

figbox マニュアル

@monaqa

1. figbox とは

figbox は SATySF_I で図を整列配置するためのパッケージです。

2. figbox パッケージの概要

figbox パッケージを用いるには冒頭に以下のように記述します。

```
@require: figbox
```

```
open FigBox
```

以下の例では figbox パッケージを入れ、FigBox モジュールを open していることを前提とします。

figbox パッケージは主に以下の 2 種類の機能を提供します。

- figbox 型の値を生成する関数。1 つ以上の figbox を受け取って新たな figbox を返すものもある。
- figbox 型の値を、インラインテキストやブロックテキストとして埋め込むためのコマンド。

たとえば以下のように書けば、有名なパングラムである “The quick brown...” を横幅 100pt で行分割して得られるテキストボックスを、紙面に中央揃えで表示することができます。

```
+fig-center(textbox-with-width 100pt {The quick brown fox jumps over  
the lazy dog.});
```

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

ここで `+fig-center` は、引数として与えられた `figbox` 型の図を紙面に中央揃えで配置するコマンドです。また、`textbox-with-width` という関数は `length -> inline-text -> figbox` 型の関数であり、指定した長さを横幅に、指定したテキストを内容に持つよう行分割された段落を「図」として生成します。ここでの「図」とは、もちろん `figbox` 型の値、ということです。

`figbox` は図の内容そのものを表す型です。もう少し詳しく述べると、「テキスト処理文脈を受け取ることで、図として出力できるようになるもの」となります。そういう意味では `inline-text` 型や `ctx -> inline-boxes` 型に近いかもしれません。

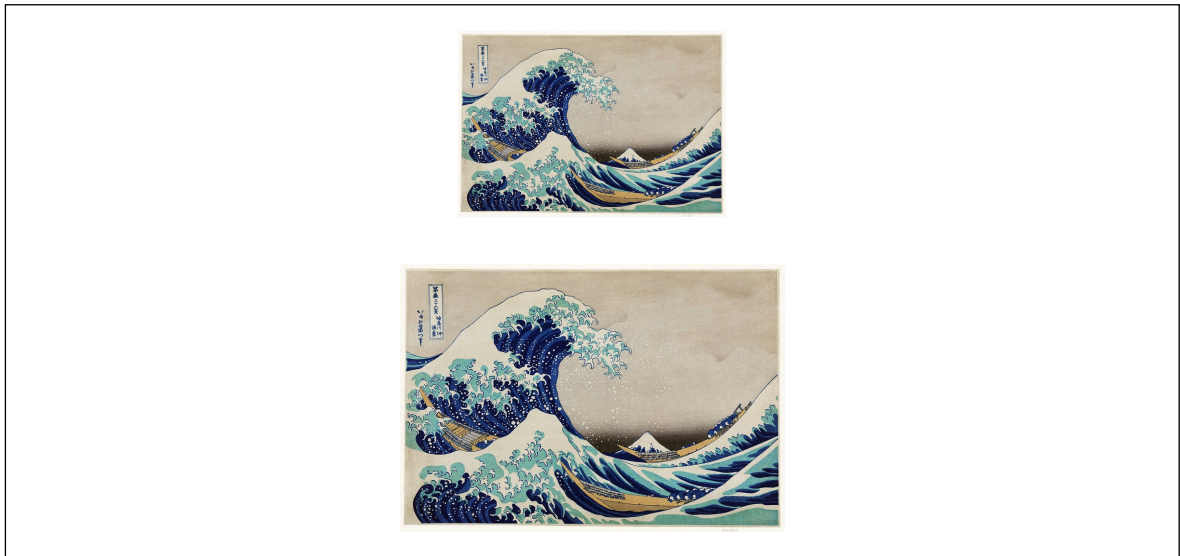
`figbox` を受け取って埋め込むコマンドは `+fig-center` 以外にもいくつか用意されており、`figbox` を生成するための関数も様々です（自作することもできます）。`figbox` 型を介したインターフェース及び関数型言語の文法を活用することで、複雑な図の配置を直感的な式で表現することができます。次の章で具体例を見てみましょう。

