

Qiita と Re:VIEW を使ってオンライン で記事を寄せあい同人誌を作って 同時 に Web コンテンツとして公開できるシ ステムがこんなに便利なわけではない

日本 Android の会秋葉原支部ロボット部 著

2017-04-09 版 発行

はじめに

この本の流れ

まず、第 1 章は本書で作った qiita2review の使い方、第 2 章は Qiita 上で数式を含んだ記事の書き方、第 3 章、第 4 章は組版システム Re:VIEW を使った例、第 5 章、第 6 章は Qiita との連携を記しています。

Web 連携

この記事は Web でも読めます。Qiita で公開していますので印刷が見づらいところは参照して下さい。

コードの公開について

公開まで手が回りませんでした。落ち着いたら Public Domain として公開しますので twitter @nanbuwks で検索するか、お問い合わせください。

お詫び

システムを作るのが長引いて、書式を調整する時間がありませんでした。コードがは見えたりしますが、Web で補完ください。



2017/04/10 03:31

目次

はじめに	i
この本の流れ	i
Web 連携	i
コードの公開について	i
お詫び	i
第 1 章 qiita2review の使い方	1
1.1 qiita2review とは	1
1.2 管理者がはじめにやること	2
1.3 執筆者がやること	2
Qiita に記事を書く	2
Qiita に記事が書けたら	2
1.4 原稿が集まったら	3
第 2 章 Qiita で数式を書きましょう	4
2.1 TeX 式を簡単に作成する	4
Qiita で書く	4
第 3 章 AWS 上に Re:VIEW 環境を構築する	6
3.1 環境	6
3.2 インストール	6
第 4 章 Re:VIEW+markdown で数式の入った薄い本を書く	10
4.1 初期化	10
4.2 雛形	10
4.3 md2review の制限	11
4.4 TeX の機能を使って中央揃えを実現する	11
4.5 画像を縮小するスクリプト	12
4.6 config.yml や cover.jpg を入れ替える	14
4.7 ひたすら書く	14
4.8 Tips	14
数式を書く	14
コードブロック中に <code>//}</code> が書けない	15

markdown と Re:VIEW ファイルの違い	16
第 5 章 組版システム ReVIEW で Qiita から同人誌原稿を作る	17
5.1 markdown から ReVIEW	17
5.2 Qiita で書くメリット	17
5.3 catalog.yml	17
5.4 ひとつの記事を元に原稿を作ってみる	18
5.5 複数の記事を元に原稿を作ってみる	19
5.6 画像を縮小する	19
第 6 章 Re:VIEW を apache で動かす	25
6.1 apache と PHP	25
6.2 DocumentRoot に review 環境を展開	25
6.3 This account is currently not available.	25
6.4 review 環境を展開します。	26
6.5 テンプレートを作ります。	26

第 1 章

qiita2review の使い方

Re:VIEW を apache で動かす <http://qiita.com/nanbuwks/items/dd15819ec7798a9eca7b> で書いた、qiita から pdf を作るシステム。これを使う前提での Qiita の記事の書き方。



図 1.1 Screenshot from 2017-04-07 09-50-56.png

1.1 qiita2review とは

- Qiita で記事を執筆してもらって Re:VIEW で PDF にするオンラインシステム。
- グループ作業で技術情報をマルチ展開するために。
- Qiita だと学習コスト少、画像楽だし数式も使える。
- PDF 作ると同時にちゃんとした Web コンテンツを公開できる。

1.2 管理者がはじめにやること

サーバにインストール、認証設定、サーバアドレスの執筆者への周知。

1.3 執筆者がやること

Qiita に記事を書く

後々 PDF にするために、ちょっと気をつける点。

画像

画像はそのままだと 100% になり、紙媒体では大きすぎることが多い。縮小設定をしておく。通常の画像は

```
![ファイル名](images/.....8d78.jpeg)
```

