The `tcolorbox` package

Manual for version 1.30 (2012/04/20)

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

1 Introduction

The package originates from the first edition of my book «*LaTeX – Einführung in das Textsatzsystem*» [8] 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 [5–7] as well. With this package, you are invited to apply these boxes for similar projects.

1.1 Loading the Package

The base package `tcolorbox` loads the packages `pgf` [9], `calc` [10], and `verbatim` [4]. `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. 3 (see there). For example, the key to typeset listings is:

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

1.2 Libraries

The base package `tcolorbox` is extendable by program libraries. This is done by usage of 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/listings` (no value)
Loads the package `listings` [2] and provides additional macros for typesetting listings which are described in section 5 from page 25.

`/tcb/library/theorems` (no value)
Provides additional macros for typesetting theorems which are described in section 6 from page 38.

`/tcb/library/documentation` (no value)
Provides additional macros for typesetting LaTeX documentations which are described in section 7 from page 42.

`/tcb/library/skins` (no value)
Loads the package `tikz` [9] and provides additional styles (skins) for the appearance of the colored boxes; see section 8 from page 48.

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` → P. 4. 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 for the complete list. A quick example is given here:

```
\begin{tcolorbox}[colback=red!5,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` → P. 4 to separate the upper box part from the optional lower box part.

`\tcbset{`⟨*options*⟩`}`

Sets options for every following `tcolorbox` → P. 4 inside the current TEX group. For example, the colors of the boxes may be defined for the whole document by this:

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

3 Option Keys

For the ⟨*options*⟩ in `tcolorbox` → P. 4 respectively `\tcbset` → P. 4 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 [8, 9] or the examples starting from page 30.

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 'Äpgjy'. 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.

3.2 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]
This is a \textbf{tcolorbox}.
\tcblower
This is the lower part (but invisible).
\end{tcolorbox}

\begin{tcolorbox}[lowerbox=ignored]
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 box in a file for an optional later usage.

```
\begin{tcolorbox}[lowerbox=invisible,savelowerto=\jobname_bspsave.tex]
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: $f(x) = \frac{1+x^2}{1-x^2}$.

/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,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!

3.3 Colors and Fonts

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

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

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

This is a **tcolorbox**.

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

Sets the frame ⟨*color*⟩ of the box.

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

This is a **tcolorbox**.

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

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

```
\begin{tcolorbox}[colupper=yellow,colback=blue!50,colframe=blue]
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=yellow,colback=blue!50,colframe=blue]
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=yellow,colback=blue!50,colframe=blue]
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=blue!50!black,colframe=blue!25,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**.

3.4 Geometry

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

Sets the total width of the colored box to ⟨*length*⟩. See also **/tcb/height** → P. 14.

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

This is a **tcolorbox**.

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

Sets the frame width to ⟨*length*⟩.

```
\begin{tcolorbox}[boxrule=2mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

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

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

```
\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/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.

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

This is a **tcolorbox**.

/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.

```
\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`).

```
\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`).

```
\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`).

```
\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.

```
\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`).

```
\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`).

```
\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).

```
\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).

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

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

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

This is a **tcolorbox**.

This is the lower part.

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

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

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

3.5 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.

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

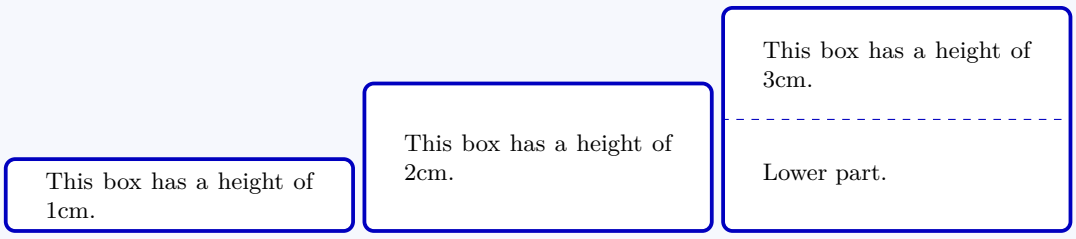
Sets the total height of the colored box to its natural height depending on the box content.

/tcb/height=⟨*length*⟩ (no default)

Sets the total height of the colored box to ⟨*length*⟩ independent of the box content.

