩ to the colored box.

`/tcb/watermark graphics pre=`⟨*file name*⟩ (no default)
Prepends a `/tcb/watermark graphics` →P. 115 referenced by ⟨*file name*⟩ to the colored box.

`/tcb/watermark graphics app on=`⟨*part*⟩ `is` ⟨*file name*⟩ (no default)
Appends a `/tcb/watermark graphics on` →P. 115 the named ⟨*part*⟩ of a break sequence. The picture is referenced by ⟨*file name*⟩.

`/tcb/watermark graphics pre on=`⟨*part*⟩ `is` ⟨*file name*⟩ (no default)
Prepends a `/tcb/watermark graphics on` →P. 115 the named ⟨*part*⟩ of a break sequence. The picture is referenced by ⟨*file name*⟩.

`/tcb/watermark tikz app=`⟨*graphical code*⟩ (no default)
Appends a `/tcb/watermark tikz` →P. 116 with the given `tikz` ⟨*graphical code*⟩ to the colored box.

`/tcb/watermark tikz pre=`⟨*graphical code*⟩ (no default)
Prepends a `/tcb/watermark tikz` →P. 116 with the given `tikz` ⟨*graphical code*⟩ to the colored box.

```
% \usepackage{tikz}
\tcbset{colback=red!5!white,colframe=red!75!black,fonttitle=\bfseries,
  watermark color=Navy,watermark opacity=0.25,
  smiley/.style={watermark tikz pre={%
    \path[fill=yellow,draw=yellow!75!red] (0,0) circle (1cm);
    \fill[red] (45:5mm) circle (1mm);
    \fill[red] (135:5mm) circle (1mm);
    \draw[line width=1mm,red] (215:5mm) arc (215:325:5mm);}}

\begin{tcolorbox}[enhanced,title=My title, watermark text=Watermark,
  smiley]
\lipsum[1-2]
\end{tcolorbox}
```

My title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`/tcb/watermark tikz app on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default)
Appends a `/tcb/watermark tikz on` →P. 116 the named ⟨*part*⟩ of a break sequence.

`/tcb/watermark tikz pre on=`⟨*part*⟩ `is` ⟨*graphical code*⟩ (no default)
Prepends a `/tcb/watermark tikz on` →P. 116 the named ⟨*part*⟩ of a break sequence.

13.6 Underlays

The following option keys extend the options given in Section 7.8 on page 141. There are no `app` type keys since underlays are stackable by default.

`/tcb/underlay pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay` → P. 141.

`/tcb/underlay unbroken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken` → P. 142.

`/tcb/underlay first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay first` → P. 142.

`/tcb/underlay middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay middle` → P. 142.

`/tcb/underlay last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay last` → P. 142.

`/tcb/underlay boxed title pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay boxed title` → P. 142.

`/tcb/underlay broken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay broken` → P. 142.

`/tcb/underlay unbroken and first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken and first` → P. 142.

`/tcb/underlay middle and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay middle and last` → P. 142.

`/tcb/underlay unbroken and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay unbroken and last` → P. 142.

N 2014-09-19 `/tcb/underlay first and middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/underlay first and middle` → P. 142.

13.7 Finishes

The following option keys extend the options given in Section 7.9 on page 143. There are no `app` type keys since finishes are stackable by default.

`/tcb/finish pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish`→P. 143.

`/tcb/finish unbroken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken`→P. 144.

`/tcb/finish first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish first`→P. 144.

`/tcb/finish middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish middle`→P. 144.

`/tcb/finish last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish last`→P. 144.

`/tcb/finish broken pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish broken`→P. 144.

`/tcb/finish unbroken and first pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken and first`→P. 144.

`/tcb/finish middle and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish middle and last`→P. 144.

`/tcb/finish unbroken and last pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish unbroken and last`→P. 144.

N 2014-09-19 `/tcb/finish first and middle pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/finish first and middle`→P. 144.

13.8 Skin Code

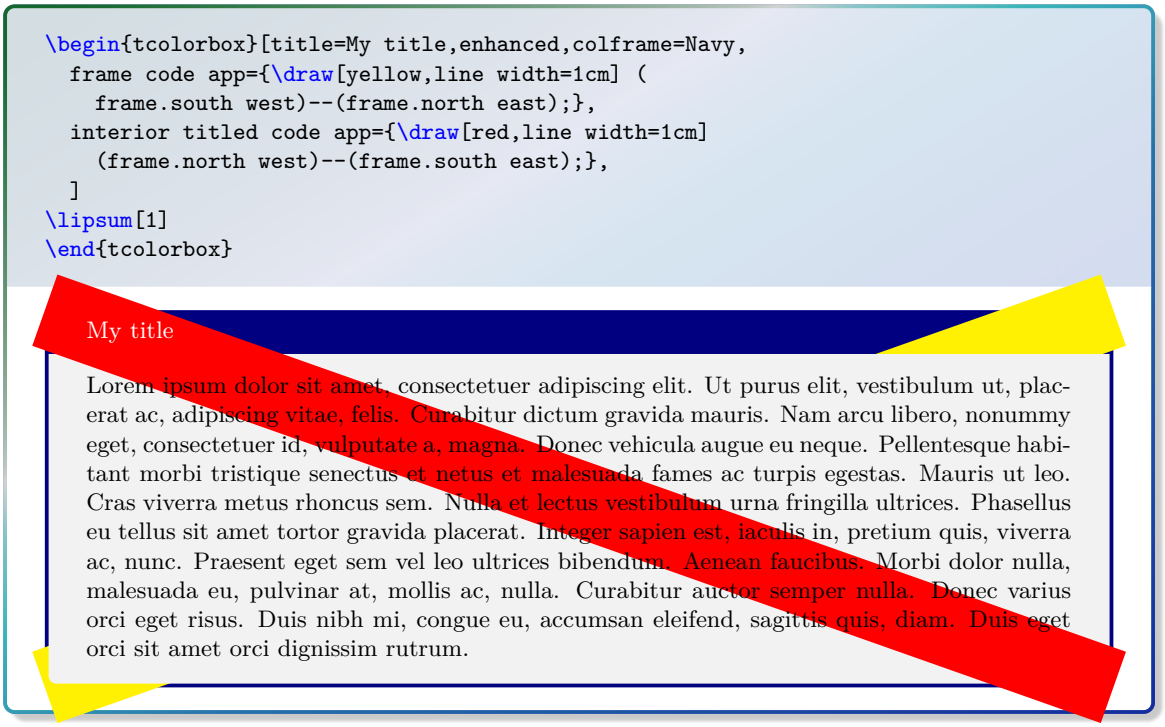
The following option keys extend the options given in Subsection 6.2 from page 93.

`/tcb/frame code app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/frame code`→P. 93.

`/tcb/frame code pre`=⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/frame code`→P. 93.

`/tcb/interior titled code app`=⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/interior titled code`→P. 93.

```
\begin{tcolorbox}[title=My title,enhanced,colframe=Navy,
  frame code app={\draw[yellow,line width=1cm] (
    frame.south west)--(frame.north east);},
  interior titled code app={\draw[red,line width=1cm]
    (frame.north west)--(frame.south east);},
  ]
\lipsum[1]
\end{tcolorbox}
```



`/tcb/interior titled code pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/interior titled code` → P. 93.

`/tcb/interior code app=`⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/interior code` → P. 94.

`/tcb/interior code pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/interior code` → P. 94.

`/tcb/segmentation code app=`⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/segmentation code` → P. 94.

`/tcb/segmentation code pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/segmentation code` → P. 94.

`/tcb/title code app=`⟨*graphical code*⟩ (no default)
Appends the given ⟨*graphical code*⟩ to `/tcb/title code` → P. 95.

`/tcb/title code pre=`⟨*graphical code*⟩ (no default)
Prepends the given ⟨*graphical code*⟩ to `/tcb/title code` → P. 95.

14 Library LIB xparse

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{xparse}
```

This also loads the package `xparse` [11].

The purpose of this library is to give comfortable access to the powerful document command production with `xparse` for `tcolorbox`. See the `xparse` package documentation [11] for details about the argument ⟨*specification*⟩ used in this section.

14.1 Option Keys

`/tcb/verbatim` (style, no value)

Sets options for a *verbatim* style `\tcbox` → P. 11. Since the indented boxes may contain only very few words, the dimensions are made smaller and `/tcb/nobeforeafter` → P. 65 and `/tcb/tcbox raise base` → P. 78 are set.

```
\DeclareTotalTCBox{\myverb}{ v }{verbatim,
  colframe=red!75!black,colupper=blue}{#1}

\myverb{\textbf} is a \myverb{\LaTeX} command.
```

\textbf is a \LaTeX command.

`/tcb/IfNoValueTF={`⟨*argument*⟩`}{`⟨*true options*⟩`}{`⟨*false options*⟩`}` (no default)

Wraps the `\IfNoValueTF` command of `xparse` for option setting. If the ⟨*argument*⟩ has no value, the ⟨*true options*⟩ are set. Otherwise, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ o }{colframe=red!75!black,
  IfNoValueTF={#1}{colback=red!5!white}{enhanced,interior style image=#1}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[goldshade.png]
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox.

`/tcb/IfValueTF={`⟨*argument*⟩`}{`⟨*true options*⟩`}{`⟨*false options*⟩`}` (no default)

Wraps the `\IfValueTF` command of `xparse` for option setting. If the ⟨*argument*⟩ has a value, the ⟨*true options*⟩ are set. Otherwise, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ o }{colframe=red!75!black,colback=red!5!white,
  IfValueTF={#1}{title={\flqq #1\frqq},fonttitle=\bfseries}{}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[My title]
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

«My title»

This is a tcolorbox.

`/tcb/IfBooleanTF={`⟨*argument*⟩`}{`⟨*true options*⟩`}{`⟨*false options*⟩`}` (no default)

Wraps the `\IfBooleanTF` command of `xparse` for option setting. If the ⟨*argument*⟩ is `\BooleanTue`, the ⟨*true options*⟩ are set. If the ⟨*argument*⟩ is `\BooleanFalse`, the ⟨*false options*⟩ are set.

```
\DeclareTColorBox{mybox}{ s }{colframe=red!75!black,
  IfBooleanTF={#1}{colback=yellow!50!red}{colback=red!5!white}}

\begin{mybox}
This is a tcolorbox.
\end{mybox}

\begin{mybox}*
This is a tcolorbox.
\end{mybox}
```

This is a tcolorbox.

This is a tcolorbox.

14.2 Producing tcolorbox Environments and Commands

`\DeclareTColorBox[⟨init options⟩]{⟨name⟩}{⟨specification⟩}{⟨options⟩}`

Creates a new environment ⟨*name*⟩ based on tcolorbox → P. 9.

Basically, `\DeclareTColorBox` operates like `\DeclareDocumentEnvironment`. This means, the new environment ⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying tcolorbox → P. 9.

Note that /tcb/savedelimiter → P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new environment is always created, irrespective of an already existing environment with the same name.

```
% counter from previous example
\DeclareTColorBox[use counter from=pabox]{mybox}{ O{red} m d"" O{} }
  {enhanced,colframe=#1!75!black,colback=#1!5!white,
   fonttitle=\bfseries,title={\thetcbcounter~#2},
   IfValueTF={#3}{watermark text={#3}}{},#4}

\begin{mybox}{My title}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[blue]{My title}
This is a tcolorbox.
\end{mybox}

\begin{mybox}[green]{My title}"My Watermark"
This is a tcolorbox.
\end{mybox}

\begin{mybox}[yellow]{My title}[colbacktitle=yellow!50!white,coltitle=black]
This is a tcolorbox.
\end{mybox}

\begin{mybox}[purple]{My title}"All together"[coltitle=yellow]
This is a tcolorbox.
\end{mybox}
```

14.1 My title

This is a tcolorbox.

14.2 My title

This is a tcolorbox.

14.3 My title

This is a tcolorbox. My Watermark

14.4 My title

This is a tcolorbox.

14.5 My title

This is a tcolorbox. All together

`\NewTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 295, but based on `\NewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An error is issued if ⟨*name*⟩ has already been defined.

`\RenewTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 295, but based on `\RenewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An existing environment is redefined.

`\ProvideTColorBox[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTColorBox` → P. 295, but based on `\ProvideDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. The environment ⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Creates a new command \⟨*name*⟩ based on `tcolorbox` → P. 9. In contrast to `\DeclareTColorBox` → P. 295, also the ⟨*content*⟩ of the `tcolorbox` is specified.

Basically, `\DeclareTotalTColorBox` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `tcolorbox` → P. 9 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` → P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new command is always created, irrespective of an already existing command with the same name.

