

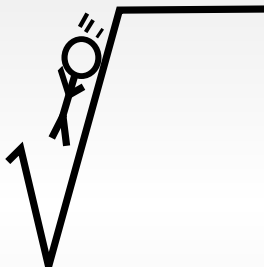
# Markdown to L<sup>A</sup>T<sub>E</sub>X

金子達哉 (id:catatsuy)

April 9, 2013

# 自己紹介

- 金子達哉
- はてな ID: catatsuy
- twitter: catatsuy



URL:

- <http://www.catatsuy.org>
- <http://blog.catatsuy.org>
- <https://matw.co>

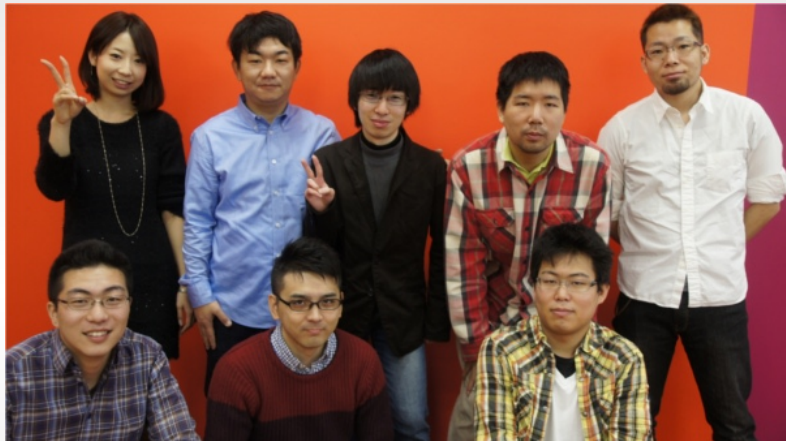
# 所属

- 東京工業大学
- 情報工学科 4 年（9 月卒業予定）
- 吉瀬研究室
  - コンピュータアーキテクチャ
  - C++ でハードウェアのシミュレーションを高速化する研究しています（詳しくは長くなるので省略）

# 就職活動

- はてなインターン 2012
- pixiv インターン



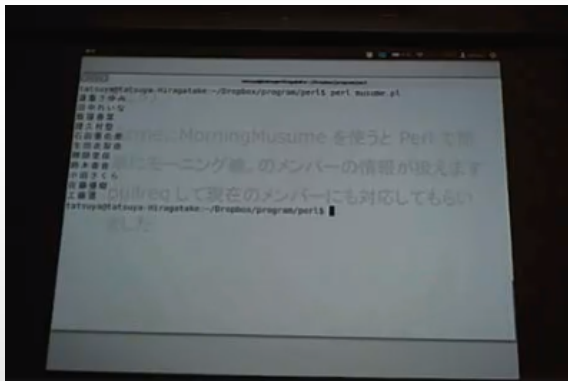


10 月から [pixiv](#) へ！

# 前回の Dentoo.LT

Acme::MorningMusume の話をしました

<http://blog.catatsuy.org/a/256>



卒業したい

卒業論文

(8 月提出のため)

絶賛追い込みなう☆



卒論といえば...

I ♥ L<sup>A</sup>T<sub>E</sub>X

# L<sup>A</sup>T<sub>E</sub>X

- `\verb/gcc/` とか書くの面倒
- `\begin{}` - `\end{}` で囲うの面倒
- 最初のプリアンブルがうざい

時代は軽量マークアップへ

L<sup>A</sup>T<sub>E</sub>X

Markdown

⇒

wiki 記法

HTML

はてな記法

# I ♡ Markdown

- Github などで採用
- 電子メールからの装飾から着想
  - 海外の人には直感的らしい
- シンプルに書ける
- 各言語でパーサーが実装されている

今回のタイトル

**Markdown to L<sup>A</sup>T<sub>E</sub>X**

# Pandoc

- ・ マークアップ言語の相互変換ツール
- ・ 関数型言語 haskell で実装
- ・ 機能ごとに綺麗にモジュール化されている
- ・ 多彩なフォーマットに対応
- ・ Markdown  $\Rightarrow$   $\text{\LaTeX}$  も可能