```
\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}
```

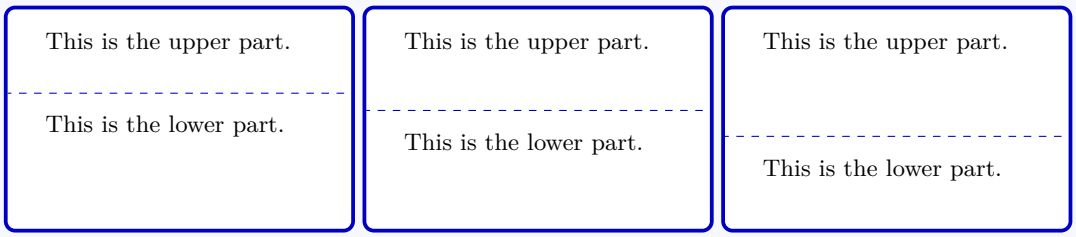


/tcb/space=⟨*fraction*⟩ (no default, initially 0)

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)

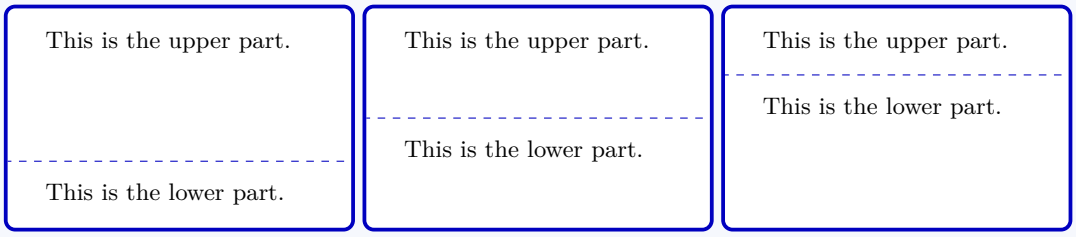
This is an abbreviation for `space=0`, i. e. all extra space is added to the lower part (if there is any).

/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}}
```

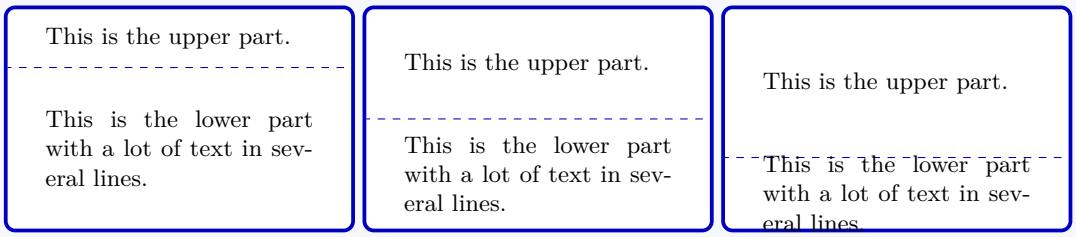


/tcb/split=⟨*fraction*⟩ (no default)

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}}
```

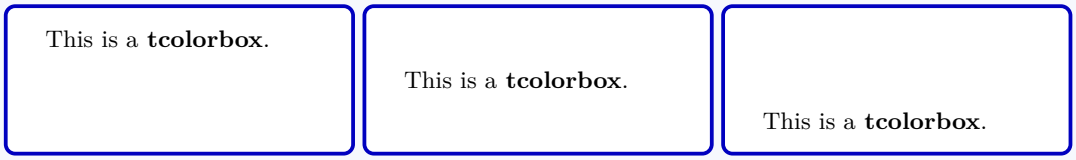


/tcb/valign=⟨*alignment*⟩ (no default, initially `top`)

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. 14 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}%
```

My first box. All boxes will get 3.5cm times 3.5cm if the content height is not too large.

My second box.

This is the lower part.

```
\textbf{Mixed}
with a listing.
```

Mixed with a listing.

Fourth box

My final box.

Floating box from `floatplacement`

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

3.6 Floating Objects

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

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

```
\begin{tcolorbox}[floatplacement=t,float,
                  title=Floating box from |floatplacement|]
  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|]
  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.

3.7 Embedding into the Surroundings

`/tcb/before`=⟨*macros*⟩ (no default, initially `\par\pagebreak[0]\noindent`)

Sets the ⟨*macros*⟩ which are executed before the colored box. They are not used for floating boxes.

`/tcb/after`=⟨*macros*⟩ (no default, initially `\par`)

Sets the ⟨*macros*⟩ which are executed after the colored box. They are not used for floating boxes.

`/tcb/parskip` (style, no value)

Sets the keys `before` and `after` to their default values. This is recommended, if the package `parskip` is used and there is no better idea for `before` and `after`.

`/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]\noindent},
                          after={\par\smallskip}}}
```

3.8 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.