```
\DeclareTotalTColorBox{\diabox}{ O{} v  m }
  { bicolor,nobeforeafter,equal height group=diabox,width=5.7cm,
    fonttitle=\bfseries\ttfamily,adjusted title={#2},center title,
    colframe=blue!20!black,leftupper=0mm,rightupper=0mm,colback=black!75!white,#1}
  { \tikz\path[fill zoom image={#2}] (0,0) rectangle (\linewidth,4cm);%
    \tcblower#3}

\diabox{blueshade.png}{Created with |GIMP|.\\\url{http://www.gimp.org}}
\diabox{goldshade.png}{Created with |GIMP|.\\\url{http://www.gimp.org}}
```

`\NewTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if \⟨*name*⟩ has already been defined.

`\RenewTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTotalTColorBox[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}{`⟨*content*⟩`}`

Operates like `\DeclareTotalTColorBox`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command \⟨*name*⟩ is only created if it is not already defined.

14.3 Producing tcbox Commands

\DeclareTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Creates a new command \⟨*name*⟩ based on \tcbox → P. 11. Basically, \DeclareTCBox operates like \DeclareDocumentCommand. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying \tcbox → P. 11.

Note that /tcb/savedelimiter → P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new command is always created, irrespective of an already existing command with the same name.

```
% counter from previous example
\DeclareTCBox[use counter from=pabox]{\mybox}{ s m s }
{ nobeforeafter,colback=red!5!white,colframe=red!75!black,
  title={#2 (Box \thetcbcounter)},fonttitle=\bfseries,
  IfBooleanTF={#1}{enhanced,drop shadow}{},
  IfBooleanTF={#3}{colbacktitle=red!50!white}{}  }

\mybox{Bird}{This is my first box.}
  \hfill
\mybox*{Tree}{This is my second box.}
  \par\bigskip
\mybox{Bike}*{This is my third box.}
  \hfill
\mybox*{City}*{This is my fourth box.}
```

Bird (Box 14.6)
This is my first box.

Tree (Box 14.7)
This is my second box.

Bike (Box 14.8)
This is my third box.

City (Box 14.9)
This is my fourth box.

\NewTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \NewDocumentCommand instead of \DeclareDocumentCommand. An error is issued if \⟨*name*⟩ has already been defined.

\RenewTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \RenewDocumentCommand instead of \DeclareDocumentCommand. An existing command is redefined.

\ProvideTCBox[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}

Operates like \DeclareTCBox, but based on \ProvideDocumentCommand instead of \DeclareDocumentCommand. The command \⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTCBox`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Creates a new command \⟨*name*⟩ based on `\tcbox` → P. 11. In contrast to `\DeclareTCBox` → P. 298, also the ⟨*content*⟩ of the `tcbox` is specified.

Basically, `\DeclareTotalTCBox` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcbox` → P. 11 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` → P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new command is always created, irrespective of an already existing command with the same name.

```
\DeclareTotalTCBox{\myverb}{ O{red} v O{} }
{ fontupper=\ttfamily,nobeforeafter,tcbox raise base,arc=0pt,outer arc=0pt,
  top=0pt,bottom=0pt,left=0mm,right=0mm,
  leftrule=0pt,rightrule=0pt,toprule=0.3mm,bottomrule=0.3mm,boxsep=0.5mm,
  colback=#1!10!white,colframe=#1!50!black,#3}{#2}

To set a word \textbf{bold} in \myverb{\LaTeX}, one uses
\myverb[green]{\textbf{bold}}. Alternatively, write
\myverb[yellow]{{\bfseries bold}}.
In \myverb[blue]{\LaTeX}[enhanced,fuzzy halo], other font settings are
done in the same way, e.\,g. \myverb{\textit}, \myverb{\itshape}\\
or \myverb[brown]{\texttt}, \myverb[brown]{\ttfamily}.
```

To set a word **bold** in `\LaTeX`, one uses `\textbf{bold}`. Alternatively, write `{\bfseries bold}`.
In `\LaTeX`, other font settings are done in the same way, e. g. `\textit`, `\itshape`
or `\texttt`, `\ttfamily`.

The next example uses `\lstinline` from the `listings` package to typeset the verbatim content.

```
% \usepackage{listings} or \tcbuselibrary{listings}
\DeclareTotalTCBox{\commandbox}{ s v }
{verbatim,colupper=white,colback=black!75!white,colframe=black}
{\IfBooleanTF{#1}{\textcolor{red}{\ttfamily\bfseries > }}{}%
  \lstinline[language=command.com,keywordstyle=\color{blue!35!white}\bfseries]^#2^}

\commandbox*{cd "My Documents"} changes to directory \commandbox{My Documents}.

\commandbox*{dir /A} lists the directory content.

\commandbox*{copy example.txt d:\target} copies \commandbox{example.txt} to
  \commandbox{d:\target}.
```

`> cd "My␣Documents"` changes to directory `My Documents`.
`> dir /A` lists the directory content.
`> copy example.txt d:\target` copies `example.txt` to `d:\target`.

`\NewTotalTCBox`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Operates like `\DeclareTotalTCBox` → P. 299, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if \⟨*name*⟩ has already been defined.

`\RenewTotalTCBox`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Operates like `\DeclareTotalTCBox` → P. 299, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

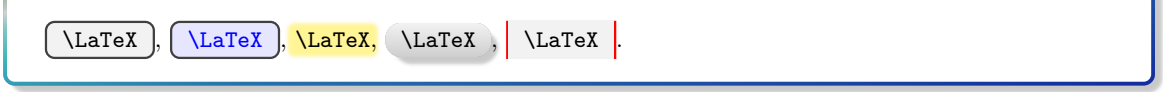
`\ProvideTotalTCBox`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Operates like `\DeclareTotalTCBox` → P. 299, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command \⟨*name*⟩ is only created if it is not already defined.

`\tcboxverb`[⟨*options*⟩]{⟨*verbatim box content*⟩}

Creates a colored box based on `\tcbox` → P. 11 which is fitted to the width of the given ⟨*verbatim box content*⟩. The underlying `\tcbox` → P. 11 is styled with `/tcb/verbatim` → P. 293 plus the given ⟨*options*⟩. The difference to `\tcbox` → P. 11 is that the ⟨*verbatim box content*⟩ is interpreted *verbatim*. Therefore, `\tcboxverb` acts similar to `\verb`.

```
\tcboxverb{\LaTeX}, \tcboxverb[colback=blue!10!white,colupper=blue]{\LaTeX},
\tcboxverb[blank,fuzzy halo]{\LaTeX}, \tcboxverb[beamer]{\LaTeX},
\tcboxverb[enhanced,skin=enhancedmiddle jigsaw,colframe=red]{\LaTeX}.
```



14.4 Producing `tcblisting` Environments

The following commands need the LIB listings library to be included.

`\DeclareTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Creates a new environment ⟨*name*⟩ based on `tcblisting` → P. 200.
Basically, `\DeclareTCBListing` operates like `\DeclareDocumentEnvironment`. This means, the new environment ⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `tcblisting` → P. 200.
Note that `/tcb/savedelimiter` → P. 19 is set to the given ⟨*name*⟩ automatically.
The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.
The new environment is always created, irrespective of an already existing environment with the same name.

```
\DeclareTCBListing{mybox}{ s O{} m }{%
  colback=red!5!white,
  colframe=red!75!black,
  fonttitle=\bfseries,
  IfBooleanTF={#1}
    {listing side text}
    {text side listing},
  title=#3,#2}

\begin{mybox}{Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}*{Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
\bigskip

\begin{mybox}[colback=yellow]
  {Listing Box}
This is my
\LaTeX\ box.
\end{mybox}
```

Listing Box

This is my LaTeX box.

```
This is my
\LaTeX\ box.
```

Listing Box

```
This is my
\LaTeX\ box.
```

This is my LaTeX box.

Listing Box

This is my LaTeX box.

```
This is my
\LaTeX\ box.
```

`\NewTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\NewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An error is issued if ⟨*name*⟩ has already been defined.

`\RenewTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\RenewDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. An existing environment is redefined.

`\ProvideTCBListing[`⟨*init options*⟩`]{`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBListing`, but based on `\ProvideDocumentEnvironment` instead of `\DeclareDocumentEnvironment`. The environment ⟨*name*⟩ is only created if it is not already defined.

Caveats of using an environment ending with an optional argument

```
\DeclareTCBListing{mybox}{ O{} }{listing only,#1}

\begin{mybox}[colframe=red]
\good
\end{mybox}

\begin{mybox}[colframe=red]\good\end{mybox}

\begin{mybox}
\good
\end{mybox}

\begin{mybox} \good\end{mybox}

\begin{mybox}\bad!\end{mybox}

\begin{mybox}
[\good]
\end{mybox}

\begin{mybox} [\good]\end{mybox}

\begin{mybox}[\bad!]\end{mybox}
```

14.5 Producing `tcbinputlisting` Commands

The following commands need the LIB listings library to be included.

`\DeclareTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Creates a new command `\`⟨*name*⟩ based on `\tcbinputlisting` → P. 202. Basically, `\DeclareTCBInputListing` operates like `\DeclareDocumentCommand`. This means, the new command `\`⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcbinputlisting` → P. 202.
The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.
The new command is always created, irrespective of an already existing command with the same name.

```
% counter from previous example
\DeclareTCBInputListing[use counter from=pabox]{\mylisting}{ O{} O{red} m }{%
  listing file={#3},title=Listing~\thetcbcounter,
  colback=#2!5!white,colframe=#2!50!black,colbacktitle=#2!75!black,
  fonttitle=\bfseries,listing only,#1}

\mylisting[before upper=\textit{This is the included file content:}]
  [blue]{\jobname.tcbtemp}
```

Listing 14.10

This is the included file content:

```
% counter from previous example
\DeclareTCBInputListing[use counter from=pabox]{\mylisting}{ O{}
    O{red} m }{%
  listing file={#3},title=Listing~\thetcbcounter,
  colback=#2!5!white,colframe=#2!50!black,colbacktitle=#2!75!black,
  fonttitle=\bfseries,listing only,#1}

\mylisting[before upper=\textit{This is the included file content:}]
  [blue]{\jobname.tcbtemp}
```

`\NewTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if `\`⟨*name*⟩ has already been defined.

`\RenewTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTCBInputListing[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBInputListing`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command `\`⟨*name*⟩ is only created if it is not already defined.

14.6 Producing `tboxfit` Commands

The following commands need the LIB fitting library to be included.

`\DeclareTCBoxFit[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

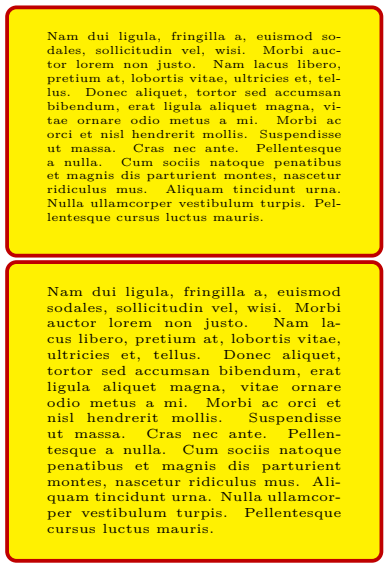
Creates a new command `\`⟨*name*⟩ based on `\tcboxfit` → P. 273. Basically, `\DeclareTCBoxFit` operates like `\DeclareDocumentCommand`. This means, the new command `\`⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcboxfit` → P. 273.

Note that `/tcb/savedelimiter` → P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new command is always created, irrespective of an already existing command with the same name.

```
% \usepackage{lipsum}

\DeclareTCBoxFit{\mybox}{ O{} m o }
 {colback=red!5!white,
  colframe=red!75!black,
  width=#2,height=#2/3*2,
  IfValueTF={#3}{height=#3}{},
  #1}

\mybox[colback=yellow]{5cm}%
  {\lipsum[2]}

\mybox[colback=yellow]{5cm}[4cm]{\lipsum[2]}
```



`\NewTCBoxFit[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBoxFit`, but based on `\NewDocumentCommand` instead of `\DeclareDocumentCommand`. An error is issued if `\`⟨*name*⟩ has already been defined.

`\RenewTCBoxFit[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBoxFit`, but based on `\RenewDocumentCommand` instead of `\DeclareDocumentCommand`. An existing command is redefined.

`\ProvideTCBoxFit[`⟨*init options*⟩`]{\`⟨*name*⟩`}{`⟨*specification*⟩`}{`⟨*options*⟩`}`

Operates like `\DeclareTCBoxFit`, but based on `\ProvideDocumentCommand` instead of `\DeclareDocumentCommand`. The command `\`⟨*name*⟩ is only created if it is not already defined.

`\DeclareTotalTCBoxFit`[⟨*init options*⟩]{\⟨*name*⟩}{⟨*specification*⟩}{⟨*options*⟩}{⟨*content*⟩}

Creates a new command \⟨*name*⟩ based on `\tcboxfit` →P. 273. In contrast to `\DeclareTCBoxFit` →P. 304, also the ⟨*content*⟩ of the `tcboxfit` is specified.

Basically, `\DeclareTotalTCBoxFit` operates like `\DeclareDocumentCommand`. This means, the new command \⟨*name*⟩ is constructed with the given argument ⟨*specification*⟩. The ⟨*options*⟩ are given to the underlying `\tcboxfit` →P. 273 which is filled with the specified ⟨*content*⟩.

Note that `/tcb/savedelimiter` →P. 19 is set to the given ⟨*name*⟩ automatically.

The ⟨*init options*⟩ allow setting up automatic numbering, see Section 4 from page 82.

The new command is always created, irrespective of an already existing command with the same name.