のようになってますが

```
[] ( scale=0.5 )![ファイル名](https://qiita-image-store.s3.amazonaws.com/0/.....8d78.jpeg)
```

のように頭に `scale=0.5`) をつけると、Qiita では 100%, PDF にしたときには 50% サイズになります。

Re:VIEW の制限

PDF 化に使用している Re:VIEW 受け付けられない書式にならないように注意- コメントの入れ子- コードブロック開始前に改行を入れる

コードブロック開始前に改行がない場合、Markdown としてもヘンになることが多いので改行を入れる習慣をつけよう。

Qiita に記事が書けたら

PDF 化の確認をします。

qiita2review サーバページから、記事一覧が見えます。

画面下部の「Add new article title」のフォームに入力して送信すると新しい記事が登録できる。

(認証が必要)

Qiita の 1 記事ごとに PDF になる。別刷りのようなイメージ。

1.4 原稿が集まったら

本としての装丁は管理者が行います。1 記事になっているものはそれぞれ章にして、まとめて 1 冊としてレンダリング 印刷管理者のお仕事となります。今のところはサーバに ssh ログインして Re:VIEW を使って手作業です。

第2章

Qiita で数式を書きましょう

河野悦昌 オープンフォース

2.1 TeX 式を簡単に作成する

<https://webdemo.myscript.com/views/math.html#> ここで数式を手書きで書くと TeX 式に直してくれます

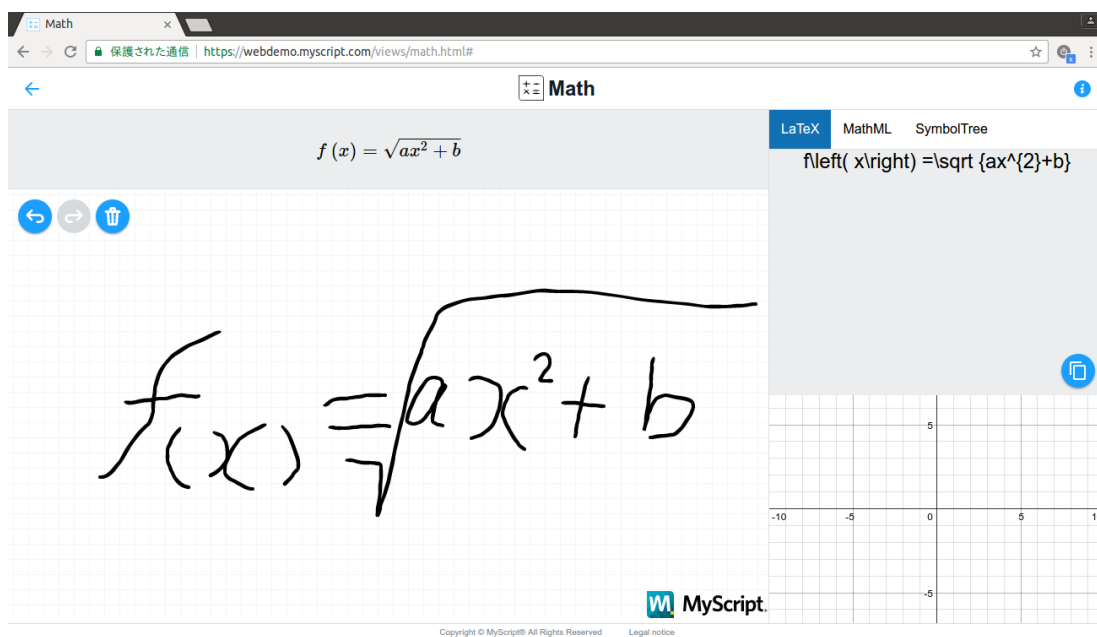


図 2.1 image

Qiita で書く

先のページで作った LaTeX 式をコピーします。


```
f\left( x\right) =\sqrt {ax^{2}+b}
```

この数式を Qiita で書いてみます。

「 TeX で作った式 $f\left(x\right) =\sqrt {ax^{2}+b}$ です 」

と書くと

「 TeX で作った式 $f(x) = \sqrt{ax^2 + b}$ です 」

となります。

ブロック書式で書くには

“ math

「 TeX で作った式 $f\left(x\right) =\sqrt {ax^{2}+b}$ です 」

“

と書くと

「*TeX* で作った式 $f(x) = \sqrt{ax^2 + b}$ です」

となります。

第3章

AWS 上に Re:VIEW 環境を構築する

AWS 上に Re:VIEW 環境を構築してみました。

3.1 環境

- T2.micro
- Ubuntu 16.04 64bit (Ubuntu 14.04 から dist-upgrade をかけています。) 既に ruby や rack は別の用事でインストール終わっていました。

3.2 インストール

```
gem install review
gem install md2review
```

試してみます

```
review-init testwrite
cd testwrite
rake pdf
```

エラー

```
review-pdfmaker config.yml
compiling testwrite.tex
/var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdfmaker.rb:42:in 'system_or_raise': failed to run command
  from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdfmaker.rb:276:in 'block in copy_images'
  from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdfmaker.rb:269:in 'chdir'
  from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdfmaker.rb:269:in 'copy_images'
```

```

from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdftmaker.rb:235:in 'generate_pdf'
from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdftmaker.rb:132:in 'execute'
from /var/lib/gems/2.3.0/gems/review-2.2.0/lib/review/pdftmaker.rb:86:in 'execute'
from /var/lib/gems/2.3.0/gems/review-2.2.0/bin/review-pdftmaker:18:in '<top (required)>'
