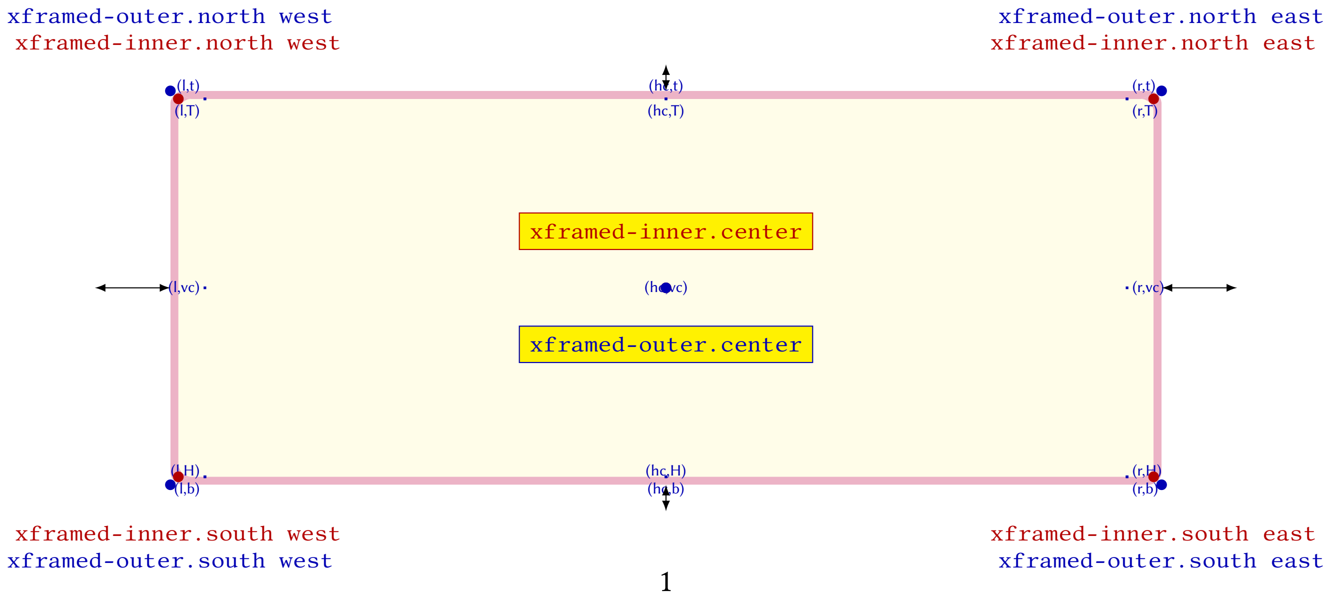
The xframed package

Marco Daniel

XXXXXXX



Contents

Preface **4**

Introduction . . . 4

Bug reports . . . 4

Installation . . . 4

1 Idea behind xframed **5**

2 Usage **6**

2.1 Loading the package . . . 6

2.2 Specifying the options . . . 7

2.3 Helper functions . . . 8

2.4 Creating new environments . . . 10

2.5 Elements of the frame drawn by xframed . . . 11

3 Package options **12**

3.1 Drawing method . . . 12

3.2 Outside the frame . . . 13

3.3 Rules around the frame . . . 15

3.4 Main body of the frame . . . 19

3.5 Title of the frame . . . 22

3.6 Foot of the frame . . . 24

3.7 Tikz elements of xframed . . . 26

3.8 Footnotes . . . 29

3.9 Subtitle(s) . . . 30

3.10 shadow . . . 31

3.11 Hooks . . . 31

3.12 Important typographical notes 33

4 Breaking across pages 37

5 Developer Info 39

6 Examples 43

A Appendix 44

A.1 Thanks 44

A.2 Bugs 44

A.3 ToDo 44

B Revision history 45

Change History 46

Index 47

Preface

Introduction

I am interested in LaTeX and specially in LaTeX3. With this package I want to improve my skills using this great language. However I am a beginner and so the package has only an *alpha* version. If you use this package be aware of this situation. I am sure the great guys at ➔TeX.SX will help me to improve this package.

Bug reports

Bug reports can be done at xframed at Github. If you have no account at Github you can drop me an e-mail ✉marco.daniel@mada-nada.de

Installation

As long as the package isn't available at CTAN you must install (if you dare it) manual. Therefor you can clone the repository in your local texmf tree. I provided the correct folder structure at Github to simplify the installation.

If you are only interested in the usage of the package you can skip this chapter. All options are explain in section 2

Section 1

Idea behind xframed

The idea is very simple. Draw a frame around given material. During my study I wanted a package which can be break across pages and put a frame around this. The package framed[1] didn't require my needs. So I started to write my own package. The result can be found at CTAN, too. It's the package mdframed[2].

After passing my study I started to improve the package mdframed. In 2011 I registered at TeX.SX and learned something about the new language expl3[3]. I was so fascinated about the great work of the LaTeX3 core team that I started my first steps with simple functions. I also wrote a small article about the frontend xparse for the German community. The article was published in *Die TeXnische Komödie 2/2012*. After a while I wanted to provide my own expl3-package. Now here it is.

I know most users love examples. So I am trying to provide a lot. All frames in this documentation are done by `xframed`. So I hope you will have some inspiration. The highlight of listings is done by minted[4].

By the way. The compilation of this document is done with the typesetting engine LuaLaTeX. To simplify my compilation steps I am using the cool tool arara[5].

Now it's time to introduce the package.

[1] Package framed by Don-ald Arse-neau, see CTAN: framed
[2] Package mdframed by Marco Daniel, see CTAN: mdframed
[3] see: http://latex-project.org/latex3.html
[4] Package minted by Kon-rad Ru-dolph, see CTAN: minted, now maintained by G. Poore.
[5] Tool arara by Paulo Cereda, see CTAN: arara

$$\begin{bmatrix}\cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ\end{bmatrix}\begin{bmatrix}a1 \\ a2\end{bmatrix} = \rotatebox{-90}{$\begin{bmatrix}a1 \\ a2\end{bmatrix}$}$$

(*Jaime Soto at TeX.SX*)

Section 2

Usage

The following sections describe the options of the package and the provided environments. The basic environment is equal to the package name `xframed`.

2.1 Loading the package

Before you can use the package, you must load it in your preamble. As usual the package is loaded by `\usepackage`. The following listings shows it.

Listing 1: Loading the package

```
c% Preamble
k\usepackage{xframed}
```

If you have done this you can use the basic environment `xframed`.

```
\begin{xframed}[option-list]
\end{xframed}
```

The environment has one optional argument where you can specify options which are only used for this frame. The following listings demonstrates the usage.

Listing 3: Loading the package

```
c%document body
k\begin{xframed}[option-list]
 The contents of the frame
```

```
k\end{xframed}
```

2.2 Specifying the options

Before you setup any options you must understand hot the frame is drawn. The default method is by using TikZ[6]. TikZ allows a very user friendly way to setup high quality graphics. Therefor all elements are specified by TikZ options. The basic command to manipulate these elements is `\xframedsetuptikz`. The usage is explained in ??. All other options can be set by `\xframedsetup`. Both command can be used in the preamble or inside the document body. If you enclose these commands inside a group the settings will be local, too.

```
\xframedsetuptikz{option-list}
```

The command has one mandatory argument. The mandatory argument accepts only defined options which are explained in ??

All other options can be setup by the command `\xframedsetup`.

```
\xframedsetup{option-list}
```

This command has one mandatory argument. All allowed options are explained in section 3.

```
\newxframedstyle{style-name}{option-list}
```

Often it is useful to declare a style with the needed options. Therefor you can use the command `\newxframedstyle`. The command has two mandatory arguments. The first mandatory argument is the name of the style. Internal the style name is saved as `xframed_style_<style-name>_user`. That means normally you can use every style name without any risk. If the style already exists xframed uses the commmand `\renewxframedstyle` and provides a warning.

[6] Package TikZ by Till Tantau & friends, see CTAN: TikZ

After you have defined a style you can use the name of the style as a legal value of the option `style`.

```
\renewxframedstyle{style-name}{option-list}
```

If you wan to redefine an existing style you can do this by `\renewxframedstyle`. The syntax of the command is equal to `\newxframedstyle`. If the style doesn't exist xframed uses the command `\newxframedstyle` and provides a warning.

```
\addtoxframedstyle{style-name}{option-list}
```

If you want to extend a predefined style you can do this with `\addtoxframedstyle`. The command has the same syntax as `\newxframedstyle`. If the style doesn't exist xframed uses the command `\newxframedstyle` and provides a warning.