```
% \usepackage{lipsum}

\DeclareTotalTCBoxFit{\multibox}{ O{} m O{10} m }
 {nobeforeafter,colback=red!5!white,colframe=red!75!black,width=#2,height=#2/3*2,
  valign=center,#1}
 { \foreach \n in {1,...,#3} { #4} }

\multibox{5cm}{I shall not repeat.}
\multibox[colframe=blue!75!white]{5cm}[20]{I shall not repeat.}\\
\multibox[colback=yellow,height=5cm]{14cm}[100]{I shall not repeat.}
```

I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat.

I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.
I shall not repeat. I shall not repeat.

I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat. I shall not repeat.

15 Library documentation

This library has the single purpose to support LaTeX package documentations like this one. Actually, the visual nature follows the approach from Till Tantau's `pgf` [20] documentation. Typically, this library is assumed to be used in conjunction with the class `ltxdoc` or alike.

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{documentation}
```

This also loads the library listings, see Section 9 on page 199, the library skins, see
N 2014-09-19 Section 7 on page 101, the library xparse , see Section 14 on page 293, and a bunch of packages, namely `doc`, `pifont`, `marvosym`, `makeidx`, `marginnote`, `refcount`, and `hyperref`.

The package `makeidx` is loaded only, if `\printindex` is *not* already defined. Therefore, one can include an alternative to `makeidx` like `imakeidx` *before* the library `documentation` is used.

For UTF-8 support, load:

```
\tcbuselibrary{listingsutf8,documentation}
```

For `minted` [14] support, load:

```
\tcbuselibrary{documentation,minted}
\tcbset{listing engine=minted}
```

15.1 Macros of the Library

`\begin{docCommand}[`⟨*options*⟩`]{`⟨*name*⟩`}{`⟨*parameters*⟩`}`
⟨*command description*⟩
`\end{docCommand}`

Documents a LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The given ⟨*options*⟩ are set with `\tcbset` →P. 10. This macro takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with `\refCom` →P. 312`{`⟨*name*⟩`}`.

```
\begin{docCommand}{foomakedocSubKey}{\marg{name}\marg{key path}}
  Creates a new environment \meta{name} based on \refEnv{docKey} for the
  documentation of keys with the given \meta{key path}.
\end{docCommand}
```

`\foomakedocSubKey{`⟨*name*⟩`}{`⟨*key path*⟩`}`
Creates a new environment ⟨*name*⟩ based on `docKey` →P. 307 for the documentation of keys with the given ⟨*key path*⟩.

```
\begin{docCommand}[color definition=blue]{foomakedocSubKey*}%
    {\marg{name}\marg{key path}}
  Creates a new environment \meta{name} based on \refEnv{docKey} for the
  documentation of keys with the given \meta{key path}.
\end{docCommand}
```

`\foomakedocSubKey*{`⟨*name*⟩`}{`⟨*key path*⟩`}`
Creates a new environment ⟨*name*⟩ based on `docKey` →P. 307 for the documentation of keys with the given ⟨*key path*⟩.

```
\begin{docCommand*}[⟨options⟩]{⟨name⟩}{⟨parameters⟩}
  ⟨command description⟩
\end{docCommand*}
```

Identical to docCommand → P. 306, but without index entry.

```
\begin{docEnvironment}[⟨options⟩]{⟨name⟩}{⟨parameters⟩}
  ⟨environment description⟩
\end{docEnvironment}
```

Documents a LaTeX environment with given ⟨*name*⟩. The given ⟨*options*⟩ are set with \tcbset → P. 10. This environment takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with \refEnv → P. 312{⟨*name*⟩}.

```
\begin{docEnvironment}{foocolorbox}{\oarg{options}}
  This is the main environment to create an accentuated colored text box with
  rounded corners and, optionally, two parts.
\end{docEnvironment}
```

```
\begin{foocolorbox}[⟨options⟩]
  ⟨environment content⟩
\end{foocolorbox}
```

This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts.

```
\begin{docEnvironment}%
    [doclang/environment content=My content text]%
    {foocolorbox*}{\oarg{options}}
  This is the main environment to create an accentuated colored text box with
  rounded corners and, optionally, two parts.
\end{docEnvironment}
```

```
\begin{foocolorbox*}[⟨options⟩]
  ⟨My content text⟩
\end{foocolorbox*}
```

This is the main environment to create an accentuated colored text box with rounded corners and, optionally, two parts.

```
\begin{docEnvironment*}[⟨options⟩]{⟨name⟩}{⟨parameters⟩}
  ⟨environment description⟩
\end{docEnvironment*}
```

Identical to docEnvironment, but without index entry.

U 2014-09-29

```
\begin{docKey}[⟨key path⟩][⟨options⟩]{⟨name⟩}{⟨parameters⟩}{⟨description⟩}
  ⟨key description⟩
\end{docKey}
```

Documents a key with given ⟨*name*⟩ and an optional ⟨*key path*⟩. The given ⟨*options*⟩ are set with \tcbset → P. 10. This key takes mandatory or optional ⟨*parameters*⟩ as value with a short ⟨*description*⟩. It is automatically indexed and can be referenced with \refKey → P. 313{⟨*name*⟩}.

```
\begin{docKey}[foo]{footitle}{=\meta{text}}{no default, initially empty}
  Creates a heading line with \meta{text} as content.
\end{docKey}
```

/foo/footitle=⟨*text*⟩ (no default, initially empty)

Creates a heading line with ⟨*text*⟩ as content.

U 2014-09-29 `\begin{docKey*}[⟨key path⟩][⟨options⟩]{⟨name⟩}{⟨parameters⟩}{⟨description⟩}`
`⟨key description⟩`
`\end{docKey*}`

Identical to docKey → P. 307, but without index entry.

`\docValue{⟨name⟩}`

Documents a value with given ⟨*name*⟩. Typically, this is a value for a key. This value is automatically indexed.

```
A feasible value for \refKey{/foo/footitle} is \docValue{foovalue}.
```

A feasible value for /foo/footitle → P. 307 is foovalue.

`\docValue*{⟨name⟩}`

Identical to \docValue, but without index entry.

U 2014-10-09 `\docAuxCommand{⟨name⟩}`

Documents an auxiliary or minor LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. This macro is automatically indexed.

```
The macro \docAuxCommand{fooaux} holds some interesting data.
```

The macro \fooaux holds some interesting data.

U 2014-10-09 `\docAuxCommand*{⟨name⟩}`

Identical to \docAuxCommand, but without index entry.

`\docAuxEnvironment{⟨name⟩}`

Documents an auxiliary or minor LaTeX environment with given ⟨*name*⟩. This macro is automatically indexed.

```
The environment \docAuxEnvironment{fooauxenv} holds some interesting data.
```

The environment fooauxenv holds some interesting data.

`\docAuxEnvironment*[⟨key path⟩]{⟨name⟩}`

Identical to \docAuxEnvironment, but without index entry.

`\docAuxKey[⟨key path⟩]{⟨name⟩}`

Documents an auxiliary key with given ⟨*name*⟩ and an optional ⟨*key path*⟩. It is automatically indexed.

```
The key \docAuxKey[foo]{fooaux} holds some interesting data.
```

The key /foo/fooaux holds some interesting data.

`\docAuxKey*[⟨key path⟩]{⟨name⟩}`

Identical to \docAuxKey, but without index entry.

`\docColor{`⟨*name*⟩`}`

Documents a color with given ⟨*name*⟩. The color is automatically indexed.

```
The color \docColor{foocolor} is available.
```

The color foocolor is available.

`\docColor*{`⟨*name*⟩`}`

Identical to `\docColor`, but without index entry.

`\cs{`⟨*name*⟩`}`

Macro from `ltxdoc` [3] to typeset a command word ⟨*name*⟩ where the backslash is prefixed. The library overwrites the original macro.

```
This is a \cs{foocommand}.
```

This is a \foocommand.

`\meta{`⟨*text*⟩`}`

Macro from `doc` [8] to typeset a meta ⟨*text*⟩. The library overwrites the original macro.

```
This is a \meta{text}.
```

This is a ⟨*text*⟩.

`\marg{`⟨*text*⟩`}`

Macro from `ltxdoc` [3] to typeset a ⟨*text*⟩ with curly brackets as a mandatory argument. The library overwrites the original macro.

```
This is a mandatory \marg{argument}.
```

This is a mandatory {⟨*argument*⟩}.

`\oarg{`⟨*text*⟩`}`

Macro from `ltxdoc` [3] to typeset a ⟨*text*⟩ with square brackets as an optional argument. The library overwrites the original macro.

```
This is an optional \oarg{argument}.
```

This is an optional [⟨*argument*⟩].

`\brackets{`⟨*text*⟩`}`

Sets the given ⟨*text*⟩ with curly brackets.

```
  Here we use \brackets{some text}.
```

Here we use {some text}.

U 2014-10-10

```
\begin{dispExample}
 ⟨environment content⟩
\end{dispExample}
```

Creates a colored box based on a tcolorbox → P. 9. It displays the environment content as source code in the upper part and as compiled text in the lower part of the box. The appearance is controlled by /tcb/documentation listing style → P. 315 and the style /tcb/docexample → P. 315. It may be changed by redefining this style.

```
\begin{dispExample}
This is a \LaTeX\ example.
\end{dispExample}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

U 2014-10-10

```
\begin{dispExample*}{⟨options⟩}
 ⟨environment content⟩
\end{dispExample*}
```

The starred version of dispExample takes tcolorbox → P. 9 ⟨*options*⟩ as parameter. These ⟨*options*⟩ are executed after /tcb/docexample → P. 315.

```
\begin{dispExample*}{sidebyside}
This is a \LaTeX\ example.
\end{dispExample*}
```

```
This is a \LaTeX\ example.
```

This is a LaTeX example.

```
\begin{dispListing}
 ⟨environment content⟩
\end{dispListing}
```

Creates a colored box based on a tcolorbox → P. 9. It displays the environment content as source code. The appearance is controlled by /tcb/documentation listing style → P. 315 and the style /tcb/docexample → P. 315. It may be changed by redefining this style.

```
\begin{dispListing}
This is a \LaTeX\ example.
\end{dispListing}
```

```
This is a \LaTeX\ example.
```

```
\begin{dispListing*}{⟨options⟩}
 ⟨environment content⟩
\end{dispListing*}
```

The starred version of dispListing takes tcolorbox → P. 9 ⟨*options*⟩ as parameter. These ⟨*options*⟩ are executed after /tcb/docexample → P. 315.

```
\begin{dispListing*}{title=My listing}
This is a \LaTeX\ example.
\end{dispListing*}
```

My listing

```
This is a \LaTeX\ example.
```

```
\begin{absquote}
 ⟨environment content⟩
\end{absquote}
```

Used to typeset an abstract as quoted and small text.

```
\begin{absquote}
|tcolorbox| provides an environment for colored and framed text boxes with a
heading line. Optionally, such a box can be split in an upper and a lower part.
\end{absquote}
```

> tcolorbox provides an environment for colored and framed text boxes with a heading line. Optionally, such a box can be split in an upper and a lower part.

`\tcbmakedocSubKey{`⟨*name*⟩`}{`⟨*key path*⟩`}`

Creates a new environment ⟨*name*⟩ based on docKey → P. 307 for the documentation of keys with the given ⟨*key path*⟩ as default. The new environment ⟨*name*⟩ takes the same parameters as docKey → P. 307 itself. A second starred environment ⟨*name*⟩ is also created, which is identical to ⟨*name*⟩ but without index entry.

```
\tcbmakedocSubKey{docFooKey}{foo}

\begin{docFooKey}{foodummy}{=\meta{nothing}}{no default, initially empty}
Some key.
\end{docFooKey}

\begin{docFooKey*}{foo another dummy}{=\meta{nothing}}{no default, initially empty}
Some key (not indexed).
\end{docFooKey*}
```

`/foo/foodummy=`⟨*nothing*⟩ (no default, initially empty)
Some key.

`/foo/foo another dummy=`⟨*nothing*⟩ (no default, initially empty)
Some key (not indexed).

`\refCom{`⟨*name*⟩`}`

References a documented LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The page reference is suppressed if it links to the same page.

```
We have created \refCom{foomakedocSubKey} as an example.
```

We have created \foomakedocSubKey → P. 306 as an example.

`\refCom*{`⟨*name*⟩`}`

References a documented LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. There is no page reference.

```
We have created \refCom*{foomakedocSubKey} as an example.
```

We have created \foomakedocSubKey as an example.

`\refEnv{`⟨*name*⟩`}`

References a documented LaTeX environment with given ⟨*name*⟩. The page reference is suppressed if it links to the same page.