from /usr/local/bin/review-pdftmaker:22:in 'load'
from /usr/local/bin/review-pdftmaker:22:in '<main>'
rake aborted!
Command failed with status (1): [review-pdftmaker config.yml...]
/home/ubuntu/testwrite/Rakefile:60:in 'block in <top (required)>'
/var/lib/gems/1.9.1/gems/rake-11.2.2/exe/rake:27:in '<top (required)>'
Tasks: TOP => pdf => book.pdf
(See full trace by running task with --trace)

```

Re:VIEW は 2.0 からは uplatex を使うらしいので、インストールします。

```

$ apt-cache search uplatex
texlive-lang-cjk - TeX Live: Chinese/Japanese/Korean

```

ということなので、

```

$ sudo apt-get install texlive-lang-cjk
sudo: unable to resolve host ip-172-30-0-222
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  fontconfig-config fonts-dejavu-core fonts-ipaexfont-gothic
  fonts-ipaexfont-mincho fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-lmodern ghostscript gsfonts ko.tex-extra-hlfont latex-beamer
  latex-cjk-all latex-cjk-chinese latex-cjk-chinese-arphic-bkai00mp
  latex-cjk-chinese-arphic-bsmi00lp latex-cjk-chinese-arphic-gbsn00lp
  latex-cjk-chinese-arphic-gkai00mp latex-cjk-common latex-cjk-japanese
  latex-cjk-japanese-wadalab latex-cjk-korean latex-cjk-thai latex-xcolor
  libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcups2
  libcupsfilters1 libcupsimage2 libdatatrie1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libfile-basedir-perl libfile-desktopentry-perl
  libfile-mimeinfo-perl libfontconfig1 libfontenc1 libgl1-mesa-dri
  libgl1-mesa-glx libglapi-mesa libgraphite2-3 libgs9 libgs9-common
  libharfbuzz0b libice6 libijs-0.35 libjbig0 libjbig2dec0 libjpeg-turbo8
  libjpeg8 libkpathsea6 liblcms2-2 liblvm3.4 libpaper-utils libpaper1
  libpciaccess0 libpixmap-1-0 libpoppler44 libptexenc1 libsm6 libtcl8.6
  libtiff5 libtk8.6 libtxc-dxtn-s2tc0 libutempter0 libx11-xcb1 libxaw7
  libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-render0
  libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcomposite1 libxcursor1
  libxdamage1 libxf86vm3 libxft2 libxi6 libxinerama1 libxmu6 libxpm4
  libxrandr2 libxrender1 libxshmfence1 libxss1 libxt6 libxtst6 libxv1
  libxxf86dga1 libxxf86vm1 lmodern luatex pgf poppler-data preview-latex-style
  prosper ps2eps ruby swath tcl tcl8.6 tex-common texlive-base
  texlive-binaries texlive-extra-utils texlive-font-utils