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

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

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

\begin{tcolorbox}[enlarge top by=-5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge top by=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

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

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

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

\begin{tcolorbox}[enlarge bottom by=5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[enlarge bottom by=-5mm]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

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

This is a **tcolorbox**.

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

This is a **tcolorbox**.

This is a **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*⟩ (no default, 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. Note that the page number detection may fail in certain cases. For these cases, use the `forced` value.

```
\tcbset{colframe=blue!75!black,colback=white,
enlarge left by=-20mm,enlarge right by=5mm,width=\linewidth+15mm}

\begin{tcolorbox}[toggle enlargement=none]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=forced]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
\begin{tcolorbox}[toggle enlargement=evenpage]
  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).
\end{tcolorbox}
```

This is a **tcolorbox**.

This is a **tcolorbox**.

This page is an even page. Therefore, the left and right enlargements are toggled (with some luck).

```
\begin{tcolorbox}[colframe=red!60!black,colback=red!15!white,
  fonttitle=\bfseries,title=Floating box from \texttt{toggle enlargement},
  width=\textwidth+20mm,enlarge right by=-20mm,
  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}
```

3.9 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. 24 and `\tcbusetemp` → P. 24 implicitely.

3.10 Skins

There are additional option keys which change the appearance of a `tcolorbox`. If only the core package is used, there is only one *skin* and these keys are meaningless. The library 'skins' adds more skins. The appropriate option keys for skins of the core package are therefore described in section 8.1 from page 48.

4 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` → P. 24, 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` → P. 24.

5 Library 'listings'

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{listings}
```

This also loads the package `listings` [2].

5.1 Macros of the Library

`\begin{tcblisting}{`⟨*options*⟩`}`
⟨*environment content*⟩
`\end{tcblisting}`

Creates a colored box based on a `tcolorbox` → P. 4. Controlled by the given ⟨*options*⟩, the environment content is typesetted 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,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.

```
\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>
```

\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. 25.

```
\tcbinputlisting{colback=red!5,colframe=red!75!black,text only}
\tcbinputlisting{colback=green!5,colframe=green!75!black,listing only}
```

This **text** is written to a standardized file for later usage.

```
This \textbf{text} is written to a standardized file for later usage.
```

\tcbuselistingtext

Loads text from a file named by the key value of `listing file`.

```
\tcbuselistingtext
```

This **text** is written to a standardized file for later usage.

\tcbuselistinglisting

Typesets text as listing from a file named by the key value of `listing file`.

```
\tcbuselistinglisting
```

```
This \textbf{text} is written to a standardized file for later usage.
```

\tcbusetemplisting

Typesets text as listing from a temporary file which was written by `tcbwritetemp` → P. 24.

5.2 Option Keys of the Library

For the ⟨*options*⟩ in `tcblisting` → P. 25 respectively `\tcbinputlisting` → P. 26 the following `pgf` keys can be applied. The key tree path `/tcb/` is not to be used inside these macros.

`/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 options`=⟨*key list*⟩ (no default, initially `style=tcblatex`)
Sets the options from the package `listings` [2] 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,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/listing style`=⟨*style*⟩ (no default, initially `tcblatex`)
Abbreviation for `listing options={style=...}`. This key sets a ⟨*style*⟩ for the `listings` package, see [2]. For LaTeX, there is a predefined style named `tcblatex`.

```
\begin{tcblisting}{colback=red!5,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 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,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,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,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,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 only in conjunction with /tcb/listing and comment → P. 29 and /tcb/comment and listing → P. 29.

```
\begin{tcblisting}{comment={This comment is really only a comment},
  colback=red!5,colframe=red!75!black}
This is a \textbf{tcolorbox}.
\end{tcblisting}
```

This is a \textbf{tcolorbox}.

This is a **tcolorbox**.

/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,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,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.

5.3 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 and automatic numbering.

```
\newcounter{texexp}

\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/.code 2 args={\refstepcounter{texexp}\label{#2}%
      \pgfkeysalso{texexp,title={Example \thetexexp: #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{tcblisting}{example={Direct application of \texttt{example}}{firstExample}}
Here, we use Example \ref{firstExample} with a title line.
\end{tcblisting}
```

Example 1: Direct application of `example`

Here, we use Example \ref{firstExample} with a title line.