```
We have created \refEnv{foocolorbox} as an example.
```

We have created foocolorbox → P. 307 as an example.

`\refEnv*{`⟨*name*⟩`}`

References a documented LaTeX environment with given ⟨*name*⟩. There is no page reference.

```
We have created \refEnv*{foocolorbox} as an example.
```

We have created foocolorbox as an example.

`\refKey{⟨name⟩}`

References a documented key with given ⟨*name*⟩ where ⟨*name*⟩ is the full path name of the key. The page reference is suppressed if it links to the same page.

```
We have created \refKey{/foo/footitle} as an example.
```

We have created /foo/footitle → P. 307 as an example.

`\refKey*{⟨name⟩}`

References a documented key with given ⟨*name*⟩ where ⟨*name*⟩ is the full path name of the key. There is no page reference.

```
We have created \refKey*{/foo/footitle} as an example.
```

We have created /foo/footitle as an example.

`\refAux{⟨name⟩}`

References some auxiliary environment, key, value, or color. The hyperlink color is used, but there is no real link.

```
Some pages back, one can see \refAux{/foo/footitle} as an example.
```

Some pages back, one can see /foo/footitle as an example.

`\refAuxcs{⟨name⟩}`

References some auxiliary macro ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. The hyperlink color is used, but there is no real link.

```
Some pages back, one can see \refAuxcs{fooaux} as an example.
```

Some pages back, one can see \fooaux as an example.

`\colDef{⟨text⟩}`

Sets ⟨*text*⟩ with the command color, see /tcb/color command → P. 315.

```
This is my \colDef{text}.
```

This is my text.

`\colOpt{⟨text⟩}`

Sets ⟨*text*⟩ with the option color, see /tcb/color option → P. 315.

```
This is my \colOpt{text}.
```

This is my text.

N 2014-09-19 **\tcbdocmarginnote**[⟨*options*⟩]{⟨*text*⟩}

Creates a `tcolorbox` note with the given ⟨*text*⟩ inside the margin using the `marginnote` package. The style of the `tcolorbox` is predefined and can be altered by `/tcb/doc marginnote` → P. 320 and the given ⟨*options*⟩.

```
Some text\tcbdocmarginnote{Note A}
which is commented by a note inside the margin.
Alternatively to |\tcbdocmarginnote|, you can always use
|\marginnote| with a |tcolorbox| directly.\par
This is further text%
\tcbdocmarginnote[colframe=blue!50!white,colback=blue!5!white]{Note B}
with another note.
```

Note A

Note B

Some text which is commented by a note inside the margin. Alternatively to `\tcbdocmarginnote`, you can always use `\marginnote` with a `tcolorbox` directly.
This is further text with another note.

N 2014-09-19 **\tcbdocnew**{⟨*date*⟩}

Auxiliary macro which typesets the `/tcb/doclang/new` → P. 317 text with the given ⟨*date*⟩. It may be redefined for customization.

```
\tcbdocnew{1981-10-29}.
% Next one is displayed in the margin:
\tcbdocmarginnote{\tcbdocnew{1978-02-09}}
```

New: 1978-02-09

New: 1981-10-29.

N 2014-09-19 **\tcbdocupdated**{⟨*date*⟩}

Auxiliary macro which typesets the `/tcb/doclang/updated` → P. 317 text with the given ⟨*date*⟩. It may be redefined for customization.

```
\tcbdocupdated{2014-09-19}.
```

Updated: 2014-09-19.

15.2 Option Keys of the Library

`/tcb/docexample` (style, no value)
Sets the style for `dispExample` → P. 310 and `dispListing` → P. 311 with the colors `ExampleBack` and `ExampleFrame`. To change the appearance of the examples, this style can be redefined.

`/tcb/documentation listing options=`⟨*key list*⟩ (no default, initially `style=tcbdocumentation`)
Sets the options from the package `listings` [6]. They are used inside `dispExample` → P. 310 and `dispListing` → P. 311 to typeset the listings. Note that this is not identical to the key `/tcb/listing options` → P. 206 which is used for 'normal' listings.
Used for `/tcb/listing engine` → P. 211`=listings` only.

`/tcb/documentation listing style=`⟨*listing style*⟩ (no default, initially `tcbdocumentation`)
Abbreviation for `documentation listing options={style=...}`. This key sets a ⟨*style*⟩ for the `listings` package, see [6]. Note that this is not identical to the key `/tcb/listing style` → P. 206 which is used for 'normal' listings.
Used for `/tcb/listing engine` → P. 211`=listings` only.

`/tcb/documentation minted style=`⟨*key list*⟩ (no default, initially unset)
Sets a ⟨*style*⟩ known to `Pygments` [12] for the package `minted` [14], if used. Note that this is not identical to the key `/tcb/minted style` → P. 210 which is used for 'normal' listings.
Used for `/tcb/listing engine` → P. 211`=minted` only.

`/tcb/documentation minted options=`⟨*minted style*⟩ (no default, initially `tabsize=2,fontsize=\small`)
Sets the options from the package `minted` [14] which are used during typesetting of the listing, if used. Note that this is not identical to the key `/tcb/minted options` → P. 209 which is used for 'normal' listings.
Used for `/tcb/listing engine` → P. 211`=minted` only.

`/tcb/color command=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by macro definitions.

`/tcb/color environment=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by environment definitions.

`/tcb/color key=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by key definitions.

`/tcb/color value=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by value definitions.

`/tcb/color color=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by color definitions.

`/tcb/color definition=`⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color for `/tcb/color command`, `/tcb/color environment`, `/tcb/color key`, `/tcb/color value`, and `/tcb/color color`.

`/tcb/color option=`⟨*color*⟩ (no default, initially `Option`)
Sets the color used for optional arguments.

`/tcb/color hyperlink=`⟨*color*⟩ (no default, initially `Hyperlink`)
Sets the color for all hyper-links, i. e. all internal and external links.

`/tcb/before example=`⟨*macros*⟩ (no default, initially `\par\smallskip`)
Sets the ⟨*macros*⟩ which are executed before `dispExample` → P. 310 and `dispListing` → P. 311 additional to `/tcb/before` → P. 65.

`/tcb/after example=`⟨*macros*⟩ (no default, initially empty)

Sets the ⟨*macros*⟩ which are executed after `dispExample` → P. 310 and `dispListing` → P. 311 additional to `/tcb/after` → P. 65.

`/tcb/index format=`⟨*format*⟩ (no default, initially `pgf`)

Determines the basic ⟨*format*⟩ of the generated index. Feasible values are:

- `pgfsection`: The index is formatted like in the `pgf` documentation (as a section).
- `pgfchapter`: The index is formatted like in the `pgf` documentation (as a chapter).
- `pgf`: Alias for `pgfsection`.
- `doc`: The index is assumed to be formatted by `doc` or `ltxdoc`. The usage of `makeindex` with `-s gind.ist` is assumed. The package `hypdoc` has to be loaded *before* `tcolorbox`.
- `off`: The index is not formatted by `tcolorbox`.

`/tcb/index actual=`⟨*character*⟩ (no default, initially `@`)

Sets the character for 'actual' in automatic indexing.

`/tcb/index quote=`⟨*character*⟩ (no default, initially `"`)

Sets the character for 'quote' in automatic indexing.

`/tcb/index level=`⟨*character*⟩ (no default, initially `!`)

Sets the character for 'level' in automatic indexing.

`/tcb/index default settings` (style, no value)

Sets the `makeindex` default values for `/tcb/index actual`, `/tcb/index quote`, and `/tcb/index level`.

`/tcb/index german settings` (style, no value)

Sets the `makeindex` values recommended for German language texts. This is identical to setting the following:

```
\tcbset{index actual={=},index quote={!},index level={>}}
```

`/tcb/index annotate=true|false` (default `true`, initially `true`)

If set to `true`, the index entries are annotated with short descriptions given by `/tcb/doclang/environment` → P. 317, `/tcb/doclang/key` → P. 317, and others.

`/tcb/index colorize=true|false` (default `true`, initially `false`)

If set to `true`, the index entries colorized according to the color settings given by `/tcb/color environment` → P. 315, `/tcb/color key` → P. 315, and others.

The following keys are provided for language specific settings. The English language is predefined.

`/tcb/english language` (style, no value)
Sets all language specific settings to English.

`/tcb/doclang/color`=⟨*text*⟩ (no default, initially `color`)
Text used in the index for colors.

`/tcb/doclang/colors`=⟨*text*⟩ (no default, initially `Colors`)
Heading text in the index for colors.

`/tcb/doclang/environment`=⟨*text*⟩ (no default, initially `environment`)
Text used in the index for environments.

`/tcb/doclang/environments`=⟨*text*⟩ (no default, initially `Environments`)
Heading text in the index for environments.

`/tcb/doclang/environment content`=⟨*text*⟩ (no default, initially `environment content`)
Text used in `docEnvironment` → P. 307.

`/tcb/doclang/key`=⟨*text*⟩ (no default, initially `key`)
Text used in the index for keys.

`/tcb/doclang/keys`=⟨*text*⟩ (no default, initially `Keys`)
Heading text used in the index for keys.

`/tcb/doclang/index`=⟨*text*⟩ (no default, initially `Index`)
Heading text for the index.

N 2014-09-19 `/tcb/doclang/new`=⟨*text*⟩ (no default, initially `New`)
Announcement text for new content.

`/tcb/doclang/pageshort`=⟨*text*⟩ (no default, initially `P.`)
Short text for page references.

N 2014-09-19 `/tcb/doclang/updated`=⟨*text*⟩ (no default, initially `Updated`)
Announcement text for updated content.

`/tcb/doclang/value`=⟨*text*⟩ (no default, initially `value`)
Text used in the index for values.

`/tcb/doclang/values`=⟨*text*⟩ (no default, initially `Values`)
Heading text in the index for values.

/tcb/doc left=⟨*length*⟩ (no default, initially 2em)

Sets the left hand offset of the documentation texts from docCommand → P. 306, docEnvironment → P. 307, docKey → P. 307, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc left=2cm,doc left indent=-2cm]{myCommandA}{\marg{argment}}
  This is the documentation of \refCom{myCommandA} which takes one \meta{argument}.
  \refCom{myCommandA} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandA{⟨*argment*⟩}
This is the documentation of \myCommandA which takes one ⟨*argument*⟩. \myCommandA does some funny things with its ⟨*argument*⟩.

/tcb/doc right=⟨*length*⟩ (no default, initially 0em)

Sets the right hand offset of the documentation texts from docCommand → P. 306, docEnvironment → P. 307, docKey → P. 307, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc right=2cm]{myCommandB}{\marg{argment}}
  This is the documentation of \refCom{myCommandB} which takes one \meta{argument}.
  \refCom{myCommandB} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandB{⟨*argment*⟩}
This is the documentation of \myCommandB which takes one ⟨*argument*⟩. \myCommandB does some funny things with its ⟨*argument*⟩.

/tcb/doc left indent=⟨*length*⟩ (no default, initially -2em)

Sets the left hand indent of documentation heads from docCommand → P. 306, docEnvironment → P. 307, docKey → P. 307, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc left indent=2cm]{myCommandC}{\marg{argment}}
  This is the documentation of \refCom{myCommandC} which takes one \meta{argument}.
  \refCom{myCommandC} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandC{⟨*argment*⟩}
This is the documentation of \myCommandC which takes one ⟨*argument*⟩. \myCommandC does some funny things with its ⟨*argument*⟩.

/tcb/doc right indent=⟨*length*⟩ (no default, initially 0pt)

Sets the right hand indent of documentation heads from docCommand → P. 306, docEnvironment → P. 307, docKey → P. 307, etc, to ⟨*length*⟩.

```
\begin{docCommand*}[doc right indent=-10mm,doc right=10mm,
    doc description=test value]{myCommandD}{\marg{argment}}
  This is the documentation of \refCom{myCommandD} which takes one \meta{argument}.
  \refCom{myCommandD} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandD{⟨*argment*⟩} (test value)
This is the documentation of \myCommandD which takes one ⟨*argument*⟩. \myCommandD does some funny things with its ⟨*argument*⟩.

The head lines of the main documentation environments docCommand → P. 306, docEnvironment → P. 307, docKey → P. 307, etc, are set inside tcolorboxes. Options to these tcolorboxes can be given using the following keys.

/tcb/doc head command=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docCommand → P. 306 and docCommand* → P. 307.

```
\tcbset{doc head command={interior style={fill,left color=red!20!white,
  right color=blue!20!white}}}

\begin{docCommand*}{myCommandE}{\marg{argment}}
  This is the documentation of \refCom{myCommandE} which takes one \meta{argument}.
  \refCom{myCommandE} does some funny things with its \meta{argument}.
\end{docCommand*}
```

\myCommandE{⟨*argment*⟩}

This is the documentation of \myCommandE which takes one ⟨*argument*⟩. \myCommandE does some funny things with its ⟨*argument*⟩.