```

```

texlive-generic-recommended texlive-lang-other texlive-latex-base
texlive-latex-base-doc texlive-latex-extra texlive-latex-extra-doc
texlive-latex-recommended texlive-latex-recommended-doc texlive-luatex
texlive-pictures texlive-pictures-doc texlive-pstricks texlive-pstricks-doc
tk tk8.6 x11-common x11-utils x11-xserver-utils xbitmaps xdg-utils xterm
Suggested packages:
ghostscript-x hpijs hbf-cns40-b5 hbf-jfs56 fonts-arphic-bkai00mp
fonts-arphic-bsmi00lp fonts-arphic-gbsn00lp fonts-arphic-gkai00mp
hbf-kanji48 cups-common libglide3 fonts-droid liblcms2-utils poppler-utils
fonts-arphic-ukai fonts-arphic-uming fonts-unfonts-core pdf-viewer
postscript-viewer ri ruby-dev libthai-data tcl-tclreadline debhelper perl-tk
chktex fragmaster xindy latexdiff lacheck latexmk dvidvi purifyeps dvipng
t1utils psutils latex-fonts-sipa-arundina libfile-which-perl python-pygments
dot2tex libtcltk-ruby mesa-utils nickle cairo-1.5 xorg-docs-core gvfs-bin
xfonts-cyrillic
Recommended packages:
wish
The following NEW packages will be installed:
fontconfig-config fonts-dejavu-core fonts-ipaexfont-gothic
fonts-ipaexfont-mincho fonts-ipafont-gothic fonts-ipafont-mincho
fonts-lmodern ghostscript gsfonts ko.tex-extra-hlfont latex-beamer
latex-cjk-all latex-cjk-chinese latex-cjk-chinese-arphic-bkai00mp
latex-cjk-chinese-arphic-bsmi00lp latex-cjk-chinese-arphic-gbsn00lp
latex-cjk-chinese-arphic-gkai00mp latex-cjk-common latex-cjk-japanese
latex-cjk-japanese-wadalab latex-cjk-korean latex-cjk-thai latex-xcolor
libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcups2
libcupsfilters1 libcupsimage2 libdat1 libdrm-intel1 libdrm-nouveau2
libdrm-radeon1 libfile-basedir-perl libfile-desktopentry-perl
libfile-mimeinfo-perl libfontconfig1 libfontenc1 libgl1-mesa-dri
libgl1-mesa-glx libglapi-mesa libgraphite2-3 libgs9 libgs9-common
libharfbuzz0b libice6 libijs-0.35 libjbig0 libjbig2dec0 libjpeg-turbo8
libjpeg8 libkpathsea6 liblcms2-2 libllvm3.4 libpaper-utils libpaper1
libpciaccess0 libpixmap-1-0 libpoppler44 libptexenc1 libsm6 libtcl8.6
libtiff5 libtk8.6 libtxc-dxtn-s2tc0 libutempter0 libx11-xcb1 libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-render0
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcomposite1 libxcursor1
libxdamage1 libxfixes3 libxft2 libxi6 libxinerama1 libxmu6 libxpm4
libxrandr2 libxrender1 libxshmfence1 libxss1 libxt6 libxtst6 libxv1
libxxf86dga1 libxxf86vm1 lmodern luatex pgf poppler-data preview-latex-style
prosper ps2eps ruby swath tcl tcl8.6 tex-common texlive-base
texlive-binaries texlive-extra-utils texlive-font-utils
texlive-generic-recommended texlive-lang-cjk texlive-lang-other
texlive-latex-base texlive-latex-base-doc texlive-latex-extra
texlive-latex-extra-doc texlive-latex-recommended
texlive-latex-recommended-doc texlive-luatex texlive-pictures
texlive-pictures-doc texlive-pstricks texlive-pstricks-doc tk tk8.6
x11-common x11-utils x11-xserver-utils xbitmaps xdg-utils xterm
0 upgraded, 133 newly installed, 0 to remove and 0 not upgraded.
Need to get 852 MB of archives.
After this operation, 1,625 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

さて、どうかな？