Here, we use Example 1 with a title line.

For the next examples, two environments `texexp` and `texexptitled` are defined for abbreviation.

```
\newenvironment{texexp}[1]{\tcblisting{texexp,#1}}{\endtcblisting}

\newenvironment{texexptitled}[3][]{\tcblisting{example={#2}{#3},#1}}{\endtcblisting}
```

For special cases, `tcolorbox` with the key `texexp` can be used instead of these two environments.

Now, we have a flexible box for LaTeX examples which is demonstrated in the following.

```
\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 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 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 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 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 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 4 on page 32.

```
\begin{tcolorbox}[example={Direct application of the style |texexp|}{texexpbox1}]
\begin{lstlisting}[style=tcblatex]
Some \LaTeX\ source code.
\end{lstlisting}
\tcblower
For special cases, the environment |tcolorbox| 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{tcolorbox}
```

Example 5: Direct application of the style `texexp`

```
Some \LaTeX\ source code.
```

For special cases, the environment `tcolorbox` 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.

5.4 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}

\newwrite\solout
\def\openoutsol{\immediate\openout\solout\jobname.sol}
\def\solfile#1{solutions/texercise#1.tex}
\def\writesol#1{\immediate\write\solout{\noexpand\processsol{\thetexercise}{#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. 26 environment. Note, that you can use this content also inside the exercise with `\tcbuselistingtext` → P. 26 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 excercise 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. 24 and used in compiled form by `\tcbusetemp` → P. 24 or as source code by `\tcbusetemplisting` → P. 26.

At first, we generate some new keys. Since exercises und solutions should be numbered, we force to use a label ⟨*marker*⟩ while using the style `texercise`. 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},
texercise/.code={\refstepcounter{texercise}\label{exe:#1}\writesol{#1}%
  \pgfkeysalso{texercisestyle,
    listing file={\solfile\thetexercise},
    title={Exercise \arabic{texercise}\hfill\mdseries Solution on page \pageref{sol:#1}}
  }}
}
```

With these preparations, the kernel environment `texercise` for our exercises is created quickly:

```
\newenvironment{texercise}[2][]{%
  \tcolorbox[texercise=#2,savedelimiter=texercise,#1]}%
  {\endtcolorbox}
```

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 1 Solution on page 37

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 2 Solution on page 37

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 3 Solution on page 37

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 4 Solution on page 37

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.

5.5 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.

```
\newcommand{\processsol}[2]{%
  \tcbinputlisting{texercisestyle,listing only,
    title={Solution for Exercise \ref{exe:#2} on page \pageref{exe:#2}\label{sol:#2}},
    listing file={\solfile#1}%
  }}
```

The loading of all solutions is done by:

```
\inputsol
```

With this, we get:

Solution for Exercise 1 on page 34

```
\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 2 on page 35

```
\newcommand{\headingline}[1]{%
  \begin{center}\Large\bfseries #1\end{center}}
```

Solution for Exercise 3 on page 35

```
\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 4 on page 36

```
\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}}
```

6 Library 'theorems'

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{theorems}
```

6.1 Macros of the Library

\tcbmaketheorem{⟨*name*⟩}{⟨*display name*⟩}{⟨*options*⟩}{⟨*counter*⟩}{⟨*prefix*⟩}

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*⟩.

```
\tcbmaketheorem{theo}{My Theorem}{colback=green!5,colframe=green!35!black,
  fonttitle=\bfseries}{texercise}{th}

\begin{theo}{This is my title}{theoexample}
  This is the text of the theorem. As can be seen, the counter \texttt{texercise}
  is reused. The theorem is numbered with \ref{th:theoexample} and is
  given on page \pageref{th:theoexample}.
\end{theo}
```

My Theorem 5: This is my title

This is the text of the theorem. As can be seen, the counter `texercise` is reused. The theorem is numbered with 5 and is given on page 38.

6.2 Option Keys of the Library

/tcb/theorem={⟨*display name*⟩}{⟨*counter*⟩}{⟨*title*⟩}{⟨*marker*⟩} (no default)

This key is internally used by `\tcbmaketheorem` → P. 38, 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.

```
\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 6: Direct usage

Here, we see the test 6.

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. 38.