`style` = <value>

The option `style` needs the name of a predefined style by `\newxframedstyle` . All options declared with `\newxframedstyle` will be used.

2.3 Helper functions

The next function will normally be used by advanced users. However I decided to put this information here instead of in section 5 Developer Info. All commands have one mandatory argument. The mandatory argument is one of the explained keys in the following sections.

ℹ The mandatory argument of any helper function doesn't accept a meta key.

```
\usexframendlength{key}
```

This command allows you to use the provided length inside an other command. The `key` is one of the length keys explained in the following sections.

```
\showxframendlength{key}
```

The command `\showxframendlength` has one important difference to `\usexframendlength`. The provided length can be printed inside the document. For you my TEX-hacker that means the primitive `\the` is used.

```
\usexframendskip{key}
```

This command is equal to `\usexframendlength` whereby a skip key is required.

```
\showxframendskip{key}
```

I think you know what happen here. See `\showxframendlength`, it's only for skip length.

```
\usexframendcolor{key}
```

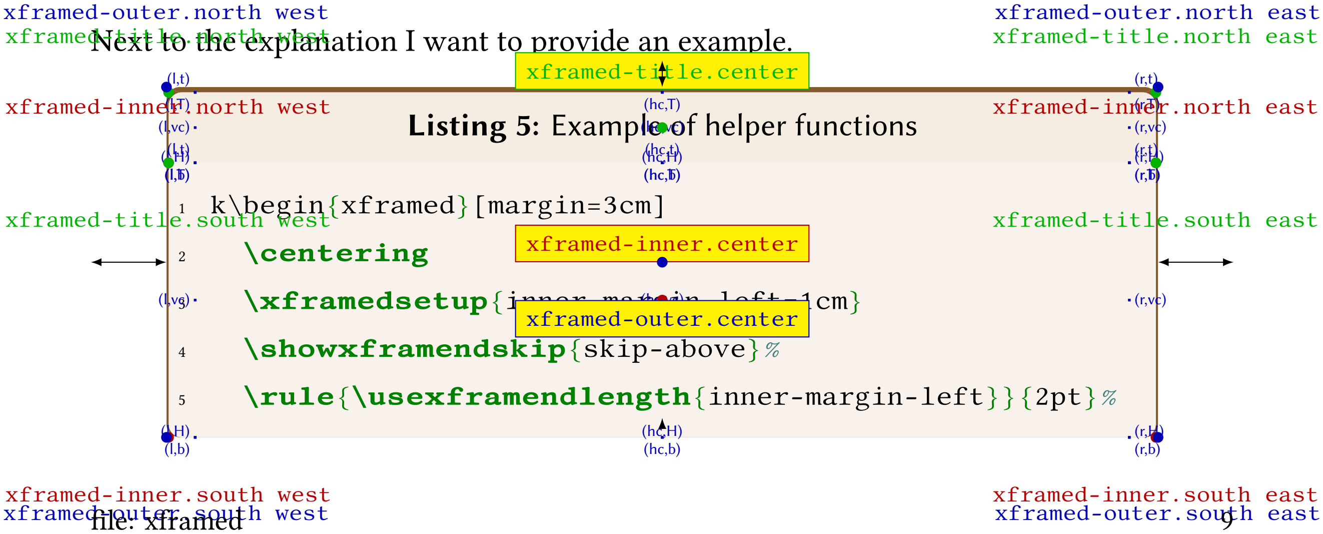
This command is similar to the command `\color`, whereby the color of the provided option is used.

```
\showxframendcolor{key}
```

This command print out the color which is hidden under the key.

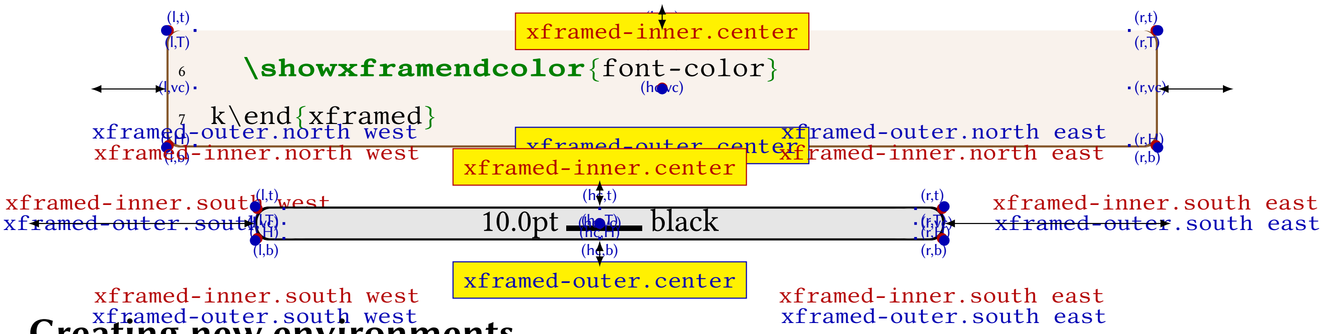
Example

Next to the explanation I want to provide an example.



Listing 5: Example of helper functions

```
k\begin{xframed}[margin=3cm]
   \centering
   \xframedsetup{inner-margin-left=1cm}
   \showxframendskip{skip-above}%
   \rule{\usexframendlength{inner-margin-left}}{2pt}%
```



```
\showxframendcolor{font-color}
k\end{xframed}
```

2.4 Creating new environments

As explained above the package xframed provides one basis environment `xframed`. To provide new environments which use `xframed` you can work with the normal LaTeX 2ε command `\newenvironment` or the command `\NewDocumentEnvironment` provided by xparse. However xframed tries to simplify the process by the following commands.

```
\Newxframedenv[option-list]{env-name}
```

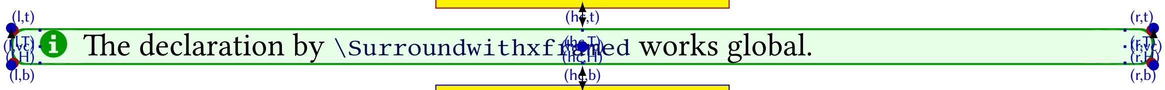
Create a new environment with the name of the mandatory argument of `Newxframedenv`. The optional argument accepts all defined keys of xframed.

```
\Renewxframedenv[env-name]
```

This command is similar to `\Newxframedenv` whereby the command must already exist.

```
\Surroundwithxframed[option-list]{env-name}
```

Sometimes you have predefined environments like `verbatim` where you want to get a colored background. To do this you can surround existing environments with `\Surroundwithxframed` where the mandatory argument is the name of the existing environment. The optional argument accepts all keys defined by xframed.



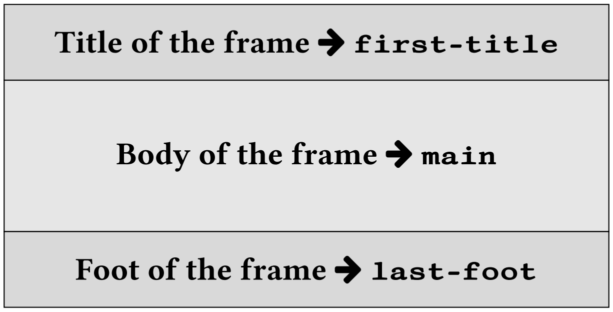
ℹ The declaration by `\Surroundwithxframed` works global.

Before we start with options we need to understand the provided elements of the frame.

2.5 Elements of the frame drawn by xframed

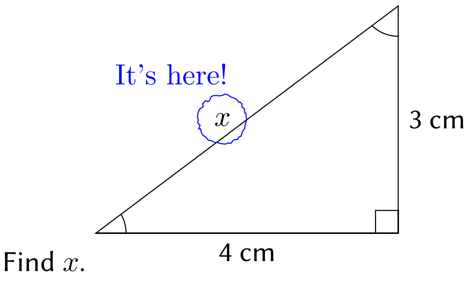
It should be clear the a frame has some rules around. So we can check mark the first relevant point.The next point of the agenda is the main body, the title and the foot. These three elements are very important to understand the behavior. The simple picture below should show the elements and the provided names inside the package.