```
$ rake pdf
.
.
.
<to be read again>
      relax
1.100 \fontencoding\encodingdefault\selectfont
?
```

と出たので、フォントをインストール

```
$ sudo apt-get install texlive-fonts-recommended
```

として、`@<tt>{ rake pdf }` とすると、`book.pdf` ができました。

第 4 章

Re:VIEW+markdown で数式の入った 薄い本を書く

河野悦昌 秘密結社オープンフォース

軌道エレベータの本をつくる。いろいろと数式を自分でいじって計算する演習本にしようと思う。
TeX を日本語で使うには人類には早すぎる。なので Re:VIEW でつくる。けれども頑張ろうと思う
と結局 TeX をすることになる。なので人類には早すぎる。

4.1 初期化

orbitcalc というコードネームで作る。review-init orbitcalc

4.2 雛形

解析概論 1943 プロジェクトから最初の 1 ページをコピーしてこれを雛形とする。

```
##基本的ノ概念
###数ノ概念及ビ四則算法
ハ既知ト彼定スル@<fn>{23-1} . 始メノ中ハ實数ノミヲ取扱フカラ
一々断ラナイ . 次ノ用語ハ周知デアル .

**自然数 . ** 1,2,3 等 . 物ノ順位又ハ物ノ集合ノ個数ヲ示ス篤二用ヅラレル .

**整数 . ** 0 , ± 1 , ± 2 等 . 自然数ハ正ノ整数デアル .

**有理数 . **0 及ビ @<m>{\pm \dfrac {a\\} {b\\}}子 , 但 , b ハ自然数 . b=1 ナルトキ , ソレハ整数デ
アル .

**無理数 . **有理数以外ノ實数 . 例ヘバ

//texequation{
\begin{array}{l}
\sqrt {2}=1.4142135\ldots,\\
e=2.718281828\ldots,\\
\end{array}
```

```
pi=3.1415926535...
\end{array}
//}
```

(但, ソレラガ有理数デナイコトハ護明ヲ要スル)

十進法. 實数ヲ十進法デ表ハスコトモ周知デアル. 有理数ヲ十進法デ表ハセバ, 数字ハ有限力, 又ハ無限ナラバ循環小数ニナル. 但, 有限位数ノ十進数ヲ循環小数ノ形ニ表ハスコトモ出来ル. 例ヘバ $0.6 = 0.5999\dots$. 無理数ヲ十進法デ表ハスナラバ, 無限ノ位数ヲ要シ, 数字ハ決シテ循環シナイ.

吾々が十進法ニヨツテ数ヲ表ハズニ至ツタノハ, 手指ノ数ニソノ原因ガアルノデアラウガ, 理論上ハ 1 以外ノ任意ノ自然数ヲ基本トシテ, 十進法ト同様ノ方法ニヨツテ, 数ヲ表ハスコトガ出来ル.

特ニ二進法デハ, 数字ハ 0 ト 1 トダケデ足ル. 有理数ヲ二進法デ表ハセバ, 分母ガ 2 ノ巾 $\langle \text{fn} \rangle_{23-2}$ ニナルモノノ外ハ, 循環二進数ニナル.