6.3 Examples for Definitions and Theorems

In the following, the application of \tcbmaketheorem → P. 38 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 section number.

```
\newcounter{mytheorem}[section]
\def\themytheorem{\thesection.\arabic{mytheorem}}

\tcbset{
  defstyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
              arc=0mm, colback=blue!5,colframe=blue!75!black},
  theostyle/.style={fonttitle=\bfseries\upshape, fontupper=\slshape,
               colback=red!10,colframe=red!75!black},
}
```

By \tcbmaketheorem → P. 38, commonly numbered theorem environments are created now. `defstyle` and `theostyle` are used for the appearance.

```
\tcbmaketheorem{Definition}{Definition}{defstyle}{mytheorem}{def}
\tcbmaketheorem{Theorem}{Theorem}{theostyle}{mytheorem}{theo}
\tcbmaketheorem{Corollary}{Corollary}{theostyle}{mytheorem}{cor}
```

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 6.1 and referenced with the marker `theo:diffbarstetig`.

Theorem 6.1: 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 6.2 and referenced with the marker `def:diffbarkeit`.

Definition 6.2: 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 6.3 and referenced with the marker `cor:nullstellen`.

Corollary 6.3: 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]{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 6.4: 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.

7 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` [9] 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 5, and a bunch of packages, namely `doc`, `pifont`, `marvosym`, `hyperref`, and `makeidx`.

7.1 Macros of the Library

`\begin{docCommand}{`⟨*name*⟩`}{`⟨*parameters*⟩`}`
⟨*environment content*⟩
`\end{docCommand}`

Documents a LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash. This macro takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with `\refCom`→ P. 45`{`⟨*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. 43 for the documentation of keys with the given ⟨*key path*⟩.

`\begin{docEnvironment}{`⟨*name*⟩`}{`⟨*parameters*⟩`}`
⟨*environment content*⟩
`\end{docEnvironment}`

Documents a LaTeX environment with given ⟨*name*⟩. This environment takes mandatory or optional ⟨*parameters*⟩. It is automatically indexed and can be referenced with `\refEnv`→ P. 45`{`⟨*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{docKey}[⟨*key path*⟩]{⟨*name*⟩}{⟨*parameters*⟩}{⟨*description*⟩}
⟨*environment content*⟩
\end{docKey}

Documents a key with given ⟨*name*⟩ and an optional ⟨*key path*⟩. This key takes mandatory or optional ⟨*parameters*⟩ as value with a short ⟨*description*⟩. It is automatically indexed and can be referenced with \refKey → P. 45{⟨*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.

\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.

\docColor{⟨*name*⟩}

Documents a color with given ⟨*name*⟩. The color is automatically indexed.

```
The color \docColor{foocolor} is available.
```

The color foocolor is available.

\cs{⟨*name*⟩}

Macro from ltxdoc [1] 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 [3] to typeset a meta ⟨*text*⟩.

```
This is a \meta{text}.
```

This is a ⟨*text*⟩.

\marg{⟨*text*⟩}

Macro from ltxdoc [1] 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` [1] 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}.

\begin{dispExample}
⟨*environment content*⟩
\end{dispExample}

Creates a colored box based on a `tcolorbox` → P. 4. 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. 46 and the style `/tcb/docexample` → P. 46. 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.

\begin{dispListing}
⟨*environment content*⟩
\end{dispListing}

Creates a colored box based on a `tcolorbox` → P. 4. It displays the environment content as source code. The appearance is controlled by `/tcb/documentation listing style` → P. 46 and the style `/tcb/docexample` → P. 46. It may be changed by redefining this style.

```
\begin{dispListing}
This is a \LaTeX\ example.
\end{dispListing}
```

```
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. 43 for the documentation of keys with the given ⟨*key path*⟩ as default. The new environment ⟨*name*⟩ takes the same parameters as docKey → P. 43 itself.

```
\tcbmakedocSubKey{docFooKey}{foo}

\begin{docFooKey}{foodummy}{=\meta{nothing}}{no default, initially empty}
Some key.
\end{docFooKey}
```

> /foo/foodummy=⟨*nothing*⟩ (no default, initially empty)
> Some key.

\refCom{⟨*name*⟩}

References a documented LaTeX macro with given ⟨*name*⟩ where ⟨*name*⟩ is written without backslash.

```
We have created \refCom{foomakedocSubKey} as an example.
```

> We have created \foomakedocSubKey → P. 42 as an example.

\refEnv{⟨*name*⟩}

References a documented LaTeX environment with given ⟨*name*⟩.

