

GITHUB PAGESでプレゼンテーション

DEMO : このプレゼンテーションのリポジトリ

作り方

1. まず、普通にMarkdown文書を作る
2. `$ mdpress readme.md`でプレゼンテーションが *readme* フォルダに作られる
3. originブランチにpush
4. Setting -> GitHub Pagesで空のGitHub Pagesをつくる
5. gh-pagesブランチがGitHub上にできる
6. pullするとgh-pagesブランチもローカルに付いてくる
7. `$ git checkout gh-pages`
8. `$ cp -rf readme/* .`
9. `$ git push -u origin gh-pages` (最初だけ)
10. `$ git push` (2回目以降)

良いところ

- 改訂したとき、diffが見れる
- プルリクも送ることができる
- ソース(Markdown)とプレゼンテーション(HTML)を同じ一つのリポジトリで管理できる
- Qiitaやはてなブログのように、画像を一枚一枚アップしなくて良い
- LT後の質疑などの内容を、筆者だけでなく聞いた人もプルリクで追記できる

悪いところ

- 更新がややこしい

更新のやり方

1. [origin] まずoriginブランチであるか確認
2. [origin] Markdownに追記
3. [origin] Markdownエディタを終了
4. [origin] git push
5. [origin] mdpressでリポジトリとは別のフォルダにプレゼン作成
6. [gh-pages] gh-pagesブランチに切り替える
7. [gh-pages] プレゼンをリポジトリにコピー
8. [gh-pages] git push
9. [origin] originブランチに戻しておく

スクリプト化してみる：

PUSH.SH

```
# レポジトリに入る
# フォルダ名は引数にしたい
cd 140127-2013-soukatsu-2014-houshin

# Markdownをpush
git add .
git commit -m "committed automatically by push.sh"
git push

# mdpressコマンドでreadmeフォルダを生成
cd ..
mdpress 140127-2013-soukatsu-2014-houshin/readme.md
```

gh-pages ブランチに切り替える

```
cd 140127-2013-soukatsu-2014-houshin
```

```
git checkout gh-pages
```

先ほど生成した readme フォルダの中身をレポジトリにコピーする

```
cp -rf ../readme/* .
```

自動的に commit + push

```
git add .
```

```
git commit -m "committed automatically by push.sh"
```

```
git push
```

origin ブランチに戻る

```
git checkout master
```

元いたディレクトリに戻る

```
cd ..
```


めんどくさい

GULP-GH-PAGES



(作者のMicheal Benedictさん。Twitter社に勤務。)

```
$ gulp deploy
```

これで全部やってくれます

ひえー

さて、プレゼンが終わって

- 作ったプレゼンをSpeakerDeckやSlideShareにアップしたい
- 今までは・・・HTMLページを一枚ずつPDFにして、
- あとでたばねて一つのPDFファイルにしていた

めんどくさい

DECK2PDF



(作者のCedricさん。フランスからPivotal社にリモート勤務)

```
$ deck2pdf --profile=impressjs index.html
```

これで1つのPDFファイルにしてくれます

ひえー

まとめ

- mdpress
- mdpress-generator
- gulp-gh-pages
- deck2pdf

．．．で、MarkdownでGitHubな
プレゼンテーションライフを実現しましょう
ご清聴ありがとうございました