Figure 1: Base elements of xframed


Title of the frame ➔ first-title
Body of the frame ➔ main
Foot of the frame ➔ last-foot

I know the picture looks very poor at the beginning but we want to concentrate on the main issue. It describes the three base elements of the frame drawn by `xframed`.

Now let's start with all options. Be aware the list ist long.



(Paulo Cereda at TeX.SX)

Section 3

Package options

Every user has his/her own wishes. It's very difficult to implement an evironment which meets all requirements. I hope with the following options you can setup your requirements as best as I was able to implement. As described in section 1 the package uses expl3 in the background. So I can provide more intuitive names. During the explanation I refer to the environment `xframed`. However this is only symbolic. The options are also working for other derivations.

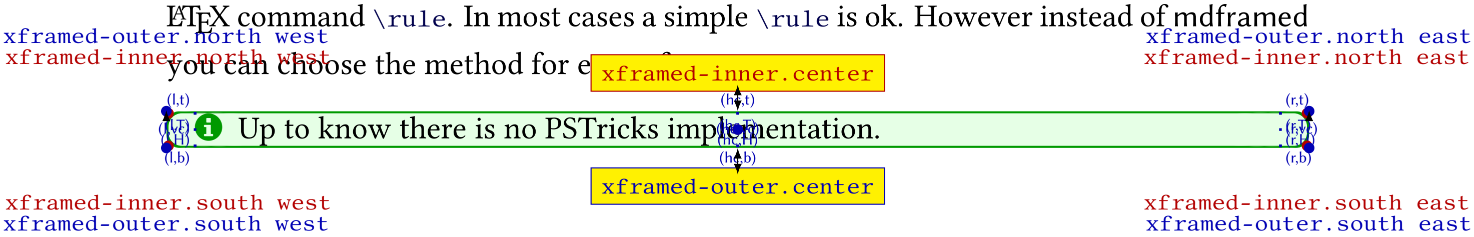
★

xframed provides some meta keys. That means if you pass a value to the meta option more than one other option are influenced. Every meta option has a star ★ right of the name.

3.1 Drawing method

As mdframed I decided to support different methods of frame drawing.

The usage of TikZ or PSTricks needs a lot more compilation time then the normal LaTeX command `\rule`. In most cases a simple `\rule` is ok. However instead of mdframed you can choose the method for e



ℹ Up to know there is no PSTricks implementation.

`frame-method` = <value> default=`tikz`

The option `frame-method` allows you to declare the drawing commands. All allowed options are:

`default, tex, latex, none`	Draw the frame with standard LaTeX 2$_\varepsilon$commands. It needs the least compilation time.
`pgf, tikz`	Draw the frame with TikZ. It needed the highest compilation time.
`pstricks, ps , postscript`	Draw the frame with PSTricks.

`tikz` ★

`default` ★

These options aren't meta option but I want to emphasize them. Theses options are short forms of `frame-method=tikz` respectively `frame-method=default`. The options don't allow any values.

3.2 Outside the frame

Drawing a frame requires some modifications around. So you want to setup the margins or the skips above or below the frame. Related to the meaning of the keys, all keys requiers a length or skip dimension. That mean that the length variables defined as `dim` have a fixed length, whereas `skip` length can have a rubber (stretch/shrink) component.

`width` = <value> default=`\linewidth`

This key allows you to specify the width of the complete frame. Normally you don't need this key. All related length (left margin, right margin) can be specified by options.

`skip` ★

`skip-above` = <value> default=10 pt

`skip-below` = <value>

The lengths represent the space before and after the environment `xframed`. This option `skip` is a meta key and sets the `skip-below` and `skip-above` to the given skip length.

`margin` ★

`margin-left` = <value> default=0 pt

`margin-right` = <value>

Normally the frame `xframed` is drawn about the complete text width. However this isn't often very common shrinking the outer margin. This keys accept a dimension length which specify the left and the right margin. You can also use negative values. In this case the frame is drawn inside the margin of the page.

`extra-skip` = <value> default=0 pt

Sometimes it's useful to add some vertical white space in front of the frame. This can be useful if you want some elements on the lines. To take care of this required space you can set the Option `extra-skip`. Negative dimensions are also allowed whereby I can't image any situation to use it.

This finished the *outside* part for the moment. The package provides also some hooks which will can be used as an option. This isn't really a low level issue so these options are described in subsection 3.11.

Before I will describe the options related to there base element as shown in Figure 1, I will start with the rules around the frame.

3.3 Rules around the frame

A normal frame has four sides. The frame of `xframed` isn't an exception. Of course you can manipulate as possible to get a triangle or a star.

`line-width` ★		default=`0.8pt`
`line-width-left`	= <value>	
`line-width-right`	= <value>	
`line-width-top`	= <value>	
`line-width-bottom`	= <value>	

The first option in this section specify the width of all four lines around the material of `xframed`. This implies the title and the foot of the frame. If you want to setup the rule width of the elements separate you can do this by the following options.

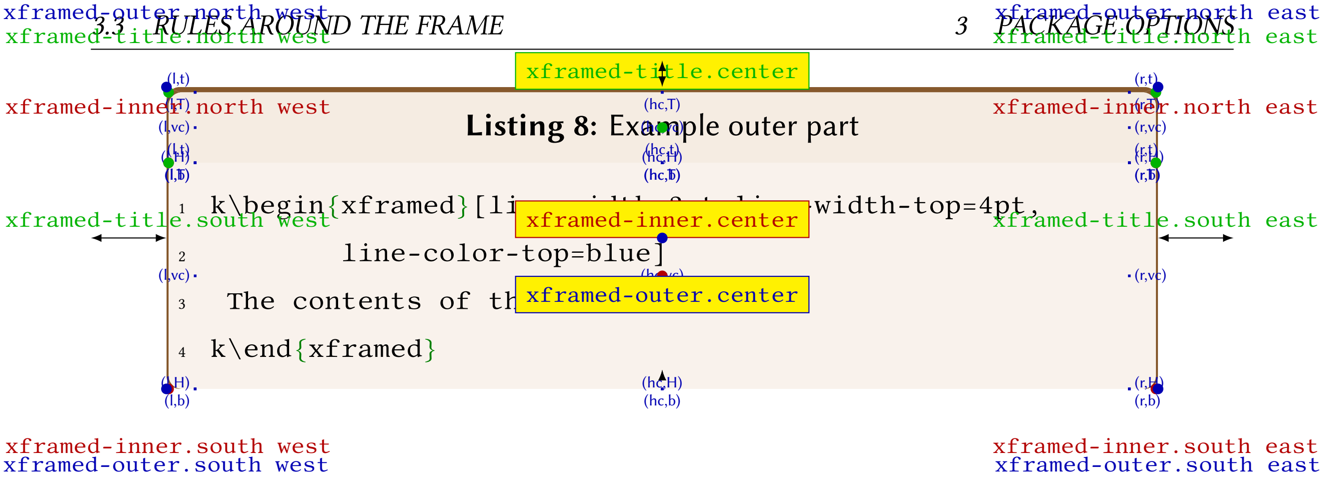
The width is only one part of the lines. I am sure you want to change the colors too.

`line-color` ★		default=`black`
`line-color-right`	= <value>	
`line-color-top`	= <value>	
`line-color-bottom`	= <value>	

Normally all lines have the same color. The color for all four lines can be specified with the Option `line-color`. However the following keys allow the color specification separate.