3. Gallery

3.1. `figbox` を生成する関数

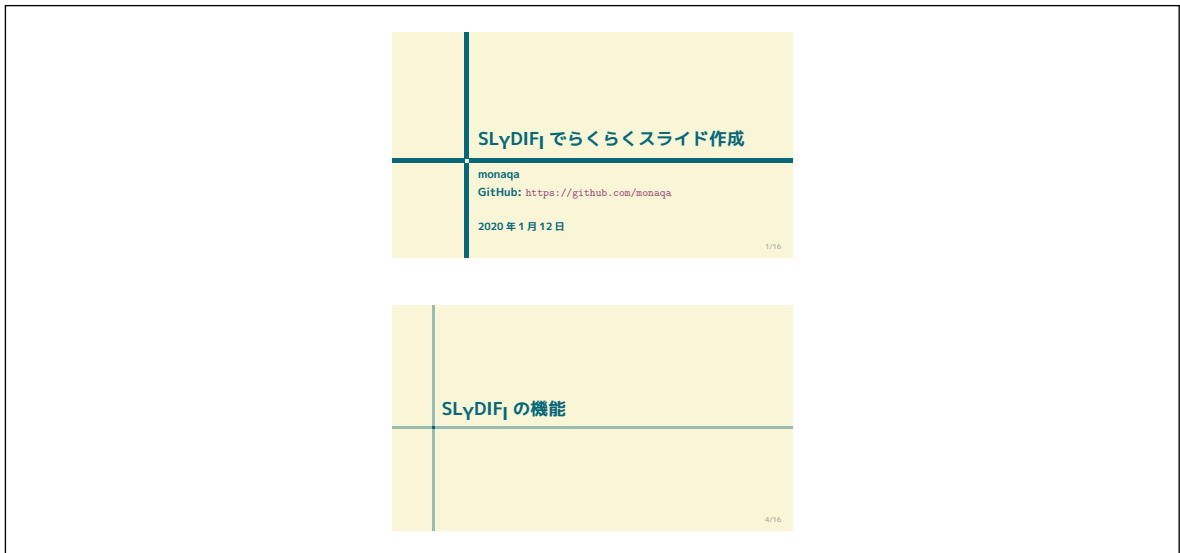
「図」として思い浮かぶ代表的なものは画像でしょう。図は `include-image` 及び `include-image-with-height` を用いて読み込むことができます。`include-image` は画像の横幅を、`include-image-with-height` は縦幅を指定します。

```
+fig-center(include-image 100pt `fig/example1.jpg`);  
+fig-center(include-image-with-height 100pt `fig/example1.jpg`);
```



上の例では JPEG 形式の図を読み込みましたが、PDF 形式の図を指定することもできます。PDF の場合はデフォルトで 1 ページ目の図を読み込むものの、オプション引数にて読み込みたいページ数を与えることもできます。

```
+fig-center(include-image 150pt `fig/example.pdf`);  
% 4 ページ目を読み込む  
+fig-center(include-image ? :4 150pt `fig/example.pdf`);
```