# Pandoc インストール

Ubuntu:

```
sudo apt-get install haskell-platform
```

Mac:

```
brew install ghc
```

```
brew install haskell-platform
```

common:

```
cabal update
```

```
cabal install pandoc
```

# ~/.cabal/bin/ ディレクトリ以下に PATH を通す

# pandoc の使い方

# 本文のみ

```
pandoc input.md -o output.tex
```

# テンプレート込み

```
pandoc input.md -s -o output.tex
```

# beamer (プレゼン) 出力

```
pandoc -t beamer input.md -o output.texw
```

# 変数指定

```
pandoc -V fontsize=12Q input.md -o output.tex
```



# pandoc の問題点

- ‘gcc’ と書くと `\texttt{...}` にされてしまう
- 本当は `\verb+...+` とかにして欲しい
- テンプレートが日本の  $\text{\LaTeX}$  向けではない

# Github で Fork しました！

```
git clone git://github.com/catatsuy/pandoc.git
cd pandoc
git submodule init
git submodule update
# git://github.com/catatsuy/pandoc-templates.git
# が data/templates 以下に clone される
```

# ‘ ‘ の挙動を変える

src/Text/Pandoc/Writers/LaTeX.hs

```
- rawCode = liftM (text . (\s -> "\\texttt{" ++ s ++ "}))  
-           $ stringToLaTeX False str  
+ rawCode = liftM (text . (\s -> "\\verb'" ++ s ++ "'"))  
+           $ stringToLaTeX True str
```

# テンプレート

- 別リポジトリ (git submodule)
- \$...\$ で変数展開
  - 変数は -V オプションで渡す

# テンプレート作成ポイント

- ・読み込むパッケージなどは最小限に
- ・`--include-in-header header.tex` として追加パッケージや余白設定などを別ファイルにできる

# cabal-dev

- cabal は ~/.cabal/ 以下にインストールする
- すでに本家の pandoc はインストール済み
- カレントディレクトリ上でコンパイルしたい

```
cabal install cabal-dev
```

```
cabal-dev install --sandbox=.
```

```
# ./bin/ 以下に実行ファイルが出力される
```

# 使ってみて分かった問題点

- 少しでも複雑なものは  $\text{\LaTeX}$  で書く必要
- Emacs の色分けが  $\text{\LaTeX}$  部分で効かない
- yatex の強力な補完機能が使えない
- 改行したところでスペースが入ることがある

# 改善案

- Markdown で  $\text{\LaTeX}$  の文章を書くのではなく  $\text{\LaTeX}$  の文章上で Markdown 記法を部分的に使うべき
- スペースが入っても問題のない所で改行する

# 長いので Makefile 作成推奨

```
pandoc -f markdown input.tex -o output.tex
```

```
# yatex はデフォルトで自動改行してしまうので .emacs に追加
(add-hook ' yatex-mode-hook '(lambda ()
                                (auto-fill-mode -1)))
```



# 改善案の長所

- これが最適解っぽい
- $\text{\LaTeX}$  ファイルがすっきりする
- `yatex` も使える
- Markdown に補完や色分けいらない

# 欠点

- 出力を意識しながら書く必要
- `\\` が書けないので `\linebreak` や `\newline` などを使う必要
- $\text{\LaTeX}$  を純粹に書くなら生じない無駄な悩みが発生すること

# サンプル

今回の一連の流れを再現するサンプル

`github.com/catatsuy/mdtolatex_sample`

Makefile など参考にしてください

# 今後の予定

- 卒論を Markdown で書きます
- その上で分かった問題を共有していきます
- Markdown で卒論書きましょう

まとめ

| ♡ L<sup>A</sup>T<sub>E</sub>X

| ♡ Markdown

| ♡ 卒論