# Colorbox

Noriyuki Abe

## 1. 基本的な使い方

**Colorbox**

このような枠囲みを実現するためのパッケージです.

基本的な使い方は次です. `@require: colorbox` します.

```
+Colorbox.colorbox [
  Colorbox.top 5mm;% 設定
] ?:({タイトル}) <
  +p{上に 5mm の空き. 枠に囲まれた文章. }
>
```

これで次のような出力を得ます.

**タイトル**

上に 5mm の空き. 枠に囲まれた文章.

このように `+Colorbox.colorbox [オプション] {タイトル} < 本文 >` の形で指定します. タイトルはオプショナル引数です. オプションは ; 区切りで指定します. 各オプションは引数をとることがあります. 例えば `Colorbox.top` オプションは枠内の上側の空きを指定するオプションですが, これは length 型の引数を一つとります. 上のように `[Colorbox.top 5mm;]` のように書きます. 以下オプションの解説です. なお, `Colorbox.` は省略します.

`left`, `right`, `top`, `bottom` : いずれも length 型の引数を一つとります. 枠内における本文と枠の間の空きです. それぞれ左, 右, 上, 下の空きを指定します.

`boxsep` : 枠内における本文と枠の間の空きを指定します. 上述の `left`, `right`, `top`, `bottom` に加算されます.

`left-skip`, `right-skip`, `top-skip`, `bottom-skip` : いずれも length 型の引数を一つとります．枠の外の空きを指定します．

`leftrule`, `rightrule`, `toprule`, `bottomrule` : いずれも length 型の引数を一つとります．枠線の太さを指定します．`boxrule` により，一括で指定することもできます，

`coltext` : color 型の引数を一つとります．`coltext Color.red` または `coltext (Color.rgb 0.1 0.2 0.3)` のように指定します．

`colframe` : color 型の引数を一つとります．枠の色を指定します．

`colback` : color 型の引数を一つとります．枠内の背景色を指定します．

`title-filled` : bool 型（`true` または `false`）の引数を一つとります．`true` を指定すると以下の `colbacktitle` の指定が有効になります．

`colbacktitle` : color 型の引数を一つとります．タイトル部の背景色を指定します．

`lefttitle,righttitle,toptitle,bottomtitle` : length 型の引数を一つとります．それぞれタイトルの左，右，上，下の空きを指定します．

`arc` : length 型の引数を一つとります． デフォルトでは枠の四隅が四分円になりますが，この半径を指定します．`0pt` を指定すると丸みがなくなります．

`sharrp-corners-northwest`, `sharrp-corners-northeast`, `sharrp-corners-southwest`, `sharrp-corners-southeast`, `sharrp-corners-north`, `sharrp-corners-south`, `sharrp-corners-east`, `sharrp-corners-west`, `sharrp-corners-downhill`, `sharrp-corners-uphill`, `sharrp-corners-all` : 枠の角をとがらせます．

`attach-boxed-title-on-top-left` : 引数はありません． 以下のようなタイトルの配置ができます．

タイトル

中身

これは次のコードにより得られます．以下このような形のタイトルをボックス型タイトルと呼びます．

```
+Colorbox.colorbox [Colorbox.attach-boxed-title-on-top-left;] ?:({タイトル}) <+p{中身}>
```

attach-boxed-title-on-top-center とするとタイトルが中央に配置されます．attach-boxed-title-on-top-right で右側です．また attach-boxed-title-on-bottom-left とすると次のように枠の下にタイトルが配置されます．

中身

タイトル

attach-boxed-title-on-bottom-center および attach-boxed-title-on-bottom-right により枠の下の中央および右側にタイトルが配置されます．

flip-title：下のように枠の下端に通常形式のタイトルが表示されます．

中身

タイトル

boxtitle-xshift，boxtitle-yshift：どちらも length 型の引数を一つとります．ボックス型タイトルが指定されているときのみ有効です．それぞれ左右方向，上下方向にタイトルの位置を移動します．右および上が正です．

boxed-title-colframe： color 型の引数をとります． ボックス型タイトルが指定されているときのみ有効です．タイトルの枠の色を指定します．

boxtitle-xshift，boxtitle-yshift： length 型の引数をとります． ボックス型タイトルが指定されているときのみ有効です． タイトル自身を横方向および縦方向にずらします．

drop-shadow：影をつけます．他にも drop-midday-shadow，drop-shadow-southeast，drop-shadow-south，drop-shadow-southwest，drop-shadow-west，drop-shadow-northwest，drop-shadow-north，drop-shadow-northeast，drop-shadow-east があり，指定された方向にずれた影がつきます．

watermark-textwatermark-text-on-brokenwatermark-text-on-unbrokenwatermark-text-on-firstwatermark-text-on-middlewatermark-text-on-lastwatermark-text-on-unbroken-and-firstwatermark-text-on-middle-and-lastwatermark-text-on-first-and-middle：透かしを入れます．inline-text を指定します．二つ目以降はどの枠に出力するかの指定です．unbroken がページで分割されていない枠です．broken は分割された枠．first はページで分割された枠の最初のページ，last が最後，middle がその他です． どの枠かの指定を含めた命令は，関係ない部分には影響を与えませ

