

# SLYDIFI デモ

Your Name

2020 年 9 月 12 日

# Example Section

# 普通のスライド

まずはコマンドを使いこなそう.

- +p で段落を作成できる
- +listing で箇条書きを作成できる
  - ネストも可能

続いて, インラインマークアップを使いこなそう.

- \emph で強調
- \textbf で太字 (太字のフォント設定が必要)
- \text-color でテキストの色を変更
- \uline で下線
- \strike で打ち消し線

# 図の挿入

FigBox モジュールの関数・コマンドで自由に図を挿入しよう.

```
+fig-center(include-image 300pt `../fig/slydifi-test.jpg`);
```



その他にも様々な関数・コマンドが用意されている. 様々な関数を使いこなせば, 次ページにあるような多少複雑な配置も構造的に配置可能.

# テーマの選択

デフォルトで3種類用意されている。一応、自作も可能。

いきなり自作は少し大変なので、まずは既存のヘッダファイルをコピーしてフォント設定などを変更する所から始めるといい。

## SLyDIF<sub>I</sub>: SATySF<sub>I</sub> のスライド作成用パッケージ

- **SATySF<sub>I</sub>** (<https://github.com/gfngfn/SATySF<sub>I</sub>>)
  - 静的型付き関数型言語ベースの新たな組版処理システム
  - 普通の文書作成時にはそこまで関数型言語を意識しなくてよい
  - パッケージの記述に用いる構文は OCaml 風
- **SLyDIF<sub>I</sub>** (<https://github.com/monaga/slydifi>)
  - SATySF<sub>I</sub> でスライドを作成することができる
  - 実は既に先駆者がいる (*Steamer: Slide Presentation in SATySF<sub>I</sub>*)
    - Steamer とは page break 周りの実装を少しばかり変えている

3/8

## Plain theme

## SLyDIF<sub>I</sub>: SATySF<sub>I</sub> のスライド作成用パッケージ

- ◆ **SATySF<sub>I</sub>** (<https://github.com/gfngfn/SATySF<sub>I</sub>>)
  - ▶ 静的型付き関数型言語ベースの新たな組版処理システム
  - ▶ 普通の文書作成時にはそこまで関数型言語を意識しなくてよい
  - ▶ パッケージの記述に用いる構文は OCaml 風
- ◆ **SLyDIF<sub>I</sub>** (<https://github.com/monaga/slydifi>)
  - ▶ SATySF<sub>I</sub> でスライドを作成することができる
  - ▶ 実は既に先駆者がいる (*Steamer: Slide Presentation in SATySF<sub>I</sub>*)
    - Steamer とは page break 周りの実装を少しばかり変えている

3/8

## Hakodate theme

## SLyDIF<sub>I</sub>: SATySF<sub>I</sub> のスライド作成用パッケージ

- ◆ **SATySF<sub>I</sub>** (<https://github.com/gfngfn/SATySF<sub>I</sub>>)
  - ▶ 静的型付き関数型言語ベースの新たな組版処理システム
  - ▶ 普通の文書作成時にはそこまで関数型言語を意識しなくてよい
  - ▶ パッケージの記述に用いる構文は OCaml 風
- ◆ **SLyDIF<sub>I</sub>** (<https://github.com/monaga/slydifi>)
  - ▶ SATySF<sub>I</sub> でスライドを作成することができる
  - ▶ 実は既に先駆者がいる (*Steamer: Slide Presentation in SATySF<sub>I</sub>*)
    - Steamer とは page break 周りの実装を少しばかり変えている

3/8

## Akasaka theme

# オーバーレイ

SLYDIF<sub>1</sub>では簡素なオーバーレイ機能を搭載している.

# オーバーレイ

このスライドもオーバーレイを使っています

SLYDIF<sub>1</sub>では簡素なオーバーレイ機能を搭載している.

オーバーレイ機能を使うと, スライドのコンテンツを順に出現させたり, スライドの一部を途中で強調したりすることが可能となる. 例えば ...

# オーバーレイ

SLyDIF<sub>1</sub>では簡素なオーバーレイ機能を搭載している.

オーバーレイ機能を使うと, スライドのコンテンツを順に出現させたり, スライドの一部を途中で**強調**したりすることが可能となる. 例えば ...

```
+p?:(only 4){  
  `+p` や `\emph` といったコマンドでは,  
  オプション引数に `int -> bool` 型の関数を入れる.  
}
```



# オーバーレイ

このスライドもオーバーレイを使っています

SLyDIF<sub>1</sub>では簡素なオーバーレイ機能を搭載している.

オーバーレイ機能を使うと, スライドのコンテンツを順に出現させたり, スライドの一部を途中で強調したりすることが可能となる. 例えば ...

```
+p?:(only 4){  
  `+p` や `\emph` といったコマンドでは,  
  オプション引数に `int -> bool` 型の関数を入れる.  
}
```

+p や \emph といったコマンドでは, オプション引数に int -> bool 型の関数を入れる.