```
We have created \refEnv{foocolorbox} as an example.
```

> We have created foocolorbox → P. 42 as an example.

\refKey{⟨*name*⟩}

References a documented key with given ⟨*name*⟩ where ⟨*name*⟩ is the full path name of the key.

```
We have created \refKey{/foo/footitle} as an example.
```

> We have created /foo/footitle → P. 43 as an example.

`\colDef{`⟨*text*⟩`}`
Sets ⟨*text*⟩ with the definition color, see `/tcb/color definition` → P. 46.

```
This is my \colDef{text}.
```

This is my text.

`\colOpt{`⟨*text*⟩`}`
Sets ⟨*text*⟩ with the option color, see `/tcb/color option` → P. 46.

```
This is my \colOpt{text}.
```

This is my text.

7.2 Option Keys of the Library

`/tcb/docexample` (style, no value)
Sets the style for `dispExample` → P. 44 and `dispListing` → P. 44 with the colors `ExampleBack` and `ExampleFrame`. To change the appearance of the examples, this style could be redefined.

`/tcb/documentation listing style`=⟨*listing style*⟩ (no default, initially `tcbdocumentation`)
Sets a ⟨*listing style*⟩ for the `listings` package [2]. It is used inside `dispExample` → P. 44 and `dispListing` → P. 44 to typeset the listings. Note that this is not identical to the key `/tcb/listing style` → P. 27 which is used for 'normal' listings.

`/tcb/color definition`=⟨*color*⟩ (no default, initially `Definition`)
Sets the highlight color used by macro and key definitions.

`/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. 44 and `dispListing` → P. 44 additional to `/tcb/before` → P. 19.

`/tcb/after example`=⟨*macros*⟩ (no default, initially empty)
Sets the ⟨*macros*⟩ which are executed after `dispExample` → P. 44 and `dispListing` → P. 44 additional to `/tcb/after` → P. 19.

`/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` → P. 46, `/tcb/index quote` → P. 46, and `/tcb/index level` → P. 46.

/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={>}}
```

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 content=⟨*text*⟩ (no default, initially `environment content`)

Text used in `docEnvironment` → P. 42.

/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/key=⟨*text*⟩ (no default, initially `key`)

Text used in the index for keys.

/tcb/doclang/index=⟨*text*⟩ (no default, initially `Index`)

Heading text for the index.

/tcb/doclang/pageshort=⟨*text*⟩ (no default, initially `P.`)

Short text for page references.

7.3 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` .

8 Library 'skins'

The library is loaded by a package option or inside the preamble by:

```
\tcbuselibrary{skins}
```

This also loads the package `tikz` [9]. Typically but not necessarily, the following skins use `tikz` instead of `pgf`.

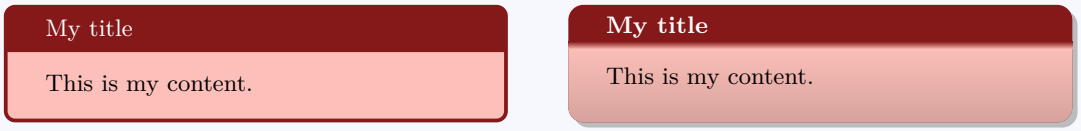
8.1 Technical Overview and Core Package Option Keys