ん．例えば次のような例を考えてみます．

```
[
  Colorbox.watermark-text {あ};
  Colorbox.watermark-text-first {い};
]
```

このとき分割された枠の 2 ページ目には「い」が透かしとして入ります．

`watermark-zoom`：浮動小数を一つ引数にとります．透かしの大きさを，枠の大きさに対する相対的な大きさで指定します．

`watermark-color`：色を一つ引数にとります．透かしの色を指定します．

## 2. 低レベルな命令

より柔軟な枠などの構成を行うために，低レベルな命令 `colorbox-scheme` が定義されています．次のように使います．

```
Colorbox.colorbox-scheme config frame config2 ctx ?:title-txt txt
```

ここで

- config: 基本的な設定．
- frame: 枠の描画の設定．
- config2: さらなる設定
- ctx: context．
- title-txt: タイトル．オプション引数，`inline-text` 型．
- txt: 本文，`block-text` 型．

です．戻り値は `block-boxes` 型です．引数については以下説明します．なお，これらの引数は原則として副作用を起こしてはなりません．

基本的な設定 `config` は以下のフィールドを持つレコード型です．

- left-skip: length 型．枠の外の左の空き．

- right-skip: length 型．枠の外の右の空き．
- before-skip: length 型．枠の外の上の空き．
- after-skip: length 型．枠の外の下の空き．
- left: length 型．枠内のテキストまでの左空き．
- right: length 型．枠内のテキストまでの右空き．
- top: length 型．枠内のテキストまでの上空き．
- bottom: length 型．枠内のテキストまでの下空き．
- color-back: color 型．枠内の背景色．

枠の描画設定 `frame` は以下のフィールドを持つレコード型です．

- left-top: 左上枠の描画を行う関数．
- right-top: 右上枠の描画を行う関数．
- left-bottom: 左下枠の描画を行う関数．
- right-bottom: 右下枠の描画を行う関数．
- top-rule: 上枠の描画を行う関数．
- bottom-rule: 下枠の描画を行う関数．
- left-rule: 左枠の描画を行う関数．
- right-rule: 右枠の描画を行う関数．

このように， 枠の描画は全体を八個にわけて行われます． 引数 `frame` の各フィールドは，`colorbox-scheme` が描画を行う際に呼び出されます．

四隅を描画する関数（`left-top`，`right-top`，`left-bottom`，`right-bottom`）は次の形で呼び出されます．

```
frame#left-top config point
```

config は colorbox-scheme の第一引数に与えられたものがそのまま与えられます. point は対応する枠の端の点です. 例えば left-top の場合は左上の点が与えられます. この関数は以下の値を返さなければなりません.

```
(from,to,path,graphic)
```

path は prepath -> prepath 型で，枠のパスを返します．反時計回りのパスを返さなければなりません. from と to はこのパスの始点と終点です. 両方とも length * length 型です. colorbox-scheme は残りの枠をこの始点と終点同士を結ぶように描画します. graphic が実際に描画されたグラフィックで, graphics list 型です.

残りの四カ所の枠（top-rule, bottom-rule, left-rule, right-rule）は次のように呼び出されます.

```
frame#top-rule config from to
```

config は colorbox-scheme の第一引数に与えられたものがそのまま与えられます. from と to はこの部分の枠の始点と終点です．返す値は次です.

```
(path,graphic)
```

path は prepath -> prepath 型で， 枠のパスを返します. graphic が実際に描画されたグラフィックで, graphics list 型です.

さらなる設定 config2 は次のフィールドを持つレコード型です.

```
title = (タイトル描画設定);
overlay = (追加描画命令);
shadow = (追加描画命令);
```

タイトルの描画は次のフィールドを持つレコード型です.

1. make-box: 与えられたタイトル（inline-text 型）から inline-boxes 型のボックスを作る．また，どの場合にタイトルを描画するかも通知します.
2. shifts: 枠や本文などのずらし量を返す関数です.
3. graphics: 実際の描画を行う関数です.

make-box は次の形で呼び出されます.

```
config2#title#make-box config frame ctx text
```

config と frame は ccolorbox-sheme に渡されているものがそのまま渡されます. ctx は context で, text が inline-boxes 型のタイトルです. これに対して, make-box は次を返さなければなりません.

```
(title-box,output-indexes)
```

title-box は与えられた title から構築された inline-boxes です. テキストに色をつけるなどの場合はここで行います. output-indexes は int list 型で, ページが分割されている場合にどこで出力するかを表します. それぞれ口型, 冂型, ｜｜型, 凵型に対応して 0,1,2,3 を入れてリストで返します. 例えば枠の下部にタイトルを出す場合には, 口型と凵型の場合に出力することになるので, [0;3] とします. この関数は枠の描画時に一度しか呼び出されません. そのことを理解した上で副作用を引き起こす関数を指定することも可能です.