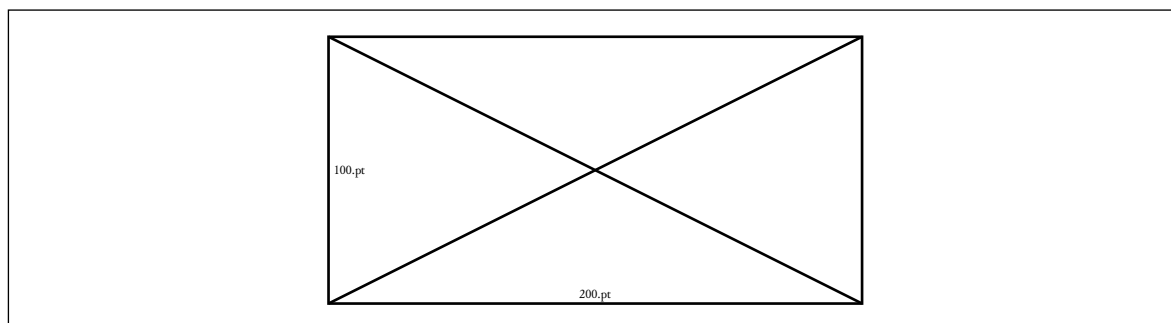


もちろん、`include-image-with-height` でも同様のことができます。

縦横の長さを指定して、ダミーの図を配置することもできます。図の大きさが判明してき

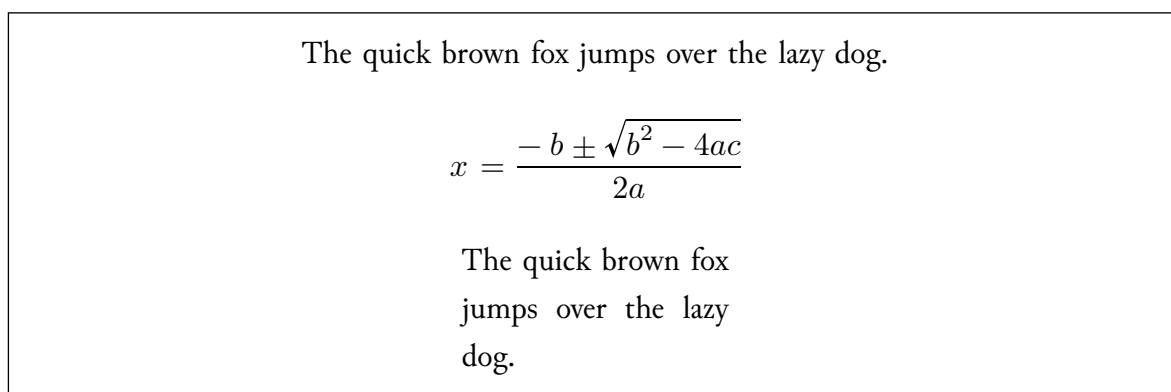
えいれば、まだ図そのものが無くても図を除いたレイアウトが再現できて便利です。

```
+fig-center(dummy-box 200pt 100pt);
```



テキストや数式をはじめとしたインラインテキストやブロックテキストも `figbox` 型に変換することができます。SATySFi では表やグラフィックスもインラインテキストに埋め込むことができるため、実質何でもあり、といえるかもしれません。

```
+fig-center(textbox {The quick brown fox jumps over the lazy dog.});  
+fig-center(textbox { $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ });  
+fig-center(textbox-with-width 100pt {The quick brown fox jumps over  
the lazy dog.});
```



SATySFi では表やグラフィックスもインラインテキストに埋め込むことができるため、実質何でもあり、といえるかもしれません。以下は `easytable` パッケージを用いて組んだ表を中央揃えにする例です。

```
+fig-center(textbox {\easytable[1; c; r;]{  
| header1      | header2      | header3
```

```
| align left | align center | align right
| a          | b          | c
| }));
```

header1	header2	header3
align left	align center	align right
a	b	c

オプション引数にテキスト処理文脈の変換関数（`context -> context` 型）を入れることで書式を変更することもできます。

```
+fig-center(
  textbox ?:(set-font-size 9pt)
    {The quick brown fox jumps over the lazy dog.}
);
+fig-center(
  textbox-with-width
    ?:(fun ctx -> ctx |> set-font-size 14pt
      |> set-text-color Color.blue)
    200pt
    {The quick brown fox jumps over the lazy dog.});
```

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the
lazy dog.

3.2. figbox を変換・結合する関数

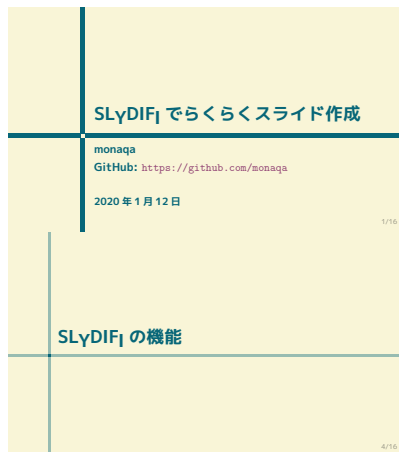
ここまでの例では、figbox パッケージをわざわざ用いる有難みが感じられなかったかもしれませんが。このパッケージの強みは、上で述べた関数を使って得られた「図」をいくつも組み合わせ、より複雑な図を簡単に作成できる点にあります。

figbox パッケージでは、複数の図を縦や横に結合することができます。

```
+fig-center(hconcat [
  include-image-with-height 100pt `fig/example1.jpg`;
  include-image-with-height 100pt `fig/example2.jpg`;
]);
```



```
+fig-center(vconcat [
  include-image 150pt `fig/example.pdf`;
  include-image ? :4 150pt `fig/example.pdf`;
]);
```