/tcb/doc head environment=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docEnvironment → P. 307 and docEnvironment* → P. 307.

```
\tcbset{doc head environment={beamer,boxsep=2pt,arc=2pt,colback=green!20!white,
  after=\par\smallskip}}

\begin{docEnvironment*}{myEnvironment}{\marg{argment}}
  This is the documentation of \refEnv{myEnvironment} which
  takes one \meta{argument}.
\end{docEnvironment*}
```

```
\begin{myEnvironment}{⟨argment⟩}
  ⟨environment content⟩
\end{myEnvironment}
```

This is the documentation of myEnvironment which takes one ⟨*argument*⟩.

/tcb/doc head key=⟨*options*⟩ (no default, initially empty)

Sets ⟨*options*⟩ for the head line of docKey → P. 307 and docKey* → P. 308.

```
\tcbset{doc head key={boxsep=4pt,arc=4pt,boxrule=0.6pt,
  frame style=fill,interior style=fill,colframe=green!50!black}}

\begin{docKey*}{/foo/myKey}{}{no value}
  This is the documentation of \refKey{/foo/myKey}.
\end{docKey*}
```

/foo/myKey (no value)

This is the documentation of /foo/myKey.

/tcb/doc head=⟨*options*⟩ (no default, initially empty)

Shortcut for setting the same ⟨*options*⟩ for /tcb/doc head command, /tcb/doc head environment, and /tcb/doc head key.

`/tcb/doc description=`⟨*text*⟩ (no default, initially empty)

Sets a (short!) additional description ⟨*text*⟩ for `docCommand` → P. 306 or `docEnvironment` → P. 307. Such a description is mandatory for `docKey` → P. 307.

```
\begin{docCommand*}[doc description=my description]{myCommandF}{\marg{argment}}
  This is the documentation of \refCom{myCommandF} which takes one \meta{argument}.
  \refCom{myCommandF} does some funny things with its \meta{argument}.
\end{docCommand*}
```

`\myCommandF{`⟨*argment*⟩`}` (my description)

This is the documentation of `\myCommandF` which takes one ⟨*argument*⟩. `\myCommandF` does some funny things with its ⟨*argument*⟩.

Note that the description ⟨*text*⟩ may overlap with the text on the left hand side if too long. Linebreaks can be used inside the ⟨*text*⟩.

`/tcb/doc into index=true|false` (default `true`, initially `true`)

If set to `false`, no index entries are written for the main documentation environments. The same effect is achieved by using e. g. `docCommand*` → P. 307 instead of `docCommand` → P. 306.

N 2014-09-19 `/tcb/doc marginnote=`⟨*options*⟩ (no default, initially empty)

Sets style ⟨*options*⟩ for the displayed box of the `\tcbdocmarginnote` → P. 314 command.

```
\tcbset{doc marginnote={colframe=blue!50!white,colback=blue!5!white}}%
This is some text\tcbdocmarginnote{Note A}
which is commented by a note inside the margin.
```

Note A

This is some text which is commented by a note inside the margin.

N 2014-09-19 `/tcb/doc new=`⟨*date*⟩ (style, no default)

Adds a a marginnote with a 'New: ⟨*data*⟩' message at the beginning of the upper box part. The intended use is inside the option list of `docCommand` → P. 306, `docEnvironment` → P. 307, etc.

```
\begin{docCommand}[doc new=2000-01-01]{foosomething}{\marg{text}}
Some command for something.
\end{docCommand}
```

New: 2000-01-01

`\foosomething{`⟨*text*⟩`}`

Some command for something.

N 2014-09-19 `/tcb/doc updated=`⟨*date*⟩ (style, no default)

Adds a marginnote with a 'Updated: ⟨*data*⟩' message at the beginning of the upper box part. See `/tcb/doc new`.

N 2014-09-19 `/tcb/doc new and updated={`⟨*new date*⟩`}{`⟨*update date*⟩`}` (style, no default)

Adds a marginnote with 'New: ⟨*new date*⟩' and 'Updated: ⟨*update data*⟩' messages at the beginning of the upper box part. See `/tcb/doc new`.

15.3 Predefined Colors of the Library

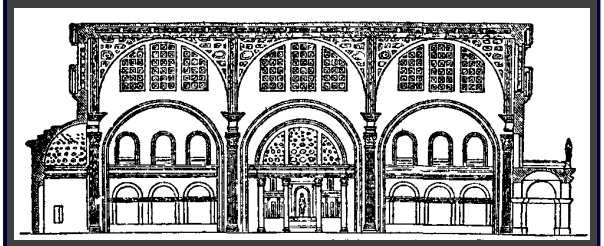
The following colors are predefined. They are used as default colors in some library commands.

`Option` , `Definition` , `ExampleFrame` , `ExampleBack` , `Hyperlink` .

A Picture Credits

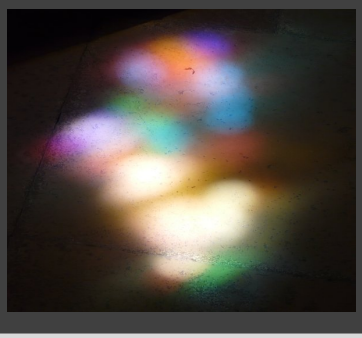
The following pictures were used inside this documentation.

Basilica_5.png



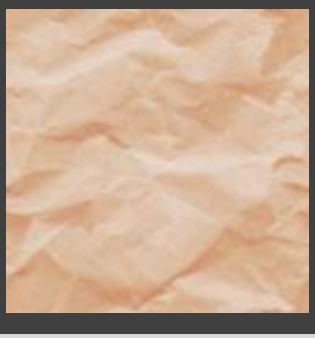
http://commons.wikimedia.org/wiki/File:Basilica_5.png

lichtspiel.jpg



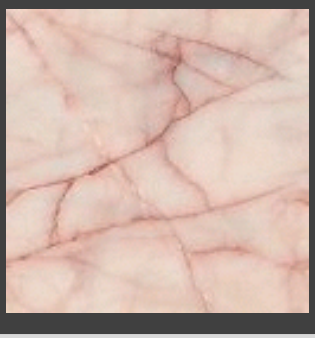
Photograph taken by Thomas F. Sturm.

crinklepaper.png



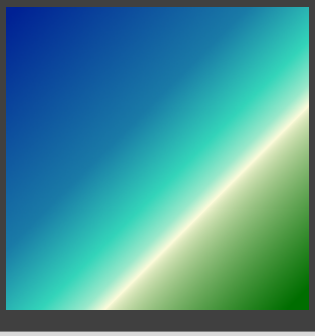
Created with GIMP.
http://www.gimp.org

pink_marble.png



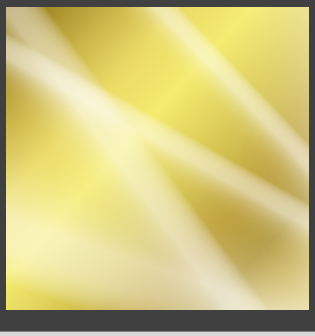
Created with GIMP.
http://www.gimp.org

blueshade.png



Created with GIMP.
http://www.gimp.org

goldshade.png



Created with GIMP.
http://www.gimp.org

References

[1] Donald Arseneau. *The varwidth package*. Sept. 26, 2011.
mirrors.ctan.org/macros/latex/contrib/varwidth/varwidth-doc.pdf.

[2] Peter Breitenlohner. *The ε-TEX manual*. Version 2. Feb. 1998.
http://ftp.gwdg.de/pub/ctan/systems/e-tex/v2/doc/etex_man.pdf.

[3] David Carlisle. *The file ltxdoc.dtx for use with LaTeX 2ε*. Nov. 11, 2007.
http://mirror.ctan.org/macros/latex/base/ltxdoc.dtx.

[4] David Carlisle. *The tabularx package*. May 13, 2014.
http://mirror.ctan.org/macros/latex/required/tools/tabularx.pdf.

[5] Toby Cubitt. *The cleveref Package*. Mar. 22, 2013.
http://mirror.ctan.org/macros/latex/contrib/cleveref/cleveref.pdf.

[6] Carsten Heinz, Brooks Moses, and Jobst Hoffmann. *The Listings Package*. Version 1.5b. Aug. 26, 2013.
http://mirror.ctan.org/macros/latex/contrib/listings/listings.pdf.

[7] Philipp Lehman. *The etoolbox Package*. Jan. 3, 2011.
http://mirror.ctan.org/macros/latex/contrib/etoolbox/etoolbox.pdf.

[8] Frank Mittelbach. *The doc and shortvrb Packages*. June 30, 2011.
http://mirror.ctan.org/macros/latex/base/doc.dtx.

[9] Frank Mittelbach. *An environment for multicolumn output*. June 21, 2014.
http://mirror.ctan.org/macros/latex/required/tools/multicol.pdf.

[10] Heiko Oberdiek. *The listingsutf8 package*. Nov. 10, 2011.
http://mirror.ctan.org/macros/latex/contrib/oberdiek/listingsutf8.pdf.

[11] The LaTeX3 Project. *The xparse Package*. Dec. 31, 2013.
http://mirror.ctan.org/macros/latex/contrib/l3packages/xparse.pdf.

[12] *Pygments. Python syntax highlighter*. 2013.
http://pygments.org/.

[13] Will Robertson. *A couple of things involving environments*. Apr. 1, 2013.
http://mirror.ctan.org/macros/latex/contrib/environ/environ.pdf.

[14] Konrad Rudolph. *The minted package. Highlighted source code in LaTeX*. Sept. 17, 2011.
http://mirror.ctan.org/macros/latex/contrib/minted/minted.pdf.

[15] Rainer Schöpf, Bernd Raichle, and Chris Rowley. *A New Implementation of LaTeX's verbatim and verbatim* Environments*. Mar. 12, 2001.
http://mirror.ctan.org/macros/latex/required/tools/verbatim.pdf.

[16] Thomas F. Sturm. *Mathematik 1 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2007.
http://www.unibw.de/bw/sturm.

[17] Thomas F. Sturm. *Mathematik 2 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2008.
http://www.unibw.de/bw/sturm.

[18] Thomas F. Sturm. *Höhere Mathematik: Fortgeschrittene mathematische Methoden (M.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2010.
http://www.unibw.de/bw/sturm.

[19] Thomas F. Sturm. *LaTeX – Einführung in das Textsatzsystem*. 10th edition. RRZN-Handbücher. Hannover: Regionales Rechenzentrum für Niedersachsen (RRZN), Mar. 2014.
http://www.rrzn.uni-hannover.de/buch.html?&titel=latex.

[20] Till Tantau. *The TikZ and PGF Packages. Manual for version 3.0.0*. Dec. 20, 2013.
http://sourceforge.net/projects/pgf/.

[21] Kresten Krab Thorub, Frank Jensen, and Chris Rowley. *The calc package – Infix notation arithmetic in LaTeX*. Aug. 22, 2007.
http://mirror.ctan.org/macros/latex/required/tools/.

Index

absquote environment, 311
add to list key, 80
adjust text key, 14
adjusted title key, 14
adjusted title after break key, 256
after key, 65
after app key, 285
after example key, 316
after lower key, 47
after lower app key, 285
after lower pre key, 285
after pre key, 285
after skip key, 66
after title key, 46
after title app key, 284
after title pre key, 284
after upper key, 47
after upper app key, 284
after upper pre key, 284
all key, 8
all value, 34, 35, 260
ams align key, 242
ams align lower key, 242
ams align upper key, 242
ams align* key, 242
ams align* lower key, 242
ams align* upper key, 242
ams equation key, 241
ams equation lower key, 241
ams equation upper key, 241
ams equation* key, 241
ams equation* lower key, 241
ams equation* upper key, 241
ams gather key, 243
ams gather lower key, 243
ams gather upper key, 243
ams gather* key, 243
ams gather* lower key, 243
ams gather* upper key, 243
ams nodisplayskip key, 244
ams nodisplayskip lower key, 244
ams nodisplayskip upper key, 244
arc key, 25
areasize value, 281
at begin tikz key, 138
at begin tikz reset key, 138
at end tikz key, 138
at end tikz reset key, 138
attach boxed title to bottom center key, 107
attach boxed title to bottom left key, 107
attach boxed title to bottom right key, 107
attach boxed title to top center key, 107
attach boxed title to top left key, 107
attach boxed title to top right key, 107
attach title key, 15
attach title to upper key, 15
auto counter key, 82
auto outer arc key, 25
autoparskip key, 65

baseline key, 65
beamer key, 168
beamer Skin, 168
beamerfirst Skin, 170
beamerlast Skin, 172
beamermiddle Skin, 171
before key, 65
before app key, 285
before example key, 315
before lower key, 47
before lower app key, 285
before lower pre key, 285
before pre key, 285
before skip key, 66
before title key, 46
before title app key, 284
before title pre key, 284
before upper key, 46
before upper app key, 284
before upper pre key, 284
beforeafter skip key, 66
bicolor key, 163
bicolor Skin, 163
bicolorfirst Skin, 165
bicolorlast Skin, 167
bicolormiddle Skin, 166
blank key, 152
blanker key, 178
blend into key, 86
borderline key, 126
bottom key, 29
bottom value, 62
bottomrule key, 23
bottomrule at break key, 259
bottomsep at break key, 259
bottomtitle key, 29
boxed title style key, 109
boxrule key, 24
boxsep key, 25
\brackets, 309
break value, 246
break at key, 257
breakable key, 7, 255
broken value, 114–116

capture key, 76
center value, 62
center lower key, 48
center title key, 48
center upper key, 48
change value, 246

change apart value, 246
change break value, 246
change standard value, 245
check odd page key, 80
clip lower key, 125
clip title key, 124
clip upper key, 124
clip watermark key, 120
code key, 54
colback key, 20
colbacklower key, 164
colbacktitle key, 20
\colDef, 313
colframe key, 20
collower key, 21
\colOpt, 313
color key, 317
color color key, 315
color command key, 315
color definition key, 315
color environment key, 315
color hyperlink key, 315
color key key, 315
color option key, 315
color value key, 315
Colors
  Definition, 320
  ExampleBack, 320
  ExampleFrame, 320
  foocolor, 309
  Hyperlink, 320
  Option, 320
colors key, 317
coltext key, 21
coltitle key, 21
colupper key, 21
comment key, 212
comment above listing key, 217
comment and listing key, 213
comment outside listing key, 215
comment side listing key, 215
Crefname key, 85
crefname key, 85
\cs, 309