タイトルがある場合には, その分枠自身や本文を上下にずらす必要が出てきます. この値は shifts で返します. これは次のように呼び出されます.

```
config2#title#shifts config frame ctx text-box
```

config, frame, ctx は make-box と同じです. text-box はタイトル文字列のボックスで, これは make-box で返したものがそのまま渡されます. 次の length * length * length * length 型の値を返さなければなりません.

```
(shift1,shift2,shift3,shift4)
```

それぞれ, 上部枠のずらし, 本文上部のずらし, 下部枠のずらし, 枠の後の文書本文のずらしを指定します. 正の値が上部方向にずらすことになります. または次のように考えることもできます : -shift1 が枠直前本文と上部枠の間の空き, -shift2 が上部枠と本文上部の空き, -shift3 が本文下部と下部枠の空き, -shift4 が下部枠と続く本文の間の空き.

実際の描画を担当する関数 graphics は次の形で呼び出されます.

```
config2#title#graphics config frame ctx index point width height depth
text-box
```

config, frame, ctx, text-box は shifts と同じです. index は枠がページで分割されている場合にどの形の枠であるかを表します. 値は同様にそれぞれ口型, 冂型, ｜｜型, 凵型に対応して 0,1,2,3 です. point は枠で囲まれた全体の基準点, width, height, depth はそれぞれ横幅, 高さ, 深さです. 例えば枠全体の左下は (x, y -' depth) となります. 関数 graphics はタイトル部分を表す graphics list を返さなければなりません.

追加描画命令 overlay と shadow はさらに加えて描画を行う時に使います. 両者の違いは描画のタイミングで, shadow は枠やタイトルなどの描画の前に, overlay はそれらの後に行われます. どちらも次の形の関数からなるリストです.

```
colorbox-config -> colorbox-frame -> int -> (length * length) -> length
-> length -> length -> (graphics list)
```

つまり, 次の形で呼び出される関数 func のリストです : func config frame path index point width height depth. config と frame は colorbox-scheme に渡されたものそのままです. index は config2#title#graphics に与えられるものと同様で, 口型, 冂型, ｜｜型, 凵型に対応して 0,1,2,3 の値が与えられます. 残りも同様で, point は枠で囲まれた全体の基準点, width, height, depth はそれぞれ横幅, 高さ, 深さです.

## 2.1. 補助関数

低レベル命令での枠の構築をサポートするために, 次の関数が提供されています.

- make-frame-path 枠のパスを構築する命令です. 次のように使います. make-frame-path config frame index pt width height depth 引数の意味は overlay などに渡されるものと同様です. 少し値をいじって渡すことで一回り大きい枠なども構築できます.

# 3. 用意されている枠とタイトル

## 3.1. circle-frame

四隅が丸くなる枠です. 次のように使います

```
Colorbox.circle-frame (|
  color = Color.red; % 枠の色
  top-rule = 1mm; % 上の枠の太さ
  bottom-rule = 1mm; % 下の枠の太さ
```

```
  right-rule = 1mm; % 右の枠の太さ
  left-rule = 1mm; % 左の枠の太さ
  left-top-radius = 3mm; % 左上隅の枠の半径
  right-top-radius = 3mm; % 右上隅の枠の半径
  left-bottom-radius = 5mm; % 左下隅の枠の半径
  right-bottom-radius = 5mm; % 右下隅の枠の半径
|)
```

## 3.2. normal-title

上下に出てくるタイトルです. `+colorbox` のデフォルトです. 次のように使います.

```
normal-title (|
  top = 1mm; % 上空き
  bottom = 1mm; % 下空き
  left = 1mm; % 左空き
  right = 1mm; % 右空き
  fonttitle = [bold] % タイトルのフォント指定. satysfi-fss の style list
  color = Color.blue; % タイトルの文字色
  color-back = Color.black: % 背景色
  bottom-rule = 1mm; % タイトルと本文を区切る枠の太さ
  color-frame = Color.red; % タイトル周りの枠の色
  at-bottom = true; % 枠の下にタイトルをおくかどうか
|)
```

## 3.3. boxed-title

枠に囲まれて独立に出るタイトルです. `attached-boxed-title-top-left` などで出てくるやつです. 次のように使います.

```
boxed-title (|
  frame = ...; % 枠指定. colorbox-scheme の第二引数と同様
  top = 1mm; % 上空き
  bottom = 1mm; % 下空き
  left = 1mm; % 左空き
  right = 1mm; % 右空き
```

```
  fonttitle = [bold] % タイトルのフォント指定. satysfi-fss の style list
  color = Color.blue; % タイトルの文字色
  color-back = Color.black: % 背景色
  shift-x = ... % 横方向のずらし. 後で説明.
  shift-y = ... % 縦方向のずらし. 後で説明.
  at-bottom = true; % 枠の下にタイトルをおくかどうか
|);
```

shift-x には枠全体の横方向のずらし量を指定します．右方向が正です．これは length -> length -> length という形の関数を指定します．第一引数に枠自身の幅，第二引数にタイトルの幅が渡されて呼び出されます. shift-y は枠全体の縦方向のずらし量を指定します． 上方向が正です． 同様に length -> length -> length という形の関数ですが， 第一引数にはタイトルの高さ，第二引数にはタイトルの深さが与えられて呼び出されます.