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 a single skin called `standard` → P. 50. The 'skins' library adds several more skins. To change 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,adjusted title=My title]
  This is my content.
\end{tcolorbox}
```



/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` → P. 48 and may seldom be used directly.

The skin of a `tcolorbox` is drawn by up to three *engines*. Afterwards, the text content is drawn which is not part of a skin. The three steps are:

1. The *frame* of the box.
2. The *interior* of the box. The interior of a box with title is drawn differently from a box without title.
3. The *segmentation* (line) of the box, if there is a lower part.

Every engine for the up to three steps can be set to one of three types:

1. `standard`: the original code from the core package.
2. `path`: a `tikz` path which can be controlled by options.
3. `freelance`: arbitrary user code.

/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. 48. 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. 50,
- **freelance**: arbitrary user code which is given by /tcb/frame code → P. 52.

/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. 48. 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. 50,
- **freelance**: arbitrary user code which is given by /tcb/interior titled code → P. 52.

/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. 48. 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. 50,
- **freelance**: arbitrary user code which is given by /tcb/interior code → P. 52.

/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. 48. Feasible values for ⟨*name*⟩ are:

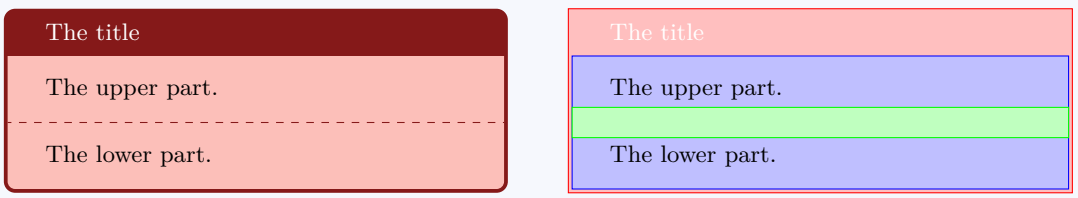
- **standard**: the original code from the core package,
- **path**: a **tikz** path which is controlled by /tcb/segmentation style → P. 50,
- **freelance**: arbitrary user code which is given by /tcb/segmentation code → P. 52.

/tcb/geometry nodes=⟨*boolean value*⟩ (default **true**, initially **false**)

If set to **true**, three **tikz** nodes are defined for a **tcolorbox** which are named **frame**, **interior**, and **segmentation**. These nodes describe the boundaries of the equally named parts of a **tcolorbox**. They are used by all engines of type **path** and they may be used by engines of type **freelance**. Typically, this key is set by a /tcb/skin → P. 48.

```
\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}[adjusted title=The title,skin=freelance,
  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);}]
  The upper part.\tcblower The lower part.
\end{tcolorbox}
```



8.2 Skin 'standard'

`/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.

8.3 Skin 'enhanced'

`/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. 50 skin. You pay for this with some prolonged compilation time. All drawing engines are set to type `path`; they use the `tikz` package and compute the `/tcb/geometry nodes` → P. 49. The `tikz` path options can be given with the following option keys.

`/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.

`/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.

`/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.

```
% needs shadings library
\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`.

```
% needs shadows and decorations.pathmorphing library
\tcbset{skin=enhanced,fonttitle=\bfseries,boxrule=1mm,
  frame style={draw=FireBrick,fill=Salmon,drop 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.

$$\int_1^2 \frac{1}{x}\, dx = \ln(2). \tag{1}$$

8.4 Skin 'freelance'

`/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. 49. This skin is useful for boxes which should differ much from the normal appearance. Note that this difference has to be programmed by the user. The drawing code can be given with the following option keys. As default value, the code from the `standard` skin is set.

`/tcb/frame code`=⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *frame* of the box.

`/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.

`/tcb/interior code`=⟨*graphical code*⟩ (code, default from `standard`)

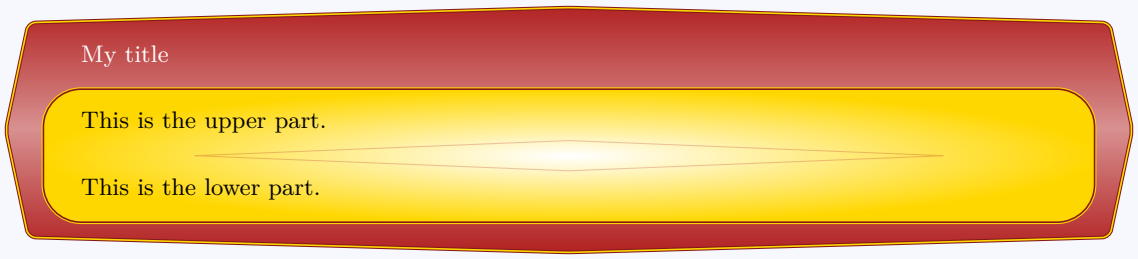
The given ⟨*graphical code*⟩ is used for drawing the *interior* of the box, if the box is without a title.

`/tcb/segmentation code`=⟨*graphical code*⟩ (code, default from `standard`)

The given ⟨*graphical code*⟩ is used for drawing the *segmentation* area of the box.