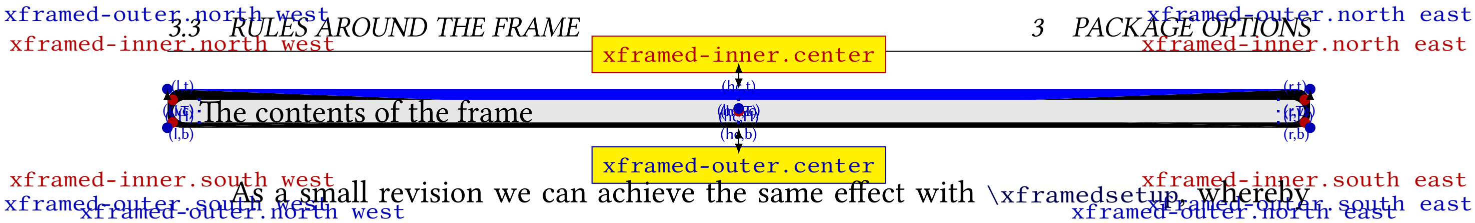
Example

I think it's time for a small example. Suppose you want that all lines has a width of 2 pt expect the top line which shall have a width of 4pt and a different color. This can be achieved by

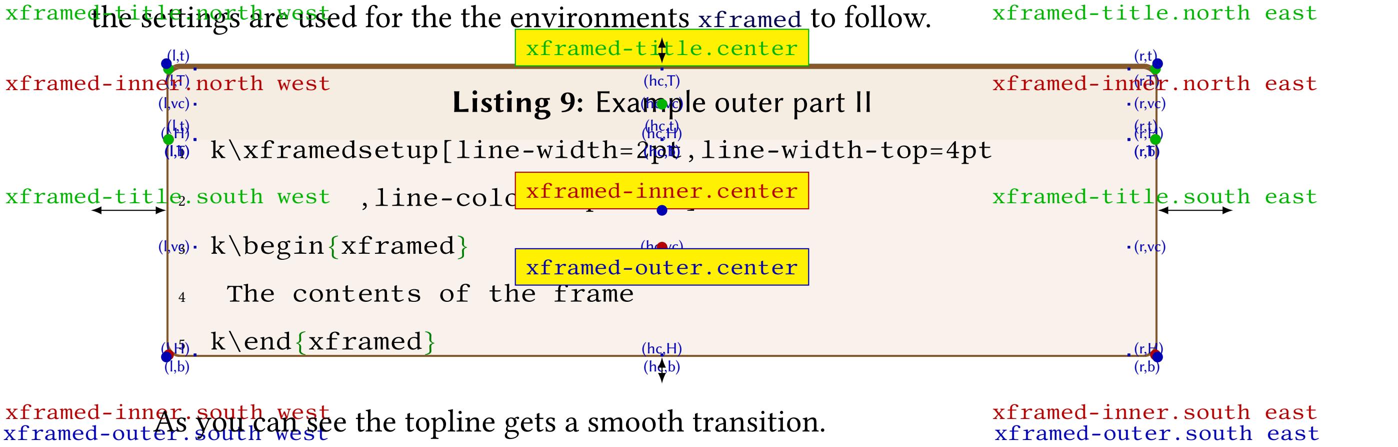


Listing 8: Example outer part

```
k\begin{xframed}[li                    width-top=4pt,
        line-color-top=blue]
 The contents of th
k\end{xframed}
```



As a small revision we can achieve the same effect with \xframedsetup, whereby the settings are used for the the environments xframed to follow.



Listing 9: Example outer part II

```
k\xframedsetup[line-width=2pt,line-width-top=4pt
  ,line-colo[illegible]
k\begin{xframed}
  The contents of the frame
k\end{xframed}
```

As you can see the topline gets a smooth transition.

show-all-lines ★

metakeys By default a frame has four lines around it. Drawing no lines can be achieved by the boolean flag show-all-lines. If you don't want to draw any lines you can pass the value false to the option show-all-lines

top-line	= <value>	
left-line	= <value>	default=true
bottom-line	= <value>	
right-line	= <value>	

bool-option The option show-all-lines controls the behavior of all lines. But you can control every line separate. The option top-line and friend influence the lines of the frame. All four are boolean keys and accepts either true or false.

round-corner ★

arc ★

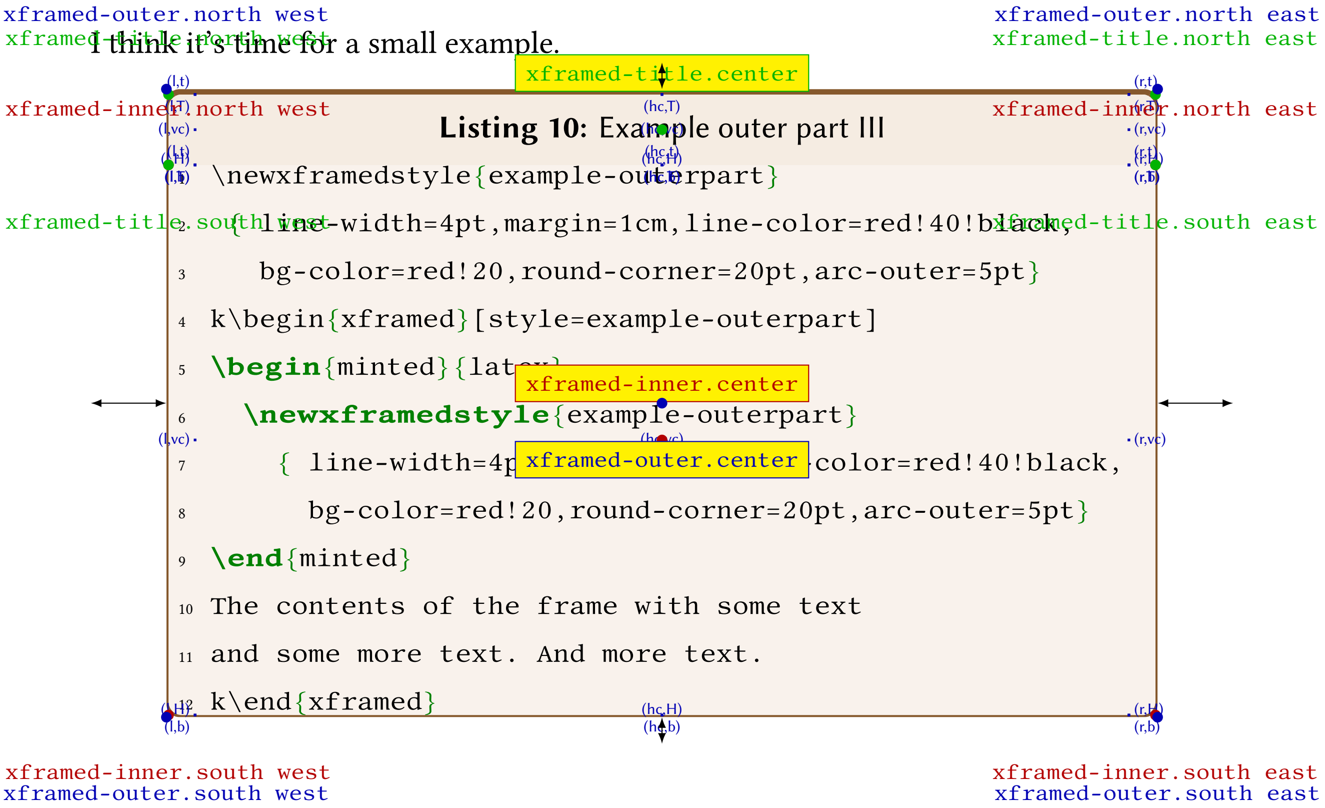
arc-outer = <value> default=10 pt

arc-inner = <value>

The arc of the corners can be controlled by the provided keys. Thereby you can manipulate the outer arc and the inner arc separate. I provided two meta keys because I couldn't provide a correct name. That means the value of round-corner or arc is passed to the options arc-outer and arc-inner.

Example

I think it's time for a small example.



Listing 10: Example outer part III

```
\newxframedstyle{example-outerpart}
  {line-width=4pt,margin=1cm,line-color=red!40!black,
    bg-color=red!20,round-corner=20pt,arc-outer=5pt}
k\begin{xframed}[style=example-outerpart]
\begin{minted}{latex}
  \newxframedstyle{example-outerpart}
    { line-width=4pt,margin=1cm,line-color=red!40!black,
      bg-color=red!20,round-corner=20pt,arc-outer=5pt}
\end{minted}
The contents of the frame with some text
and some more text. And more text.
k\end{xframed}
```

3.4 Main body of the frame

As shown in Figure 1 the body is the main part of the environment `xframed`. Inside the body you can use every material also verbatim one. This part is save in a single coffin [7] and allows such things.

`font` = <value> default=`\normalfont`

The option `font` allows the specification of the main part of `xframed`. It doesn't influence the other part.

`font-color` = <value> default=`black`

If you want to change the font color you can do this with the option `font-color`.