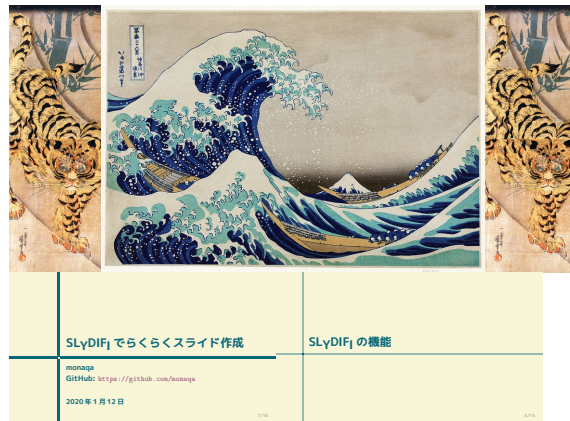
`hconcat` や `vconcat` の返り値もまた `figbox` 型であるということに注意してください。つまり、これらは自由にネストさせることができます。

```
+fig-center(
  vconcat[
    hconcat [
```

```

include-image-with-height 100pt `fig/example2.jpg`;
include-image-with-height 100pt `fig/example1.jpg`;
include-image-with-height 100pt `fig/example2.jpg`;
];
hconcat [
include-image 100pt `fig/example.pdf`;
include-image ?:4 100pt `fig/example.pdf`;
];
]
);

```



`hconcat` や `vconcat` 関数は単に図をつなげるだけであり、間に余白を入れてくれません。余白を入れるときは間に `gap` という関数により作成される特殊な `figbox` を指定します。これは `hconcat` 関数の中では横に隙間を開けるはたらきを、`vconcat` 関数の中では縦に隙間を開けるはたらきをします。

```

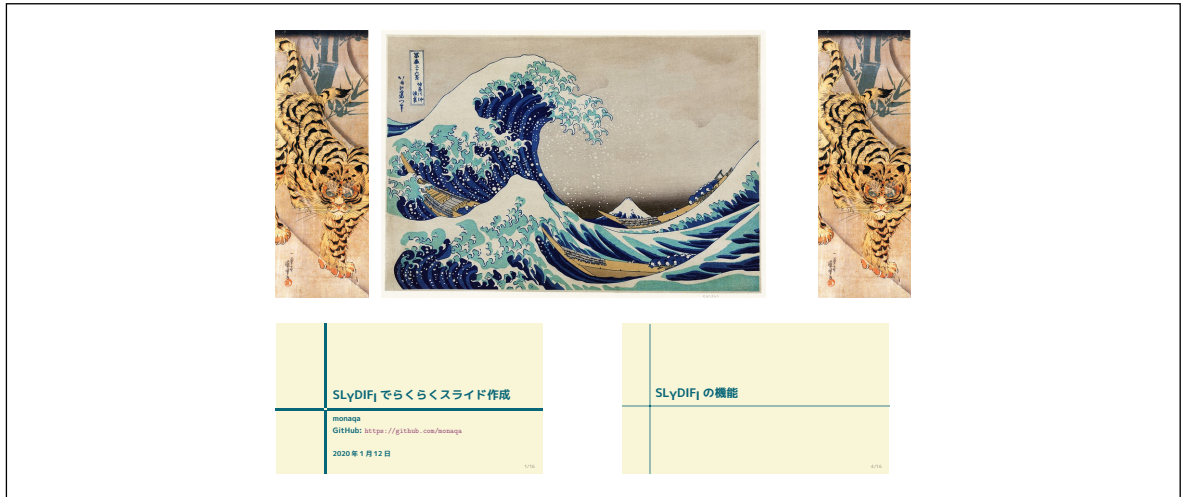
+fig-center(
vconcat[
hconcat [
include-image-with-height 100pt `fig/example2.jpg`;
include-image-with-height 100pt `fig/example1.jpg`;
include-image-with-height 100pt `fig/example2.jpg`;
];
hconcat [

```

```

include-image 100pt `fig/example.pdf`;
include-image ? :4 100pt `fig/example.pdf`;
];
]
);