\DeclareTCBInputListing, 303
\DeclareTCBListing, 301
\DeclareTCBox, 298
\DeclareTCBoxFit, 304
\DeclareTColorBox, 295
\DeclareTotalTCBox, 299
\DeclareTotalTCBoxFit, 305
\DeclareTotalTColorBox, 297
Definition color, 320
description color key, 236
description delimiters key, 236
description delimiters none key, 236
description delimiters parenthesis key, 236
description font key, 237
description formatter key, 237
detach title key, 15
dispExample environment, 310
dispExample* environment, 310
dispListing environment, 311
dispListing* environment, 311
doc value, 316
doc description key, 320
doc head key, 319
doc head command key, 319
doc head environment key, 319
doc head key key, 319
doc into index key, 320
doc left key, 318
doc left indent key, 318
doc marginnote key, 320
doc new key, 320
doc new and updated key, 320
doc right key, 318
doc right indent key, 318
doc updated key, 320
\docAuxCommand, 308
\docAuxCommand*, 308
\docAuxEnvironment, 308
\docAuxEnvironment*, 308
\docAuxKey, 308
\docAuxKey*, 308
\docColor, 309
\docColor*, 309
docCommand environment, 306
docCommand* environment, 307
docEnvironment environment, 307
docEnvironment* environment, 307
docexample key, 315
docKey environment, 307
docKey* environment, 308
documentation key, 7
documentation listing options key, 315
documentation listing style key, 315
documentation minted options key, 315
documentation minted style key, 315
\docValue, 308
\docValue*, 308
downhill value, 34, 35
draft key, 187
draft Skin, 187
draftmode key, 147
drop fuzzy midday shadow key, 131
drop fuzzy shadow key, 131
drop fuzzy shadow east key, 135
drop fuzzy shadow north key, 135
drop fuzzy shadow northeast key, 135
drop fuzzy shadow northwest key, 135
drop fuzzy shadow south key, 134
drop fuzzy shadow southeast key, 134
drop fuzzy shadow southwest key, 134
drop fuzzy shadow west key, 134
drop large lifted shadow key, 137
drop lifted shadow key, 136

drop midday shadow key, 131
drop shadow key, 131
drop shadow east key, 134
drop shadow north key, 133
drop shadow northeast key, 134
drop shadow northwest key, 133
drop shadow south key, 133
drop shadow southeast key, 133
drop shadow southwest key, 133
drop shadow west key, 133
drop small lifted shadow key, 136

east value, 34, 35
empty key, 177
empty Skin, 177
empty value, 90, 91
emptyfirst Skin, 179
emptylast Skin, 181
emptymiddle Skin, 180
enforce breakable key, 256
english language key, 317
enhanced key, 150
enhanced Skin, 150
enhanced jigsaw key, 157
enhanced jigsaw Skin, 157
enhanced standard key, 152
enhancedfirst Skin, 154
enhancedfirst jigsaw Skin, 158
enhancedlast Skin, 156
enhancedlast jigsaw Skin, 162
enhancedmiddle Skin, 155
enhancedmiddle jigsaw Skin, 159
enlarge bottom at break by key, 68
enlarge bottom by key, 68
enlarge bottom finally by key, 67
enlarge by key, 69
enlarge left by key, 68
enlarge right by key, 68
enlarge top at break by key, 68
enlarge top by key, 68
enlarge top initially by key, 67
enlargepage key, 257
enlargepage flexible key, 258
environment key, 317
environment content key, 317
Environments
    absquote, 311
    dispExample, 310
    dispExample*, 310
    dispListing, 311
    dispListing*, 311
    docCommand, 306
    docCommand*, 307
    docEnvironment, 307
    docEnvironment*, 307
    docKey, 307
    docKey*, 308
    fooauxenv, 308
    foocolorbox, 307
    foocolorbox*, 307
    tcbclipframe, 121
    tcbclipinterior, 123
    tcbcliptitle, 123
    tcbinvclipframe, 122
    tcblisting, 200
    tcboutputlisting, 202
    tcbverbatimwrite, 88
    tcbwritetemp, 88
    tcolorbox, 9
environments key, 317
equal height group key, 44
evenpage value, 32, 70
every box key, 73
every box on higher layers key, 74
every box on layer n key, 74
every float key, 61
every listing line key, 207
every listing line* key, 207
ExampleBack color, 320
ExampleFrame color, 320
extend freelance key, 189
extend freelancefirst key, 189
extend freelancelast key, 189
extend freelancemiddle key, 189
external key, 81
extrude bottom by key, 72
extrude by key, 72
extrude left by key, 71
extrude right by key, 71
extrude top by key, 72

false value, 255
fbox value, 30
figures value, 86
fill image opacity key, 196
fill image options key, 196
fill image scale key, 196
fill overzoom image key, 192
fill overzoom image* key, 192
fill overzoom picture key, 192
fill plain image key, 190
fill plain image* key, 190
fill plain picture key, 190
fill shrink image key, 194
fill shrink image* key, 194
fill shrink picture key, 194
fill stretch image key, 191
fill stretch image* key, 191
fill stretch picture key, 191
fill tile image key, 195
fill tile image* key, 195
fill tile picture key, 195
fill tile picture* key, 195
fill zoom image key, 193
fill zoom image* key, 193
fill zoom picture key, 193
final value, 282
finish key, 143
finish broken key, 144
finish broken pre key, 291

finish first key, 144
finish first and middle key, 144
finish first and middle pre key, 291
finish first pre key, 291
finish last key, 144
finish last pre key, 291
finish middle key, 144
finish middle and last key, 144
finish middle and last pre key, 291
finish middle pre key, 291
finish pre key, 291
finish unbroken key, 144
finish unbroken and first key, 144
finish unbroken and first pre key, 291
finish unbroken and last key, 144
finish unbroken and last pre key, 291
finish unbroken pre key, 291
first value, 114–116, 260
first and middle value, 114, 260
fit key, 275
fit algorithm key, 281
fit basedim key, 276
fit fontsize macros key, 277
fit height from key, 280
fit height plus key, 278
fit maxfontdiff key, 282
fit maxfontdiffgap key, 282
fit maxstep key, 282
fit maxwidthdiff key, 282
fit maxwidthdiffgap key, 282
fit skip key, 276
fit to key, 276
fit to height key, 276
fit warning key, 282
fit width from key, 279
fit width plus key, 278
fitbox value, 76
fitting key, 7
float key, 60
float* key, 60
floatplacement key, 60
flushleft lower key, 49
flushleft title key, 49
flushleft upper key, 49
flushright lower key, 50
flushright title key, 50
flushright upper key, 50
fontlower key, 22
fontsize value, 281
fonttitle key, 22
fontupper key, 22
\fooaux, 308
fooaux key, 308
fooauxenv environment, 308
foocolor color, 309
foocolorbox environment, 307
foocolorbox* environment, 307
foodummy key, 312
\foomakedocSubKey, 306
\foomakedocSubKey*, 306
\foosomething, 320
footitle key, 307
foovalue value, 308
forced value, 32, 70
frame code key, 93
frame code app key, 291
frame code pre key, 291
frame empty key, 93
frame engine key, 90
frame hidden key, 102
frame style key, 101
frame style image key, 101
frame style tile key, 102
freelance key, 189
freelance Skin, 189
freelance value, 90, 91
freelancefirst Skin, 189
freelancelast Skin, 189
freelancemiddle Skin, 189
fuzzy halo key, 132
fuzzy shadow key, 130

geometry nodes key, 92
graphical environment key, 89
grow to left by key, 69
grow to right by key, 69

halo key, 132
hbox key, 76
hbox value, 76
hbox boxed title key, 112
height key, 39
height fill key, 41
height fixed for key, 260
height from key, 40
height plus key, 39
highlight math key, 240
highlight math style key, 240
hooks key, 7
hybrid value, 281
Hyperlink color, 320
hyphenationfix key, 78

IfBooleanTF key, 294
IfNoValueTF key, 293
IfValueTF key, 294
ignored value, 17
image comment key, 212
index key, 317
index actual key, 316
index annotate key, 316
index colorize key, 316
index default settings key, 316
index format key, 316
index german settings key, 316
index level key, 316
index quote key, 316
interior code key, 94
interior code app key, 292

interior code pre key, 292
interior empty key, 94
interior engine key, 91
interior hidden key, 103
interior style key, 102
interior style image key, 103
interior style tile key, 103
interior titled code key, 93
interior titled code app key, 291
interior titled code pre key, 292
interior titled empty key, 93
interior titled engine key, 90
invisible value, 17

key key, 317
Keys
/foo/
fooaux, 308
foodummy, 312
footitle, 307
/tcb/
add to list, 80
adjust text, 14
adjusted title, 14
adjusted title after break, 256
after, 65
after app, 285
after example, 316
after lower, 47
after lower app, 285
after lower pre, 285
after pre, 285
after skip, 66
after title, 46
after title app, 284
after title pre, 284
after upper, 47
after upper app, 284
after upper pre, 284
ams align, 242
ams align lower, 242
ams align upper, 242
ams align*, 242
ams align* lower, 242
ams align* upper, 242
ams equation, 241
ams equation lower, 241
ams equation upper, 241
ams equation*, 241
ams equation* lower, 241
ams equation* upper, 241
ams gather, 243
ams gather lower, 243
ams gather upper, 243
ams gather*, 243
ams gather* lower, 243
ams gather* upper, 243
ams nodisplayskip, 244
ams nodisplayskip lower, 244
ams nodisplayskip upper, 244
arc, 25
at begin tikz, 138
at begin tikz reset, 138
at end tikz, 138
at end tikz reset, 138
attach boxed title to bottom center, 107
attach boxed title to bottom left, 107
attach boxed title to bottom right, 107
attach boxed title to top center, 107
attach boxed title to top left, 107
attach boxed title to top right, 107
attach title, 15
attach title to upper, 15
auto outer arc, 25
autoparskip, 65
baseline, 65
beamer, 168
before, 65
before app, 285
before example, 315
before lower, 47
before lower app, 285
before lower pre, 285
before pre, 285
before skip, 66
before title, 46
before title app, 284
before title pre, 284
before upper, 46
before upper app, 284
before upper pre, 284
beforeafter skip, 66
bicolor, 163
blank, 152
blanker, 178
borderline, 126
bottom, 29
bottomrule, 23
bottomrule at break, 259
bottomsep at break, 259
bottomtitle, 29
boxed title style, 109
boxrule, 24
boxsep, 25
break at, 257
breakable, 255
capture, 76
center lower, 48
center title, 48
center upper, 48
check odd page, 80
clip lower, 125
clip title, 124
clip upper, 124