```
//texequation{
```

```
\left[ 例\right] \dfrac {5} {8}=\dfrac {1} {2}+\dfrac {1} {2^{\{3\}}}=\left( \begin{matrix} 0.101\end{matrix} \right) \backslash
```

```
//footnote[23-1] [附録 (一) ヲ参照.]
```

```
//footnote[23-2] [巾ハ冪ノ假字 (和算ノ用例ニヨル).]
```

これを、orbitcalc.md と名前をつけて保存。

4.3 md2review の制限

md2review は内部で redcarpet なるものを呼び出しているようで、それによる制限？

- リストの入れ子ができない
- markdown で斜体を指定 (**文** ないし **文**) したものが re ファイルだと強調 (**文**) となる

このほか、re ファイルでは引用ブロック (`//quote{ 文 }`) の入れ子をするエラーになるので、二重引用している markdown を md2review して作った re ファイルはエラーが出る。

4.4 TeX の機能を使って中央揃えを実現する

```
//raw[|latex| \begin{center} 中央揃えにしたい文 \end{center} ]
```

```
//raw[|latex| \begin{center} ]
**筆頭著者** 共著者 *所属*
//raw[|latex| \end{center} ]
```

のようにすればいい。しかしながら//raw の部分が markdown ビューの時にゴミとして出てくる。なのでここらへんは TeX コンパイル時にフィルターとして活用するなど。

qiita の場合は以下のようにすると中央揃えとなる。

中央揃え

このようなフォーマットを通すためのフィルタは以下の通り

```
> ‘‘‘math
> \left[ 例\right] \dfrac {5} {8}=\dfrac {1} {2}+\dfrac {1} {2^{3}}=\left( \begin{matrix} 0.101\end{matrix} \right)
> ‘‘‘
```

```
//texequation{
\left[ 例\right] \dfrac {5} {8}=\dfrac {1} {2}+\dfrac {1} {2^{3}}=\left( \begin{matrix} 0.101\end{matrix} \right) \\\
//}
```

インライン命令 ‘a-\$\$

```
$\pm \dfrac {a} {b}$
```

```
@<m>{\pm \dfrac {a\\} {b\\}}
```

4.5 画像を縮小するスクリプト

写真や図も入れる。写真や図を、markdown の段階でプレビューしやすい形で作りたい。通常の書き方

```
![test](images/test.png " " )
```


だけど、このままだと縮小できない。なのでスクリプトで何とかする。`@<tt>{` 通常の書き方 `![テスト](images/test.png " ")` 拡張した書き方 `[scale=0.5]![テスト](./images/test.png " ")` }
md2review を通す

```
通常の書き方
//image[test][テスト]{
//}
拡張した書き方
//[test][テスト][scale=0.5]{
//}
```

となる。これを、

```
通常の書き方
//image[test][テスト]{
//}
拡張した書き方
//image[test][テスト][scale=0.5]{
//}
```

とするスクリプト。

```
#!/usr/bin/env ruby
while line = gets
  line.chomp!
  if md1 = line.match(/^\[.*?\]/)
    if md2 = line.match(/\[/\[/(.+)\]/)
      print md2[0],md1[0],"{\n"
    else
      puts line
    end
  else
    puts line
  end
end
end
```

scalemd.rb

```
#!/usr/bin/env ruby
while line = gets
  line.chomp!
  if md1 = line.match(/^\[.*?\]/)
```

```
if md2 = line.match(/\n\/(.+)\n\/)
  print md2[0],md1[0],"\n"
else
  puts line
end
else
  puts line
end
end
```

使い方@<tt>{ md2review orbitcalc.md | ruby scalemd.rb > orbitcalc.re }

4.6 config.yml や cover.jpg を入れ替える

cover.jpg は 1110x1,840 は 2 ページ目に配置された。1101x1825 だと OK。

4.7 ひたすら書く

(明日から頑張る)

4.8 Tips

数式を書く

Web Equation を使って、手書き数式を TEX に変換してくれる。<https://webdemo.myscript.com/#/demo/equation> 紹介ページ
<http://gigazine.net/news/20120203-latex-mathml-web-equation/>
ここで書いた数式をゲット