```

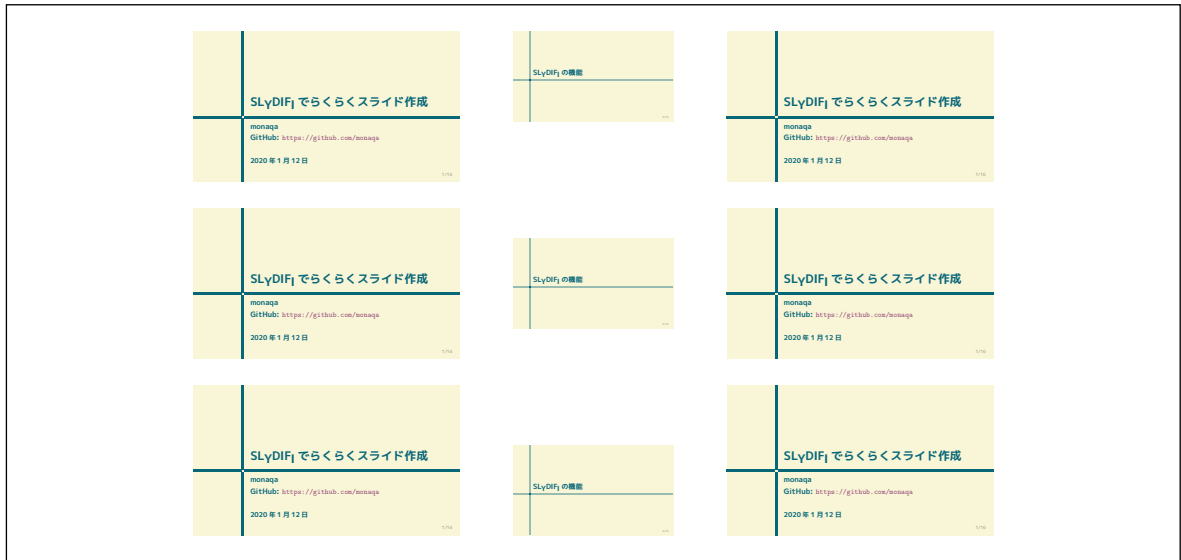


オプション引数で `align-left`, `align-center`, `align-right`, `align-top`, `align-bottom` の5種類のいずれかを指定することで、揃え方を変えることができます。

```

+fig-center(
  let img1 = include-image 100pt `fig/example.pdf` in
  let img2 = include-image ? :4 60pt `fig/example.pdf` in
  vconcat [
    hconcat ? :align-top    [img1; gap 20pt; img2; gap 20pt; img1];
    gap 10pt;
    hconcat ? :align-center [img1; gap 20pt; img2; gap 20pt; img1];
    gap 10pt;
    hconcat ? :align-bottom [img1; gap 20pt; img2; gap 20pt; img1];
  ]
);

```

この機能を使えば、図にキャプションを付けることだって簡単にできますね。

```
+fig-center(
  vconcat[
    include-image 200pt `fig/example.pdf`;
    gap 10pt;
    textbox {図 1: `slydifi` パッケージで組んだスライドの例。};
  ]
);
```



図はフレームで囲むことができます。frame 関数は length -> color -> figbox -> figbox 型を持ち、与えられた線幅・線色で、与えられた figbox を囲みます。

```
+fig-center(
  frame 1pt Color.black (include-image 100pt `fig/example1.jpg`)
);
```



もう少し外側に枠を付けたい、と感じたかもしれません。そんなときには図にマージンをつけるのが良いでしょう。vmargin は上下に、hmargin は左右に、hvmargin は上下左右にマージンをつけることができます。

```
+fig-center(
  frame 1pt Color.black
    (hvmargin 5pt (include-image 100pt `fig/example1.jpg`))
);
```



なお、SATySF_i 標準で用意されているパイプライン演算子 `|>` を用いると以下のように書くこともできます。

```
+fig-center(
  include-image 100pt `fig/example1.jpg`
  |> hvmargin 5pt
  |> frame 1pt Color.black
);
```

このようにすればカッコのネストを軽減できるだけでなく、

1. `fig/example1.jpg` を横幅 100pt で読み込む
2. 上の図の上下左右に 5pt のマージンを追加する
3. 上の図を幅 1pt の黒線で囲む

と、直感的な順番で処理を記述することができます。

その他、図の背景色を指定する `bgcolor` などの関数も用意されています。こちらもやはり `margin` 系の関数でマージンを指定しつつ使うのが良いでしょう。`figbox` を回転させる `rotate` 関数もあります。テキストや画像を回転させたいときに重宝するでしょう。

3.3. `figbox` を埋め込むコマンド

ここまでの例では `figbox` を実際にインラインテキストやブロックテキストに埋め込むために全て `fig-center` を使っていましたが、実際には他にもいくつかのコマンドがあります。

`+fig-block` は `fig-center` と似ていますが、オプション引数に `align-left`, `align-center`, `align-right` を取り、揃える位置を選ぶことができます。

```
+fig-block ?:(align-left) (  
  include-image-with-height 100pt `fig/example2.jpg` |> rotate 90.  
);  
+fig-block ?:(align-center) (  
  include-image-with-height 100pt `fig/example2.jpg` |> rotate 90.  
);  
+fig-block ?:(align-right) (  
  include-image-with-height 100pt `fig/example2.jpg` |> rotate 90.  
);
```



+fig-abs-pos は