clip watermark, 120
code, 54
colback, 20
colbacklower, 164
colbacktitle, 20
colframe, 20
collower, 21
color color, 315
color command, 315
color definition, 315
color environment, 315
color hyperlink, 315
color key, 315
color option, 315
color value, 315
coltext, 21
coltitle, 21
colupper, 21
comment, 212
comment above listing, 217
comment and listing, 213
comment outside listing, 215
comment side listing, 215
description color, 236
description delimiters, 236
description delimiters none, 236
description delimiters parenthesis, 236
description font, 237
description formatter, 237
detach title, 15
doc description, 320
doc head, 319
doc head command, 319
doc head environment, 319
doc head key, 319
doc into index, 320
doc left, 318
doc left indent, 318
doc marginnote, 320
doc new, 320
doc new and updated, 320
doc right, 318
doc right indent, 318
doc updated, 320
docexample, 315
documentation listing options, 315
documentation listing style, 315
documentation minted options, 315
documentation minted style, 315
draft, 187
draftmode, 147
drop fuzzy midday shadow, 131
drop fuzzy shadow, 131
drop fuzzy shadow east, 135
drop fuzzy shadow north, 135
drop fuzzy shadow northeast, 135
drop fuzzy shadow northwest, 135
drop fuzzy shadow south, 134
drop fuzzy shadow southeast, 134
drop fuzzy shadow southwest, 134
drop fuzzy shadow west, 134
drop large lifted shadow, 137
drop lifted shadow, 136
drop midday shadow, 131
drop shadow, 131
drop shadow east, 134
drop shadow north, 133
drop shadow northeast, 134
drop shadow northwest, 133
drop shadow south, 133
drop shadow southeast, 133
drop shadow southwest, 133
drop shadow west, 133
drop small lifted shadow, 136
empty, 177
enforce breakable, 256
english language, 317
enhanced, 150
enhanced jigsaw, 157
enhanced standard, 152
enlarge bottom at break by, 68
enlarge bottom by, 68
enlarge bottom finally by, 67
enlarge by, 69
enlarge left by, 68
enlarge right by, 68
enlarge top at break by, 68
enlarge top by, 68
enlarge top initially by, 67
enlargepage, 257
enlargepage flexible, 258
equal height group, 44
every box, 73
every box on higher layers, 74
every box on layer n, 74
every float, 61
every listing line, 207
every listing line*, 207
extend freelance, 189
extend freelancefirst, 189
extend freelancelast, 189
extend freelancemiddle, 189
external, 81
extrude bottom by, 72
extrude by, 72
extrude left by, 71
extrude right by, 71
extrude top by, 72
finish, 143
finish broken, 144
finish broken pre, 291
finish first, 144
finish first and middle, 144
finish first and middle pre, 291
finish first pre, 291
finish last, 144
finish last pre, 291

finish middle, 144
finish middle and last, 144
finish middle and last pre, 291
finish middle pre, 291
finish pre, 291
finish unbroken, 144
finish unbroken and first, 144
finish unbroken and first pre, 291
finish unbroken and last, 144
finish unbroken and last pre, 291
finish unbroken pre, 291
fit, 275
fit algorithm, 281
fit basedim, 276
fit fontsize macros, 277
fit height from, 280
fit height plus, 278
fit maxfontdiff, 282
fit maxfontdiffgap, 282
fit maxstep, 282
fit maxwidthdiff, 282
fit maxwidthdiffgap, 282
fit skip, 276
fit to, 276
fit to height, 276
fit warning, 282
fit width from, 279
fit width plus, 278
float, 60
float*, 60
floatplacement, 60
flushleft lower, 49
flushleft title, 49
flushleft upper, 49
flushright lower, 50
flushright title, 50
flushright upper, 50
fontlower, 22
fonttitle, 22
fontupper, 22
frame code, 93
frame code app, 291
frame code pre, 291
frame empty, 93
frame engine, 90
frame hidden, 102
frame style, 101
frame style image, 101
frame style tile, 102
freelance, 189
fuzzy halo, 132
fuzzy shadow, 130
geometry nodes, 92
graphical environment, 89
grow to left by, 69
grow to right by, 69
halo, 132
hbox, 76
hbox boxed title, 112
height, 39
height fill, 41
height fixed for, 260
height from, 40
height plus, 39
highlight math, 240
highlight math style, 240
hyphenationfix, 78
IfBooleanTF, 294
IfNoValueTF, 293
IfValueTF, 294
image comment, 212
index actual, 316
index annotate, 316
index colorize, 316
index default settings, 316
index format, 316
index german settings, 316
index level, 316
index quote, 316
interior code, 94
interior code app, 292
interior code pre, 292
interior empty, 94
interior engine, 91
interior hidden, 103
interior style, 102
interior style image, 103
interior style tile, 103
interior titled code, 93
interior titled code app, 291
interior titled code pre, 292
interior titled empty, 93
interior titled engine, 90
label, 79
label type, 79
left, 26
lefthand ratio, 64
lefthand width, 63
leftlower, 26
leftrule, 23
lefttitle, 26
leftupper, 26
lifted shadow, 136
lines before break, 256
list entry, 80
list text, 80
listing above comment, 217
listing above text, 216
listing and comment, 213
listing and text, 211
listing engine, 211
listing file, 211
listing inputencoding, 207
listing only, 211
listing options, 206
listing outside comment, 215
listing outside text, 214
listing remove caption, 207

listing side comment, 215
listing side text, 214
listing style, 206
listing utf8, 208
lower separated, 18
lowerbox, 17
marker, 159
math, 240
math lower, 240
math upper, 240
middle, 29
minimum for equal height group, 45
minipage, 76
minipage boxed title, 112
minipage boxed title*, 112
minted language, 209
minted options, 209
minted style, 210
natural height, 39
no borderline, 128
no finish, 144
no finish first, 144
no finish last, 144
no finish middle, 144
no finish unbroken, 144
no label type, 79
no listing options, 206
no overlay, 56
no shadow, 129
no underlay, 141
no underlay boxed title, 142
no underlay first, 142
no underlay last, 142
no underlay middle, 142
no underlay unbroken, 142
no watermark, 116
nobeforeafter, 65
nofloat, 60
noparskip, 65
nophantom, 79
notitle, 14
notitle after break, 256
on line, 78
opacityback, 37
opacitybacktitle, 37
opacityfill, 37
opacityframe, 37
opacitylower, 38
opacitytext, 38
opacitytitle, 38
opacityupper, 38
outer arc, 25
overlay, 55
overlay app, 286
overlay broken, 56
overlay broken app, 287
overlay broken pre, 287
overlay first, 56
overlay first and middle, 56
overlay first and middle app, 287
overlay first and middle pre, 287
overlay first app, 286
overlay first pre, 286
overlay last, 56
overlay last app, 287
overlay last pre, 287
overlay middle, 56
overlay middle and last, 56
overlay middle and last app, 287
overlay middle and last pre, 287
overlay middle app, 287
overlay middle pre, 287
overlay pre, 286
overlay unbroken, 56
overlay unbroken and first, 56
overlay unbroken and first app, 287
overlay unbroken and first pre, 287
overlay unbroken and last, 56
overlay unbroken and last app, 287
overlay unbroken and last pre, 287
overlay unbroken app, 286
overlay unbroken pre, 286
oversize, 31
pad after break, 259
pad at break, 259
pad at break*, 260
pad before break, 259
pad before break*, 259
parbox, 77
parskip, 65
phantom, 79
remake, 81
remember, 139
remember as, 140
reset, 81
right, 27
righthand ratio, 64
righthand width, 63
rightlower, 28
rightrule, 24
righttitle, 27
rightupper, 27
rotate, 139
rounded corners, 35
savedelimiter, 19
savelowerto, 17
scale, 139
segmentation code, 94
segmentation code app, 292
segmentation code pre, 292
segmentation empty, 94
segmentation engine, 91
segmentation hidden, 104
segmentation style, 104
separator sign, 235
separator sign colon, 235
separator sign dash, 235
separator sign none, 235

shadow, 129
sharp corners, 34
sharpish corners, 35
shield externalize, 81
show bounding box, 128
shrink break goal, 258
shrink tight, 71
sidebyside, 62
sidebyside align, 62
sidebyside gap, 63
size, 30
skin, 89
skin first, 89
skin first is subskin of, 96
skin last, 89
skin last is subskin of, 96
skin middle, 89
skin middle is subskin of, 96
space, 42
space to both, 42
space to lower, 42
space to upper, 42
spartan, 186
split, 43
standard, 148
standard jigsaw, 149
step, 79
step and label, 79
subtitle style, 16
tabularx, 51
tabularx*, 51
tcbox raise, 78
tcbox raise base, 78
tempfile, 78
terminator sign, 237
terminator sign colon, 238
terminator sign dash, 238
terminator sign none, 238
text above listing, 216
text and listing, 211
text only, 212
text outside listing, 214
text side listing, 214
theorem, 239
theorem name, 239
theorem name and number, 238
theorem number and name, 239
theorem style, 245
tikz, 138
tikz lower, 52
tikz reset, 138
tikz upper, 52
tikznode, 53
tikznode boxed title, 113
tikznode lower, 53
tikznode upper, 53
title, 14
title after break, 256
title code, 95
title code app, 292
title code pre, 292
title empty, 95
title engine, 91
title filled, 20
title hidden, 105
title style, 104
title style image, 105
title style tile, 105
titlerule, 24
toggle enlargement, 70
toggle left and right, 32
top, 28
toprule, 23
toprule at break, 259
topsep at break, 259
toptitle, 28
unbreakable, 256
underlay, 141
underlay boxed title, 142
underlay boxed title pre, 290
underlay broken, 142
underlay broken pre, 290
underlay first, 142
underlay first and middle, 142
underlay first and middle pre, 290
underlay first pre, 290
underlay last, 142
underlay last pre, 290
underlay middle, 142
underlay middle and last, 142
underlay middle and last pre, 290
underlay middle pre, 290
underlay pre, 290
underlay unbroken, 142
underlay unbroken and first, 142
underlay unbroken and first pre, 290
underlay unbroken and last, 142
underlay unbroken and last pre, 290
underlay unbroken pre, 290
valign, 43
valign lower, 43
varwidth boxed title, 113
varwidth boxed title*, 113
varwidth upper, 53
verbatim, 293
vfill before first, 260
watermark color, 119
watermark graphics, 115
watermark graphics app, 289
watermark graphics app on, 289
watermark graphics on, 115
watermark graphics pre, 289
watermark graphics pre on, 289
watermark opacity, 117
watermark overzoom, 118
watermark shrink, 118
watermark stretch, 119

watermark text, 114
watermark text app, 288
watermark text app on, 288
watermark text on, 114
watermark text pre, 288
watermark text pre on, 288
watermark tikz, 116
watermark tikz app, 289
watermark tikz app on, 289
watermark tikz on, 116
watermark tikz pre, 289
watermark tikz pre on, 289
watermark zoom, 117
widget, 173
width, 23
/tcb/boxtitle/
xshift, 108
yshift, 108
yshift*, 108
yshifttext, 108
/tcb/doclang/
color, 317
colors, 317
environment, 317
environment content, 317
environments, 317
index, 317
key, 317
keys, 317
new, 317
pageshort, 317
updated, 317
value, 317
values, 317
/tcb/library/
all, 8
breakable, 7
documentation, 7
fitting, 7
hooks, 7
listings, 7
listingsutf8, 7
many, 8
minted, 7
most, 8
skins, 7
theorems, 7
xparse, 7
/tcb/new/
auto counter, 82
blend into, 86
Crefname, 85
crefname, 85
list inside, 87
list type, 87
no counter, 83
number format, 84
number freestyle, 84
number within, 84
use counter, 83
use counter from, 83
use counter*, 83
/tikz/
fill image opacity, 196
fill image options, 196
fill image scale, 196
fill overzoom image, 192
fill overzoom image*, 192
fill overzoom picture, 192
fill plain image, 190
fill plain image*, 190
fill plain picture, 190
fill shrink image, 194
fill shrink image*, 194
fill shrink picture, 194
fill stretch image, 191
fill stretch image*, 191
fill stretch picture, 191
fill tile image, 195
fill tile image*, 195
fill tile picture, 195
fill tile picture*, 195
fill zoom image, 193
fill zoom image*, 193
fill zoom picture, 193
tcb fill frame, 106
tcb fill interior, 106
tcb fill title, 106
keys key, 317

label key, 79
label type key, 79
last value, 114–116, 260
left key, 26
lefthand ratio key, 64
lefthand width key, 63
leftlower key, 26
leftrule key, 23
lefttitle key, 26
leftupper key, 26
lifted shadow key, 136
lines before break key, 256
list entry key, 80
list inside key, 87
list text key, 80
list type key, 87
listing above comment key, 217
listing above text key, 216
listing and comment key, 213
listing and text key, 211
listing engine key, 211
listing file key, 211
listing inputencoding key, 207
listing only key, 211
listing options key, 206
listing outside comment key, 215
listing outside text key, 214
listing remove caption key, 207
listing side comment key, 215

listing side text key, 214
listing style key, 206
listing utf8 key, 208
listings key, 7
listings value, 86, 211
listingsutf8 key, 7
lower separated key, 18
lowerbox key, 17

many key, 8
\marg, 309
margin value, 247
margin apart value, 247
margin break value, 247
marker key, 159
math key, 240
math lower key, 240
math upper key, 240
\meta, 309
middle key, 29
middle value, 114–116, 260
middle and last value, 114–116, 260
minimal value, 30
minimum for equal height group key, 45
minipage key, 76
minipage value, 76
minipage boxed title key, 112
minipage boxed title* key, 112
minted key, 7
minted value, 211
minted language key, 209
minted options key, 209
minted style key, 210
most key, 8

natural height key, 39
new key, 317
\NewTCBInputListing, 303
\newtcbinputlisting, 205
\NewTCBListing, 301
\newtcblisting, 203
\NewTCBox, 298
\newtcbox, 13
\NewTCBoxFit, 304
\newtcboxfit, 274
\newtcbtheorem, 231
\NewTColorBox, 296
\newtcolorbox, 12
\NewTotalTCBox, 300
\NewTotalTCBoxFit, 305
\NewTotalTColorBox, 297
no borderline key, 128
no counter key, 83
no finish key, 144
no finish first key, 144
no finish last key, 144
no finish middle key, 144
no finish unbroken key, 144
no label type key, 79
no listing options key, 206
no overlay key, 56
no shadow key, 129
no underlay key, 141
no underlay boxed title key, 142
no underlay first key, 142
no underlay last key, 142
no underlay middle key, 142
no underlay unbroken key, 142
no watermark key, 116
nobeforeafter key, 65
nofloat key, 60
none value, 32, 70, 260
noparskip key, 65
nophantom key, 79
normal value, 30
north value, 34, 35
northeast value, 34, 35
northwest value, 34, 35
notitle key, 14
notitle after break key, 256
number format key, 84
number freestyle key, 84
number within key, 84

