

SLyDIF_I でらくらくスライド作成

monaqa

GitHub: <https://github.com/monaqa>

2020 年 1 月 12 日

はじめに

SLyDIF_I: SATySF_I のスライド作成用パッケージ

- ◆ **SATySF_I** (<https://github.com/gfngfn/SATySFi>)
 - ▶ 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_I** (<https://github.com/monaqa/slydifi>)
 - ▶ SATySF_I 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 SATySFi](#))
 - Steamer とは page break 周りの実装を少しばかり変えている

SLyDIF_I の機能

フレーム作成

- ◆ フレーム：スライド資料の 1 ページ 1 ページに値するもの
- ◆ SLyDIF₁ では 3 種類のフレームを区別する
 - ▶ 見出し：スライド全体の題目，発表者名などを載せるフレーム
 - ▶ セクション見出し：セクションのタイトルを載せる
 - ▶ 本文：通常のフレーム

テキストの記述

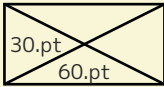
以下のようなコマンドを用いてテキストを記述できる.

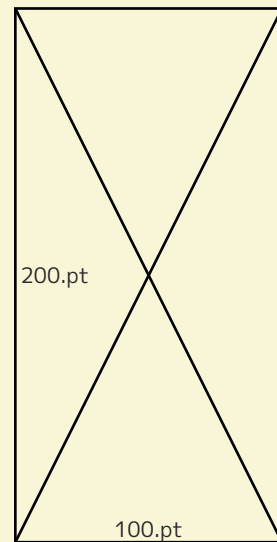
- ◆ `+p{}`: 段落
- ◆ `+listing{}`: 番号のない箇条書き
- ◆ `+enumerate{}`: 番号付きの箇条書き

さらに, インラインテキストの中では以下のマークアップが使える.

- ◆ `\emph{}`: **強調**
- ◆ `\textbf{}`: **太字**
- ◆ `\text-color(){}:` **文字色変更**
- ◆ `\link?:({Hyper Link})(`url`):` **ハイパーリンク**

図表の貼り付け (FigBox モジュール)

- ◆ 例 : `+fig-center(FigBox.include-image 200pt `path/to/image.jpg`);`
 - ▶ `FigBox.include-image` : 画像 (PDF/JPEG) を指定幅で読み込む
 - ▶ `+fig-center` : 読み込んだ図を中央揃えで配置
- ◆ その他にも様々な読み込み用の関数や配置コマンドが用意されている
 - ▶ `dummy-box` : 指定されたサイズのダミーボックス
 - ▶ `hmargin` : 水平方向に指定された量の余白を付ける
 - ▶ `\fig-inline` : テキスト中  に画像を出力
 - ▶ `+fig-on-right` : 画像を右に, 本文を左に配置



SLyDIF_I のテーマのカスタマイズ