`bg-color` = <value> default=`white`

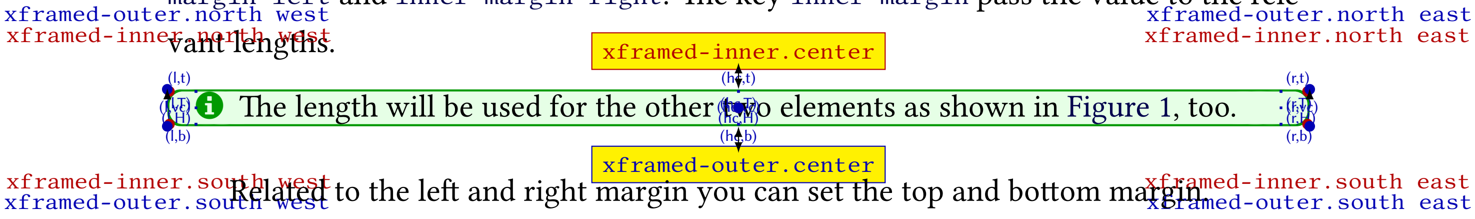
The complete background of `xframed` is specified by the color given as an argument of `bg-color`. If you want some shadings or whatever you can imagine you can the power of TikZ. How to do this is explained in subsection 3.7.

`inner-margin` ★

`inner-margin-left` = <value> default=`10 pt`

`inner-margin-right` = <value>

The distance on the left site and the right site of the frame will be controlled by `inner-margin-left` and `inner-margin-right`. The key `inner-margin` pass the value to the relevant lengths.



ℹ The length will be used for the other two elements as shown in Figure 1, too.

Related to the left and right margin you can set the top and bottom margin.

[7] See: l3coffin

`inner-top-margin` = <value>

This keys sets the top margin of the main part.

`inner-bottom-margin` = <value>

This keys sets the bottom margin of the main part.

Head of the frame

Sometimes you want to have a small head without any break. This can be achieved by the key `head`. I implement this key with some other options to simplify e.g. the creating of a new theorem. There is also another option `first-title` which is used in a new coffin and so put in an extra box. At the moment all captions of the provided listings are done in the `first-title`.

`head` = <value>

Puts the given argument of `head` at the beginning of the main body.

`head-font` = <value> default=`\normalfont\sffamily\bfseries`

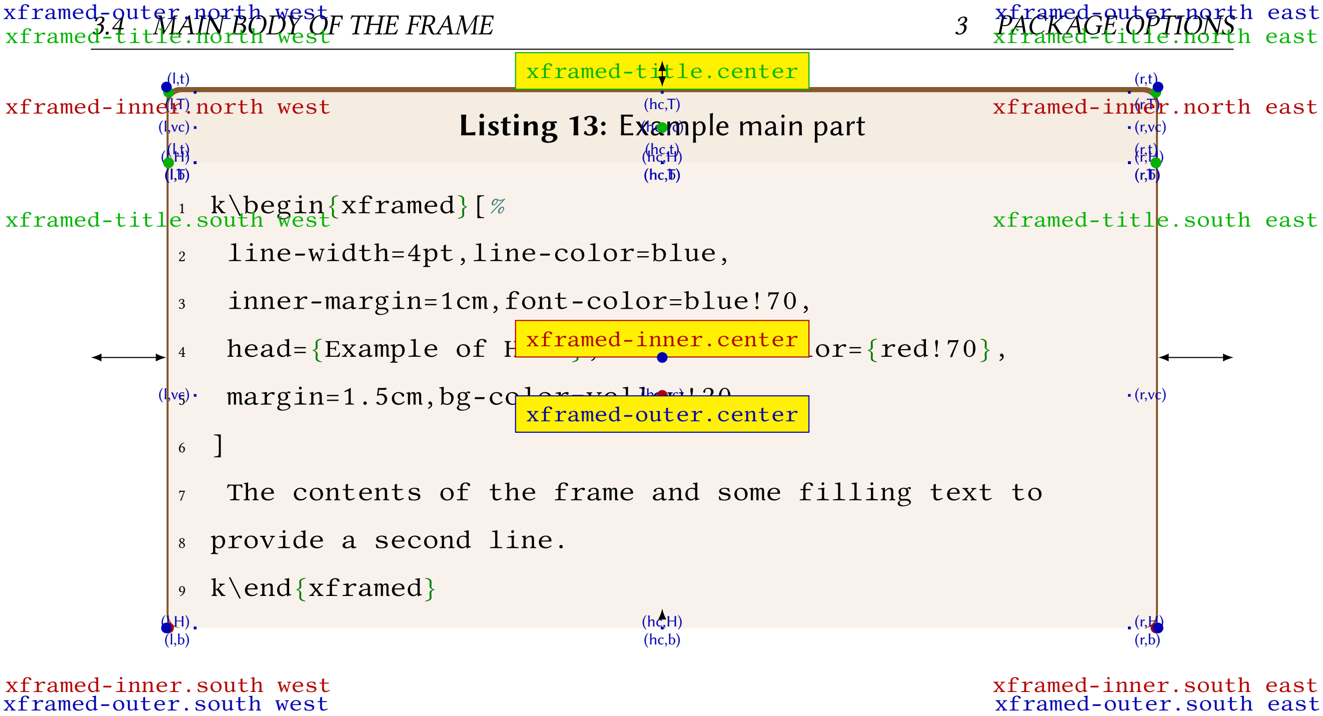
The font is specified by the option `head-font` which be set set local as also the color.

`head-font-color` = <value> default=`black`

Specifies the color of the head.

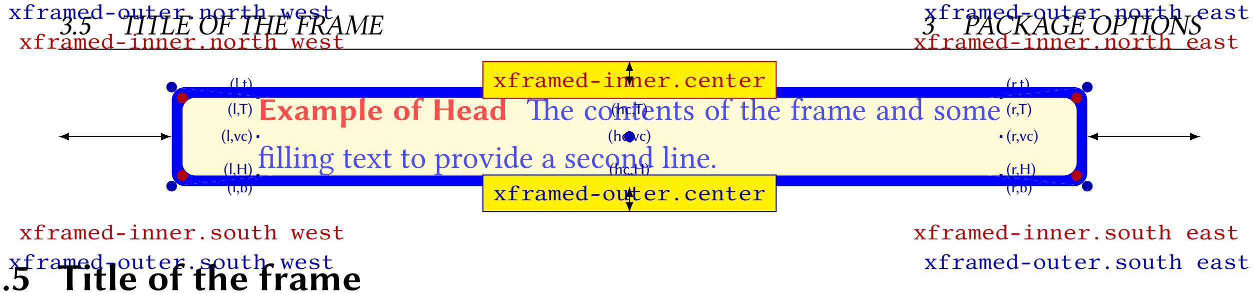
Head of the frame

I guess at this point an example is useful. Instead of explaining I only provide the example.



Listing 13: Example main part

```
k\begin{xframed}[%
 line-width=4pt,line-color=blue,
 inner-margin=1cm,font-color=blue!70,
 head={Example of Head},title-color={red!70},
 margin=1.5cm,bg-color=yellow!30
]
 The contents of the frame and some filling text to
provide a second line.
k\end{xframed}
```



3.5 Title of the frame

The title of the frame is specified by an option and also save in a coffin. This means the paragraphs of the body can't be influence by material of the title. Here you can see the first main differences between `first-title` and `head`. Let us start with options.

`first-title` = <value>

First of all we need a title. The title can be specified by `first-title`. The argument is saved in a single token. Of course you can use line breaks but please en capsule the whole argument in curly braces. Curly braces must be used if you argument contains a comma. May you wonder why it's named `first-title`. If the frame must be splitted the `first-title` is only used at the first frame. If you don't have any splitted frame, you have only one tile the `first-title`.

`title-font` = <value> default=`\bfseries\sffamily\large`

Normally the title should be highlighted. So I decided to declare the list of font commands as default. However you can use the option `title-font` to declare your own font settings.

`title-font-color` = <value> default=`black`

As usual you can specify the color of the font. This is done by the option `title-font-color`.

`title-skip` ★

`title-skip-above` = <value> default=5pt

`title-skip-below` = <value>

As written the title is put inside a new coffin. The explained options `inner-margin`, `inner-margin-left` and `inner-margin-right` will be used on the left and right side of the title component. However you can specify the length above and below the contents of the title. The option `title-skip` passes the argument to the options `title-skip-above` and `title-skip-below`.

> ℹ I know the name `skip` leads to irritations. The length are saving in a dimension register and so any glue is cut off.

`title-bg-color` = <value> default=white

As the main part you can specify a different color as background for the title. This color of the option `title-bg-color` will be used for it.

The package xframed provides a single line between the title and the main body. The following options show you the usage. Of course with methods of subsection 3.7 you can draw dashed lines or other one as well.

