

# SLYDIF<sub>I</sub> デモ

Your Name

2020 年 9 月 12 日

# Section Name

# 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．

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，次ページの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。自作も可能。

ただしいきなり自作はちょっと大変なので、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ことから初めてみよう。

## 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/monaqa/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/monaqa/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

Test

- ◆ 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/monaqa/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