

# Markdownで作るスライド

藤原 惟

藤原 由来

Twitter: sky\_y

# Markdownスライドの現状

- reveal.js (HTMLスライド)が最強
  - Markdownからreveal.jsに変換する手段がたくさん
  - Webベースでもreveal.jsバックエンドが多数

# Markdownスライド: Pandoc

- reveal.jsに変換するのがベター
- Pandoc 2.0.5でPowerPoint出力に対応
  - 今動いているのがPowerPointです
  - デザインの調整は追って調査します

# Pandoc: pptxのデザイン

- 画面比の指定 (YAML metadataの `aspectratio`) → 無効
- `$ pandoc --print-default-data-file reference.pptx > reference.pptx`
  - `reference.pptx`をカスタマイズしてデザインを設定できるはずだが . . .
  - PowerPoint

# Markdownスライド: MPE

- [Markdown Preview Enhanced](#) (MPE)
  - VS Code/Atomのプラグインとして利用可能
  - すごく高機能なMarkdown処理系
  - 個人的にはVS Codeで使っています
- [mume](#)
  - MPEのコアにあたる変換エンジン (npmパッケージ)
  - 変換処理をスクリプト化するときには便利
- GitHub Pages: 静的ホスティング

# 参考：スライドのGitHubリポジトリ

- <https://github.com/sky-y/histudy-201712>
- GitHubリポジトリさえ設定できてれば、基本的には`npm run all`でスライド生成→デプロイまで完結します
- あとで整備して、ボイラープレートとして使えるようにしたい

# 執筆お助けツール

## – [textlint](#)

- 英語・日本語の文法誤りをしてくれるツール
- プラグインでスタイルやルールをインストールできる
- [Collection of textlint rule](#)・[textlint/textlint Wiki](#)

## – [prh](#)

- textlintと組み合わせて使う
- 辞書を作っておくと、用語の誤りを指摘してくれる
- [サンプルルール](#)のtechboosterだけでもかなり使える

# 執筆のためのCI

## – 技術書界隈でCIが流行ってる

- 原稿をPush→完成品に近いPDF生成  
→GitHub/BitBucketのIssue/PRでフィードバック
- [技術書同人誌を書きましょう！ - Qiita](#)

## – 執筆CIのメリット

- 原稿を書いた片っ端から編集者さんが随時校正できる
  - 執筆フローは、伝統的にはウォーターフローモデル
- リモートでの共同執筆やチームプレイが円滑になる



# 執筆に関するおすすめ資料

- － 【もくもく執筆会】執筆お疲れさま！歓談&LTパーティー
  - ・ 資料一覧 に上がってるLT資料が良い

おわり