\oarg, 309
off value, 282, 316
on value, 282
on line key, 78
opacityback key, 37
opacitybacktitle key, 37
opacityfill key, 37
opacityframe key, 37
opacitylower key, 38
opacitytext key, 38
opacitytitle key, 38
opacityupper key, 38
Option color, 320
outer arc key, 25
overlay key, 55
overlay app key, 286
overlay broken key, 56
overlay broken app key, 287
overlay broken pre key, 287
overlay first key, 56
overlay first and middle key, 56
overlay first and middle app key, 287
overlay first and middle pre key, 287
overlay first app key, 286
overlay first pre key, 286
overlay last key, 56
overlay last app key, 287
overlay last pre key, 287
overlay middle key, 56
overlay middle and last key, 56
overlay middle and last app key, 287
overlay middle and last pre key, 287
overlay middle app key, 287
overlay middle pre key, 287
overlay pre key, 286
overlay unbroken key, 56

overlay unbroken and first key, 56
overlay unbroken and first app key, 287
overlay unbroken and first pre key, 287
overlay unbroken and last key, 56
overlay unbroken and last app key, 287
overlay unbroken and last pre key, 287
overlay unbroken app key, 286
overlay unbroken pre key, 286
oversize key, 31

pad after break key, 259
pad at break key, 259
pad at break* key, 260
pad before break key, 259
pad before break* key, 259
pageshort key, 317
parbox key, 77
parskip key, 65
path value, 90, 91
pathfirst value, 90, 91
pathfirstjigsaw value, 90
pathjigsaw value, 90
pathlast value, 90, 91
pathlastjigsaw value, 90
pathmiddle value, 90, 91
pathmiddlejigsaw value, 90
pgf value, 316
pgfchapter value, 316
pgfsection value, 316
phantom key, 79
plain value, 245
plain apart value, 246
\ProvideTCBInputListing, 303
\ProvideTCBListing, 301
\ProvideTCBox, 298
\ProvideTCBoxFit, 304
\ProvideTColorBox, 296
\ProvideTotalTCBox, 300
\ProvideTotalTCBoxFit, 305
\ProvideTotalTColorBox, 297

\refAux, 313
\refAuxcs, 313
\refCom, 312
\refCom*, 312
\refEnv, 312
\refEnv*, 312
\refKey, 313
\refKey*, 313
remake key, 81
remember key, 139
remember as key, 140
\RenewTCBInputListing, 303
\renewtcbinputlisting, 205
\RenewTCBListing, 301
\renewtcblisting, 204
\RenewTCBox, 298
\renewtcbox, 13
\RenewTCBoxFit, 304
\renewtcboxfit, 274
\renewtcbtheorem, 232
\RenewTColorBox, 296
\renewtcolorbox, 12
\RenewTotalTCBox, 300
\RenewTotalTCBoxFit, 305
\RenewTotalTColorBox, 297
reset key, 81
right key, 27
righthand ratio key, 64
righthand width key, 63
rightlower key, 28
rightrule key, 24
righttitle key, 27
rightupper key, 27
rotate key, 139
rounded corners key, 35

savedelimiter key, 19
savelowerto key, 17
scale key, 139
segmentation code key, 94
segmentation code app key, 292
segmentation code pre key, 292
segmentation empty key, 94
segmentation engine key, 91
segmentation hidden key, 104
segmentation style key, 104
separator sign key, 235
separator sign colon key, 235
separator sign dash key, 235
separator sign none key, 235
shadow key, 129
sharp corners key, 34
sharpish corners key, 35
shield externalize key, 81
show bounding box key, 128
shrink break goal key, 258
shrink tight key, 71
sidebyside key, 62
sidebyside align key, 62
sidebyside gap key, 63
size key, 30
skin key, 89
skin first key, 89
skin first is subskin of key, 96
skin last key, 89
skin last is subskin of key, 96
skin middle key, 89
skin middle is subskin of key, 96
Skins
    beamer, 168
    beamerfirst, 170
    beamerlast, 172
    beamermiddle, 171
    bicolor, 163
    bicolorfirst, 165
    bicolorlast, 167
    bicolormiddle, 166
    draft, 187
    empty, 177

emptyfirst, 179
emptylast, 181
emptymiddle, 180
enhanced, 150
enhanced jigsaw, 157
enhancedfirst, 154
enhancedfirst jigsaw, 158
enhancedlast, 156
enhancedlast jigsaw, 162
enhancedmiddle, 155
enhancedmiddle jigsaw, 159
freelance, 189
freelancefirst, 189
freelancelast, 189
freelancemiddle, 189
spartan, 186
standard, 148
standard jigsaw, 149
widget, 173
widgetfirst, 174
widgetlast, 176
widgetmiddle, 175
skins key, 7
small value, 30
south value, 34, 35
southeast value, 34, 35
southwest value, 34, 35
space key, 42
space to both key, 42
space to lower key, 42
space to upper key, 42
spartan key, 186
spartan Skin, 186
spartan value, 90, 91
split key, 43
squeeze value, 281
standard key, 148
standard Skin, 148
standard value, 90, 91, 245
standard jigsaw key, 149
standard jigsaw Skin, 149
step key, 79
step and label key, 79
subtitle style key, 16

tables value, 86
tabularx key, 51
tabularx* key, 51
tcb fill frame key, 106
tcb fill interior key, 106
tcb fill title key, 106
tcbclipframe environment, 121
tcbclipinterior environment, 123
tcbcliptitle environment, 123
\tcbcontinuedraftmode, 147
\tcbcounter, 82
\tcbdocmarginnote, 314
\tcbdocnew, 314
\tcbdocupdated, 314
\tcbfitdim, 274–277
\tcbfitsteps, 275
\tcbfontsize, 274
\tcbhighmath, 233
\tcbinputlisting, 202
\tcbinterruptdraftmode, 147
tcbinvclipframe environment, 122
\tcbline, 153
\tcbline*, 153
tcblisting environment, 200
\tcblistof, 87
\tcblower, 9
\tcbmakedocSubKey, 312
\tcbmaketheorem, 232
tcboutputlisting environment, 202
\tcbox, 11
tcbox raise key, 78
tcbox raise base key, 78
\tcboxedtitleheight, 108
\tcboxedtitlewidth, 108
\tcboxfit, 273
\tcboxmath, 233
\tcboxverb, 300
\tcbset, 10
\tcbsetforeverylayer, 10
\tcbsetmacrotoheightofnode, 198
\tcbsetmacrotowidthofnode, 198
\tcbsetmanagedlayer, 74
\tcbsettoheightofnode, 198
\tcbsettowidthofnode, 198
\tcbstartdraftmode, 147
\tcbstopdraftmode, 147
\tcbsubskin, 96
\tcbsubtitle, 16
\tcbtitle, 15
\tcbtitletext, 15
\tcbuselibrary, 7
\tcbuselistinglisting, 202
\tcbuselistingtext, 202
\tcbusetemp, 88
\tcbusetemplisting, 202
tcbverbatimwrite environment, 88
tcbwritetemp environment, 88
tcolorbox environment, 9
tempfile key, 78
terminator sign key, 237
terminator sign colon key, 238
terminator sign dash key, 238
terminator sign none key, 238
text above listing key, 216
text and listing key, 211
text only key, 212
text outside listing key, 214
text side listing key, 214
theorem key, 239
theorem name key, 239
theorem name and number key, 238
theorem number and name key, 239
theorem style key, 245
theorems key, 7

\thetcbcounter, 82, 83
tight value, 30
tikz key, 138
tikz lower key, 52
tikz reset key, 138
tikz upper key, 52
tikznode key, 53
tikznode boxed title key, 113
tikznode lower key, 53
tikznode upper key, 53
title key, 14
title value, 30
title after break key, 256
title code key, 95
title code app key, 292
title code pre key, 292
title empty key, 95
title engine key, 91
title filled key, 20
title hidden key, 105
title style key, 104
title style image key, 105
title style tile key, 105
titlerule key, 24
toggle enlargement key, 70
toggle left and right key, 32
top key, 28
top value, 62
toprule key, 23
toprule at break key, 259
topsep at break key, 259
toptitle key, 28
true value, 255

unbreakable key, 256
unbroken value, 114–116
unbroken and first value, 114–116
underlay key, 141
underlay boxed title key, 142
underlay boxed title pre key, 290
underlay broken key, 142
underlay broken pre key, 290
underlay first key, 142
underlay first and middle key, 142
underlay first and middle pre key, 290
underlay first pre key, 290
underlay last key, 142
underlay last pre key, 290
underlay middle key, 142
underlay middle and last key, 142
underlay middle and last pre key, 290
underlay middle pre key, 290
underlay pre key, 290
underlay unbroken key, 142
underlay unbroken and first key, 142
underlay unbroken and first pre key, 290
underlay unbroken and last key, 142
underlay unbroken and last pre key, 290
underlay unbroken pre key, 290
unlimited value, 254, 255
updated key, 317
uphill value, 34, 35
use counter key, 83
use counter from key, 83
use counter* key, 83

valign key, 43
valign lower key, 43
value key, 317
Values
    all, 34, 35, 260
    areasize, 281
    bottom, 62
    break, 246
    broken, 114–116
    center, 62
    change, 246
    change apart, 246
    change break, 246
    change standard, 245
    doc, 316
    downhill, 34, 35
    east, 34, 35
    empty, 90, 91
    evenpage, 32, 70
    false, 255
    fbox, 30
    figures, 86
    final, 282
    first, 114–116, 260
    first and middle, 114, 260
    fitbox, 76
    fontsize, 281
    foovalue, 308
    forced, 32, 70
    freelance, 90, 91
    hbox, 76
    hybrid, 281
    ignored, 17
    invisible, 17
    last, 114–116, 260
    listings, 86, 211
    margin, 247
    margin apart, 247
    margin break, 247
    middle, 114–116, 260
    middle and last, 114–116, 260
    minimal, 30
    minipage, 76
    minted, 211
    none, 32, 70, 260
    normal, 30
    north, 34, 35
    northeast, 34, 35
    northwest, 34, 35
    off, 282, 316
    on, 282
    path, 90, 91
    pathfirst, 90, 91
    pathfirstjigsaw, 90

pathjigsaw, 90
pathlast, 90, 91
pathlastjigsaw, 90
pathmiddle, 90, 91
pathmiddlejigsaw, 90
pgf, 316
pgfchapter, 316
pgfsection, 316
plain, 245
plain apart, 246
small, 30
south, 34, 35
southeast, 34, 35
southwest, 34, 35
spartan, 90, 91
squeeze, 281
standard, 90, 91, 245
tables, 86
tight, 30
title, 30
top, 62
true, 255
unbroken, 114–116
unbroken and first, 114–116
unlimited, 254, 255
uphill, 34, 35
visible, 17
west, 34, 35
values key, 317
varwidth boxed title key, 113
varwidth boxed title* key, 113
varwidth upper key, 53
verbatim key, 293
vfill before first key, 260
visible value, 17

watermark color key, 119
watermark graphics key, 115
watermark graphics app key, 289
watermark graphics app on key, 289
watermark graphics on key, 115
watermark graphics pre key, 289
watermark graphics pre on key, 289
watermark opacity key, 117
watermark overzoom key, 118
watermark shrink key, 118
watermark stretch key, 119
watermark text key, 114
watermark text app key, 288
watermark text app on key, 288
watermark text on key, 114
watermark text pre key, 288
watermark text pre on key, 288
watermark tikz key, 116
watermark tikz app key, 289
watermark tikz app on key, 289
watermark tikz on key, 116
watermark tikz pre key, 289
watermark tikz pre on key, 289
watermark zoom key, 117
west value, 34, 35
widget key, 173
widget Skin, 173
widgetfirst Skin, 174
widgetlast Skin, 176
widgetmiddle Skin, 175
width key, 23

xparse key, 7
xshift key, 108

yshift key, 108
yshift* key, 108
yshifttext key, 108