`title-line` = <value> default=true

The option `title-line` is a boolean key. If you say `true` a line to separate the material will be drawn. If you say `false`, you will get no line.

`line-width-title` = <value> default=0.6pt

The width of the line is specified by the option `line-width-title` whereby the width doesn't influence the length `inner-top-margin` and `title-skip-below`.

`title-line-color` = <value> default=`black`

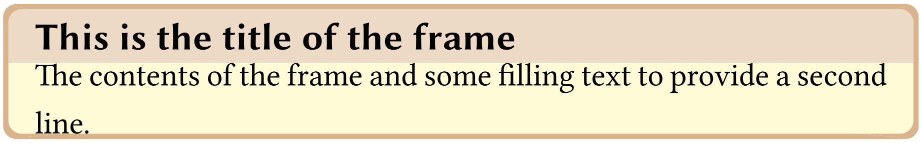
Last but not least the color of the line is done by `title-line-color`.

Example

It's time for a new example to demonstrate the title.

Listing 14: Example title part

```
k\begin{xframed}[title-bg-color=brown!30,%
 line-width=2pt,line-color=brown!60,
first-title={This is the title of the frame},
 margin=1.5cm,bg-color=yellow!20, ]
 The contents of the frame and some filling text to
provide a second line.
k\end{xframed}
```



This is the title of the frame

The contents of the frame and some filling text to provide a second line.

3.6 Foot of the frame

The settings of the foot element are equal to the title element. So the explanation will be short.

`last-foot` = <value>

The foot can be specified by `last-foot`. Of course now you know why it is named `last-foot` (related to `first-title`,

`foot-font` = <value> default=`\bfseries\sffamily\small`

The foot is normally a little bit smaller so I decided to use `\small` as default.

`foot-font-color` = <value> default=`black`

Specify the color of the font.

`foot-skip` ★

`foot-skip-above` = <value> default=`5pt`

`foot-skip-below` = <value>

The option `foot-skip` passes the argument to the options `foot-skip-above` and `foot-skip-below`. The distance between the separation line between the main part and the material of `last-foot` is controlled by `foot-skip-above`. On the other hand `foot-skip-below` specifies the length between the frame and the material of `last-foot`.

ℹ I know the name `skip` leads to irritations. The length are save in a dimension register and so any glue will be cut.

`foot-bg-color` = <value> default=`white`

As the main part you can specify a different color as background for the foot. This color of the option `title-bg-color` will be used for it.

The package xframed provides a single line between the foot and the main body. The following options show you the usage. Of course with methods of subsection 3.7 you can draw dashed lines or other one as well.

`foot-line` = <value> default=`true`

The option `foot-line` is a boolean key. If you say `true` a line to separate the material will be drawn. If you say `false`, you will get no line.

`line-width-foot` = <value> default=`0.6pt`

The width of the line is specified by the option `line-width-foot`.

`foot-line-color` = <value> default=`black`

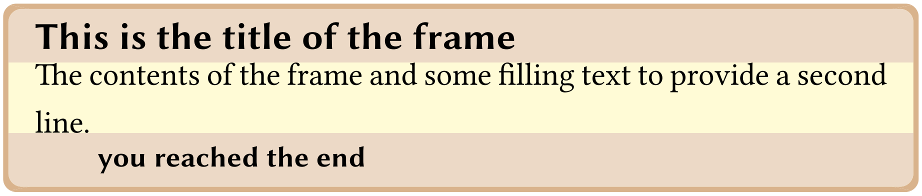
Last but not least the color of the line is done by `foot-line-color`.

Example

It's time for a new example to demonstrate the title.

Listing 15: Example foot part

```
k\begin{xframed}[title-bg-color=brown!30,%
 foot-bg-color=brown!30,line-width=2pt,
 line-color=brown!60,margin=1.5cm,bg-color=yellow!20,
 first-title={This is the title of the frame},
 last-foot={you reached the end},]
 The contents of the frame and some filling text to
provide a second line.
k\end{xframed}
```



This is the title of the frame

The contents of the frame and some filling text to provide a second line.

you reached the end

3.7 Tikz elements of xframed

I often refer this section. The reason is very simple. LaTeXwithout any extension can draw nice graphics. Therefor bundles like PSTricks or TikZ/PGF are needen. This sections shows the implementation using TikZ and so it allows a lot of modification. Please note this documentation doesn't explain TikZ. Therefor you shall consolidate the documentation of TikZ.

setup-tikz = <value>

hooks–tl The command to setup all TikZ styles was introduced in subsection 2.2. The needed command is \xframedsetup. Of course you can also use the option setup-tikz of xframed. Nevertheless the syntax of the implementation is equal, because the value of setup-tikz is passed to \xframedsetup.

Excursus

Let me do a small excursus. TikZ allows very simple to define new styles by using the syntax of pgfkeys. For example you wan to define your own style for some rectangles you can do this as follows.

Listing 16: Excursus TikZ style

```
%   \tikzset{my rectangle/.style={fill=green}}
%   \tikz\draw[my rectangle] (0,0) rectangle (2,0.5);
```

The result will be 🟩. xframed does nearly the same. Instead of using the family Tikz, xframed uses the family xframed. That means every defined style has the prefix xframed. Related to our example above xframed do:

Listing 17: Excursus TikZ style

```
^^A\xframedsetuptikz{my rectangle/.style={fill=green}}
^^A\tikz\draw[xframed/my rectangle] (0,0) rectangle (2,0.5);
```

And you get the same result verb++. end excursus

Next are all defined styles explained. You can change every style with the syntax of TiKZ. I hope I don't forget no element.

bg/.style fill = bg-color

This style controls the background of the main element. The default is a full filled rectangle whereby the color is specified by bg-color.

`title bg/.style` `fill = title-bg-color`

This style controls the background of the title element. The default is a full filled rectangle whereby the color is specified by `title-bg-color`.

`title rule/.style` `draw = title-line-color`
`line widthline-width-title`

This style controls the separation line between the main and the title element.

`foot bg/.style` `fill = foot-bg-color`

This style controls the background of the foot element. The default is a full filled rectangle whereby the color is specified by `foot-bg-color`.

`foot rule/.style` `draw = foot-line-color`
`line width = line-width-foot`

This style controls the separation line between the foot and the title element.

`inner arc/.style` `rounded corners = arc-inner`

This style controls inner arc of the frame.

`outer arc/.style` `rounded corners = arc-outer`

This style controls outer arc of the frame.

`right line/.style` `draw = line-color-right`
`line width = line-width-right`

This style controls the right line of the frame.

`left line/.style`

draw = `line-color-left`
line width = `line-width-left`

This style controls the left line of the frame.

`top line/.style`

draw = `line-color-top`
line width = `line-width-top`

This style controls the top line of the frame.

`bottom line/.style`

draw = `line-color-bottom`
line width = `line-width-bottom`

This style controls the bottom line of the frame.

3.8 Footnotes

I provided an extra section about footnotes because footnotes can't be handled as in a normal text. May you know the issue from environments like `table` or `figure`. Boxes used by xframed have the same limitation. If you use footnotes inside the environment xframed they are printed inside `xframed`. If you have any page breaks the footnotes are always printed at the end of the environment before `last-foot`.

The following options may help you to format the footnotes.

`footnote-distance` = <value> default=`10 pt`

skip keys This skip length is describes the distance between the last line of `xframed` and the the footnote rule.

`footnote-line-width` = <value> default=`.8pt`

The thickness of the footnote rule is specified by this option.

`footnote-line-length` = <value> default=`1 in`

The width of the footnote rule is provided by the value of the option `footnote-line-length`

3.9 Subtitle(s)

`subtitle-skip-above` = <value>

`subtitle-skip-below` = <value>

`subtitle-skip-above` = <value>

skip keys

`subtitle-skip-below` = <value>

skip keys

`subtitle-font-color` = <value>

colorkeys

`subtitle-line-color` = <value>

colorkeys

`subtitle-bg-color` = <value>

colorkeys

`subtitle-font` = <value>

fontoptions–tl

`subtitle-line` = <value>

bool-option

3.10 shadow

`shadow-size` = <value>

`shadow` = <value>

bool-option

`shadow-color` = <value>

colorkeys

3.11 Hooks

What is hook? First time I read hook I thought on Captain Hook of Peter Pan. However hooks in LaTeXaren't pirates. A hook is a macro that isn't used by the package itself. Normally those macros are empty. So the user can redefine hooks to influence the behavior. Common hooks are `\AtBeginDocument` or `\@minipagerestore`.