```
\tcbset{skin=freelance,boxrule=2mm,enlarge top by=2mm,enlarge bottom by=2mm,
  enlarge left by=3mm,enlarge right by=3mm,width=\linewidth-6mm,
frame code={\path[top color=FireBrick,bottom color=FireBrick,middle color=FireBrick!50,
  draw=FireBrick!75!black,double=Gold,rounded corners=1mm]
  (frame.south west) -- ([xshift=-3mm]frame.west) -- (frame.north west)
  -- ([yshift=2mm]frame.north) -- (frame.north east) -- ([xshift=3mm]frame.east)
  -- (frame.south east) -- ([yshift=-2mm]frame.south) -- cycle;},
interior titled code={\path[outer color=Gold,inner color=white,draw=Gold,
  double=FireBrick!75!black,rounded corners=5mm]
  (interior.south west) rectangle (interior.north east);},
segmentation code={\path[draw=FireBrick,opacity=0.25] ([xshift=2cm]segmentation.west)
  -- (segmentation.north) -- ([xshift=-2cm]segmentation.east)
  -- (segmentation.south) -- cycle;}}

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



8.5 Skin 'bicolor'

`/tcb/skin=bicolor` (skin)

This skin is quite similar to the `standard` → P. 50 and `enhanced` → P. 50 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.

- The most basic usage of this skin is to set the background color of the lower part by `/tcb/colbacklower` → P. 53 and all other options like for the `standard` → P. 50 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 more advanced usage of this skin is to apply the `/tcb/frame style` → P. 50 and the `/tcb/interior style` → P. 50 like for the `enhanced` → P. 50 skin. Also, the `/tcb/segmentation style` → P. 50 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}
```

`/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{skin=bicolor,colback=LightGreen,colframe=DarkGreen,
  colbacklower=LimeGreen!75!LightGreen,
  width=(\linewidth-8mm)/4,before=,after=\hfill,equal height group=bicolg,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}

\begin{tcolorbox}
  This is my content.
\end{tcolorbox}
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}
\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.

```
\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.

8.6 Skin '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 libraries shadings and shadows have to be included in the preamble by:

```
\usetikzlibrary{shadings,shadows}
```

The appearance of the skin can be controlled by /tcb/frame style → P. 50 and /tcb/interior style → P. 50, if needed. Here, the *segmentation* cannot be controlled by a style.

```
\begin{tcolorbox}[skin=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.

```
\tcbset{skin=beamer,colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-8mm)/4,before=,after=\hfill,equal height group=beamg,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}

\begin{tcolorbox}
  This is my content.
\end{tcolorbox}
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}
\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.

8.7 Skin '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. 50, /tcb/interior style → P. 50, and /tcb/segmentation style → P. 50, if needed.

```
\begin{tcolorbox}[skin=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.

```
\tcbset{skin=widget,colback=LightGreen,colframe=DarkGreen,
  width=(\linewidth-8mm)/4,before=,after=\hfill,equal height group=widg,
  left=1mm,right=1mm,top=1mm,bottom=1mm,middle=1mm}

\begin{tcolorbox}
  This is my content.
\end{tcolorbox}
\begin{tcolorbox}
  This is my content.
  \tcblower
  More content.
\end{tcolorbox}
\begin{tcolorbox}[adjusted title=My title]
  This is my content.
\end{tcolorbox}
\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.

References

[1] David Carlisle. *The file ltxdoc.dtx for use with LaTeX 2ε*. Nov. 11, 2007. http://mirror.ctan.org/macros/latex/base/ltxdoc.dtx.

[2] Carsten Heinz and Brooks Moses. *The Listings Package*. Version 1.4. Feb. 2007. http://mirror.ctan.org/macros/latex/contrib/listings/listings.pdf.

[3] Frank Mittelbach. *The doc and shortvrb Packages*. June 30, 2011. http://mirror.ctan.org/macros/latex/base/doc.dtx.

[4] 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.

[5] Thomas F. Sturm. *Mathematik 1 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2007. http://www.unibw.de/bw/Fakultat/we2/Sturm.

[6] Thomas F. Sturm. *Mathematik 2 (B.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2008. http://www.unibw.de/bw/Fakultat/we2/Sturm.

[7] Thomas F. Sturm. *Höhere Mathematik: Fortgeschrittene mathematische Methoden (M.Eng.)* Vorlesungs-Skriptum. Neubiberg: Universität der Bundeswehr München, 2010. http://www.unibw.de/bw/Fakultat/we2/Sturm.

[8] Thomas F. Sturm. *LaTeX – Einführung in das Textsatzsystem*. 8th edition. RRZN-Handbücher. Hannover: Regionales Rechenzentrum für Niedersachsen (RRZN), Sept. 2011. http://www.rrzn.uni-hannover.de/buecher.html.

[9] Till Tantau. *The TikZ and PGF Packages. Manual for version 2.10*. Oct. 25, 2010. http://mirror.ctan.org/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf.

[10] 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