To allow the user a lot of modifications xframed provides a lot of hooks next to the options.

`code-before` = <value>

`code-after` = <value>

hooks−tl These two hooks are executed before respectively after the material of `xframed`.

`code-begin` = <value>

`code-end` = <value>

hooks−tl These two hooks are executed inside the main element, that means inside the coffin directly before respectively after the material of `xframed`.

`head-pre-code` = <value>

`head-post-code` = <value>

hooks−tl These hooks are nearly equal to `code-begin` and `code-end`. They are executed inside the grouped head.

`title-pre-code` = <value>

`title-post-code` = <value>

hooks−tl These hooks are nearly equal to `code-begin` and `code-end`. They are executed inside the coffin for the title.

`foot-pre-code` = <value>

`foot-post-code` = <value>

hooks−tl These hooks are nearly equal to `code-begin` and `code-end`. They are executed inside the coffin for the foot.

`tikz-code-post` = <value>

If the frame is drawn by TikZ, the value of the option `post-tikz-code` is executed at all TikZ environments.

```
tikz-code-single   = <value>
tikz-code-first    = <value>
tikz-code-middle   = <value>
tikz-code-last     = <value>
```

hooks–tl The last part of the option name leads to the execution location. If the frame isn't splitted the option `tikz-code-single` is executed. The other elements are equal.

```
code-frame-single   = <value>
code-frame-first    = <value>
code-frame-middle   = <value>
code-frame-last     = <value>
```

hooks–tl At the moment these hooks are provided but not implemented.

```
subtitle-before   = <value>
subtitle-after    = <value>
```

hooks–tl

3.12 Important typographical notes

My first package mdframed got a lot of feature request and bug reports (of course). At the moment most of them are fixed. One very important and interesting question was provided by Tobias Weh at TeX.SX. He asked How to make mdframed ignore descenders in last line?. Inspired by the great answer of Stephan Lehmke I implented this feature. However this feature shall be missed in xframed.

ignore-last-descender = <value> default=true

bool-option The option ignore-last-descender does the same as the name say. It ignores the descenders of the last line of an element provided by xframed.

ignore-last-skip = <value> default=true

bool-option This is equal to ignore-last-descender but it ignores the last vertical skip. This is often useful if the contents of xframed ends with an other environment like itemize.

Example

Let me show the meaning of the two options ignore-last-descender and ignore-last-skip.

Listing 18: Example
ignore-last-descender=true

```
% \xframedsetup{%
%    ignore-last-descender=true}
% \begin{xframed}
%   descender not in line
% \end{xframed}
% \begin{xframed}
%   descender in line: skip
% \end{xframed}
```

descender not in line

descender in line: skip

Listing 19: Example `ignore-last-descender=false`

```
% \xframedsetup{%
%     ignore-last-descender=false}
% \begin{xframed}
%   descender not in line
% \end{xframed}
% \begin{xframed}
%   descender in line: skip
% \end{xframed}
```

descender not in line

descender in line: skip

The next example demonstrate the option `ignore-last-skip`.

Listing 20: Example `ignore-last-skip=true`

```
% \xframedsetup{%
%     ignore-last-skip=true}
% \begin{xframed}
%   \begin{itemize}
%     \item foo bar
%  \end{itemize}
% \end{xframed}
```

- foo bar

Listing 21: Example

ignore-last-skip=false

```
% \xframedsetup{%
%    ignore-last-skip=false}
% \begin{xframed}
%   \begin{itemize}
%    \item foo bar
%  \end{itemize}
% \end{xframed}
```

- foo bar

Section 4

Breaking across pages

ℹ This isn’t implemented yet!

It became very popular to have frames which automatic break across pages. As often said my first package can do this. Because xframed shall become the successor of mdframed, I implement it too. However the algorithm differs from the previous one. At the end of the day both are using `\vsplit`.

`allow-breaking` = <value> default=`true`

bool-option To decide whether a break is allowed or not you can use the `allow-breaking`. If you say `allow-breaking=false` the frame will never break. This happens also if you use `xframed` inside floating objects or other boxes.

`minimum-space` = <value> default=`line-width-top+line-width-bottom+2\baselineskip`

When you start the environment `xframed` you can specify the minimum space before the first split occurs. This height is controlled by `minimum-space`.

`split-skip` ★

`split-skip-top` = <value>

`split-skip-bottom` = <value>

skip keys The splitting leads always to added space. The space before and after a split. The option `split-skip-top` controls the distance between the top of the page and the splitted material, whereby `split-skip-top` is part of the frame. The option `split-skip-bottom` does the same for the end of a splitted frame.

`title`	= <value>
`foot`	= <value>

stringoptions I explained the options `first-title,last-foot`. If a frame is splitted you can specify the a title and a foot for all splitted frames. However the options `first-title,last-foot` don't loose there meaning. These options are inspired by longtable.

Section 5

Developer Info

The following lines provide some notes for developers and advanced users.

As explained in the introduction these package uses the new language expl3. So if you have a look at the sty file you will see the new syntax. Maybe the syntax is new for you, but I provides self-explaining function names which should help you.

However instead of the normal TeXor LaTeXcommands `\box` or `\setbox` I am using the modul l3coffin. So all parts are saved as coffins. Of course after all expansion you have a simple box.

It's also important to know that xframed does all the computation of length etc. on the LaTeX 2ε site. This is easier to support different `frame-method`.

At the begin of every environment the package writes the following information to the log file:

Listing 22: Info dimension

```
...............................................
xframed info 58

 dimension of the current page
 height = 591.53027pt = 20.78991 cm
 used = 11.0pt = 0.38661 cm
 free = 580.53027pt 20.40331 cm
 request = 27.20001pt = 0.95597 cm
 request is provided by the option: minimum-space
...............................................
```

It helps you to setup your environment.

`developer-info` = <value>

bool-option Next to the information to the `log` file I provided some helper methods which will be displayed if you set `developer-info` to `true`. An example is shown at the title page. The following happens if you use this options

- The output is done by `\coffin_display_handles:Nn ##1 { blue!70!black }` instead of `\box_use:N` So you can provide other coffins which can be joined.
- It shows you all defined TikZ nodes, if TikZ is in use.
- More information are written in the `log` file.

`\xframedprintalloption[property]`

If you want to controll whether all options are corectly passed you can print out the option by the provided user commands `\showxframendskip`, `\showxframendlength` or `\showxframendcolor`. However this is only useful for one or two options. The `\xframedprintalloption` closed this gap. It prints all option inside a `longtable` with three columns. If you don't use the optional argument all `skip`, `length` and `color` options are printed. With the help of the optional argument you can specify which option shall be printed.

Listing 23: Example `\xframedprintalloption[skip]`

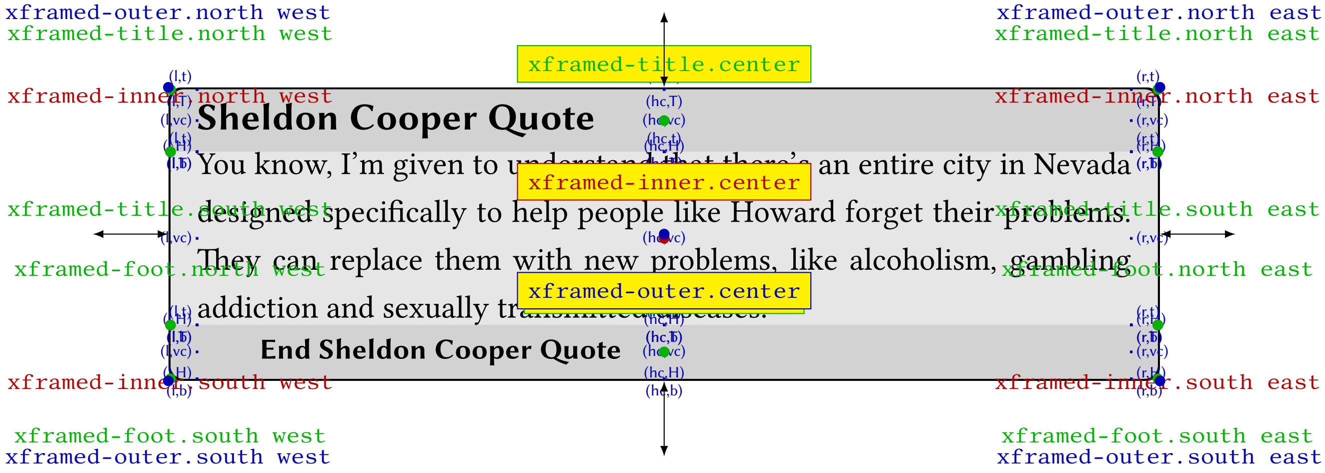
```
\begin{xframed}[margin=1cm,skip=1cm,]
k\xframedprintalloption[skip]
\end{xframed}
```

Skip options

skip-above	28.45274pt	1 cm
skip-below	28.45274pt	1 cm
subtitle-skip-above	5.0pt	0.17573 cm
subtitle-skip-below	5.0pt	0.17573 cm

Listing 24: Example `developer-info`

```
k\begin{xframed}[developer-info=true,margin=1cm,skip=1cm,%
first-title={Sheldon Cooper Quote},%
last-foot={End Sheldon Cooper Quote}]
You know, I'm given to understand that there's an entire
city in Nevada designed specifically to help people like
Howard forget their problems. They can replace them with
new problems, like alcoholism, gambling addiction and
sexually transmitted diseases.
k\end{xframed}
```



Section 6

Examples

Text

Section A

Appendix

A.1 Thanks

Text

A.2 Bugs

Text

A.3 ToDo

`twoside-mode` = <value>

bool-option

Section B

Revision history

May 5 2013

- alpha status

Change History

v0.1alpha

General: first dtx-file 1

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols

*(option) *11*

A

\addtoxframedstyle *7*

allow-breaking(option) *36*

arc*(option) *17*

arc-inner(option) *17*

arc-outer(option) *17*

B

bg-color(option) *18*

bg/.style(option) *26*

bottom-line(option) *16*

bottom line/.style(option) *28*

C

code-after(option) *31*

code-before(option) *31*

code-begin(option) *31*

code-end(option) *31*

code-frame-first(option) *32*

code-frame-last(option) *32*

code-frame-middle(option) *32*

code-frame-single(option) *32*

D

default*(option) *12*

developer-info(option) *39*

E

environments:

xframed *5*

extra-skip(option) *13*

F

first-title(option) *21*

font(option) *18*

font-color(option) *18*

foot(option) *37*

foot-bg-color(option) *24*

foot-font(option) *23*

foot-font-color(option) *24*

foot-line(option) *24*

foot-line-color(option) *25*

foot-post-code(option) *31*

foot-pre-code(option) *31*

foot-skip*(option) *24*

foot-skip-above(option) *24*

foot-skip-below(option) *24*

foot bg/.style(option) *27*

foot rule/.style(option) . . . . *27* footnote-distance(option) 28

footnote-line-length(option) 29

footnote-line-width(option) 28

frame-method(option) 12

H

head(option) 19

head-font(option) 19

head-font-color(option) 19

head-post-code(option) 31

head-pre-code(option) 31

I

ignore-last-descender(option) 33

ignore-last-skip(option) 33

inner-bottom-margin(option) 19

inner-margin*(option) 18

inner-margin-left(option) 18

inner-margin-right(option) 18

inner-top-margin(option) 19

inner arc/.style(option) 27

L

last-foot(option) 23

left-line(option) 16

left line/.style(option) 28

line-color*(option) 14

line-color-bottom(option) 14

line-color-right(option) 14

line-color-top(option) 14

line-width*(option) 14

line-width-bottom(option) 14

line-width-foot(option) 24

line-width-left(option) 14

line-width-right(option) 14

line-width-title(option) 22

line-width-top(option) 14

M

margin*(option) 13

margin-left(option) 13

margin-right(option) 13

minimum-space(option) 36

N

\Newxframedenv 9

\newxframedstyle 6

O

options:

- * 11
- allow-breaking 36
- arc* 17
- arc-inner 17
- arc-outer 17
- bg-color 18
- bg/.style 26
- bottom-line 16
- bottom line/.style 28
- code-after 31
- code-before 31
- code-begin 31
- code-end 31
- code-frame-first 32
- code-frame-last 32
- code-frame-middle 32
- code-frame-single 32
- default* 12

developer-info *39*
extra-skip *13*
first-title *21*
font *18*
font-color *18*
foot *37*
foot-bg-color *24*
foot-font *23*
foot-font-color *24*
foot-line *24*
foot-line-color *25*
foot-post-code *31*
foot-pre-code *31*
foot-skip* *24*
foot-skip-above *24*
foot-skip-below *24*
foot bg/.style *27*
foot rule/.style *27*
footnote-distance *28*
footnote-line-length *29*
footnote-line-width *28*
frame-method *12*
head *19*
head-font *19*
head-font-color *19*
head-post-code *31*
head-pre-code *31*
ignore-last-descender *33*
ignore-last-skip *33*
inner-bottom-margin *19*
inner-margin* *18*
inner-margin-left *18*
inner-margin-right *18*
inner-top-margin *19*
inner arc/.style *27*
last-foot *23*
left-line *16*
left line/.style *28*
line-color* *14*
line-color-bottom *14*
line-color-right *14*
line-color-top *14*
line-width* *14*
line-width-bottom *14*
line-width-foot *24*
line-width-left *14*
line-width-right *14*
line-width-title *22*
line-width-top *14*
margin* *13*
margin-left *13*
margin-right *13*
minimum-space *36*
outer arc/.style *27*
right-line *16*
right line/.style *27*
round-corner* *17*
setup-tikz *26*
shadow *30*
shadow-color *30*
shadow-size *30*
show-all-lines* *16*
skip* *13*
skip-above *13*
skip-below *13*
split-skip* *36*

split-skip-bottom *36*
split-skip-top *36*
style *7*
subtitle-after *32*
subtitle-before *32*
subtitle-bg-color *29*
subtitle-font *30*
subtitle-font-color *29*
subtitle-line *30*
subtitle-line-color *29*
subtitle-skip-above *29*
subtitle-skip-below *29*
tikz* *12*
tikz-code-first *32*
tikz-code-last *32*
tikz-code-middle *32*
tikz-code-post *31*
tikz-code-single *32*
title *37*
title-bg-color *22*
title-font *21*
title-font-color *21*
title-line *22*
title-line-color *23*
title-post-code *31*
title-pre-code *31*
title-skip* *22*
title-skip-above *22*
title-skip-below *22*
title bg/.style *27*
title rule/.style *27*
top-line *16*
top line/.style *28*
twoside-mode *43*
width *12*
outer arc/.style(option) *27*

R

\Renewxframedenv *9*
\renewxframedstyle *7*
right-line(option) *16*
right line/.style(option) *27*
round-corner*(option) *17*

S

setup-tikz(option) *26*
shadow(option) *30*
shadow-color(option) *30*
shadow-size(option) *30*
show-all-lines*(option) *16*
\showxframendcolor *8*
\showxframendlength *8*
\showxframendskip *8*
skip*(option) *13*
skip-above(option) *13*
skip-below(option) *13*
split-skip*(option) *36*
split-skip-bottom(option) *36*
split-skip-top(option) *36*
style(option) *7*
subtitle-after(option) *32*
subtitle-before(option) *32*
subtitle-bg-color(option) *29*
subtitle-font(option) *30*
subtitle-font-color(option) *29*
subtitle-line(option) *30*
subtitle-line-color(option) *29*

subtitle-skip-above(option) *29*

subtitle-skip-below(option) *29*

\Surroundwithxframed *9*

T

tikz*(option) *12*

tikz-code-first(option) *32*

tikz-code-last(option) *32*

tikz-code-middle(option) *32*

tikz-code-post(option) *31*

tikz-code-single(option) *32*

title(option) *37*

title-bg-color(option) *22*

title-font(option) *21*

title-font-color(option) *21*

title-line(option) *22*

title-line-color(option) *23*

title-post-code(option) *31*

title-pre-code(option) *31*

title-skip*(option) *22*

title-skip-above(option) *22*

title-skip-below(option) *22*

title bg/.style(option) *27*

title rule/.style(option) *27*

top-line(option) *16*

top line/.style(option) *28*

twoside-mode(option) *43*

U

\usexframendcolor *8*

\usexframendlength *7*

\usexframendskip *8*

W

width(option) *12*

X

xframed (environment) *5*

\xframedprintalloption *39*

\xframedsetup\MArgs␣ *6*

\xframedsetuptikz\MArgs␣ *6*