```
g=r\omega ^{2}
```

ブロック要素として数式を入れるには

```
//texequation{
g=r\omega ^{2}
//}
```

というように囲めばいい

インライン要素として数式を入れるには

```
@<m>{g=r\omega ~{2\\}}
```

というように、閉じ大括弧の前にエスケープを2つ入れる。人類にはまだ早い。

コードブロック中に `//}` が書けない

markdown 上のコードブロック中には `//}` は問題なくかけるのだけれど、Re:VIEW 形式に変換するとコードブロックがそこで閉じてしまい、エラーとなる。Re:VIEW のブロック構文は `//}` を閉じタグになっているので Re:VIEW の構文を解説する文書が Re:VIEW で書けないということになってしまう。

これは Re:VIEW 形式の仕様の問題かな。

`//embed` ブロック命令はブロック内の文字列をそのまま文書中に埋め込みます。エスケープされる文字はありません。

「Re:VIEW フォーマットガイド」<https://github.com/kmuto/review/blob/master/doc/format.ja.md>

コードブロック中の書き方でどうにかできないかと思ったけれどもエスケープできないのでこれはどうしようもない。仕方がないので、markdown の段階でフィルタを通すことにした。コードブロック中に `//}` が存在するようなら各行頭にスペースを入れるようにした。

```
slash2braceincode.rb
```

```
@<raw>{@<fn>{23-1}
//@<raw>{}}
```

```
@<fn>{23-1}
//}
```

実は下の `//}` は何も出力していない。パーサを邪魔しているだけ。

あたまの `@<tt>{ }` “ ” の前に改行が無いと Re:VIEW では

```
//emlist{
```

ではなく

```
@<tt>{
```

となる。ここにインライン命令が入ると md2review が余計なエスケープをつけてコンパイルできなくなる。

なので改行をつけよう。

また、Re:VIEW の命令に誤解釈されるものを本文に書くとおかしくなる。

markdown と Re:VIEW ファイルの違い

エンターの扱い

markdown	改行となる Re:VIEW	空白となる
----------	---------------	-------

第 5 章

組版システム ReVIEW で Qiita から同人誌原稿を作る

河野悦昌 秘密結社オープンフォース 送信

5.1 markdown から ReVIEW

ReVIEW+markdown で数式の入った薄い本を書く <http://qiita.com/nanbuwks/items/9b00e8012e328de6e440> というのを書きました。

何人かでこういうのやるときに、オンライン投稿するやりかたを考えます。

5.2 Qiita で書くメリット

Qiita ってよくできていて、割とラフな操作でもいい感じに作ってくれます。

- オンラインで投稿、書いてる途中でリアルタイムにレンダリング結果を確認
- 画像ドラッグ&ドロップで投稿できる
- URL とか自動でリンクに変換してくれる

これを使うやり方を考えてみました。

5.3 catalog.yml

章立てはこうなってます

```
PREDEF:

CHAPS:
  - 1.re

APPENDIX:
```

POSTDEF:

ここの、CHAPS に複数人の原稿を登録するという感じです。

5.4 ひとつの記事を元に原稿を作ってみる

Pi board (Single Board Computer) and Allwinner <http://qiita.com/nanbuwks/items/7f5fd126b8581e6d1a74> を元に使ってみます。

まず、

```
review-init piallwinner
cd piallwinner
```

catalog.yml と config.yml を編集しておきます。
図をダウンロードします

```
cd images
wget http://qiita.com/nanbuwks/items/7f5fd126b8581e6d1a74 -k -H -r -l 1 -nH -nd -A png,PNG,jpg,JPG,jpeg,JPE
cd ..
```

markdown ファイルをダウンロードします。

```
wget http://qiita.com/nanbuwks/items/7f5fd126b8581e6d1a74.md
```

img ファイルをローカルに変換します。

```
ruby qiitamd.rb 7f5fd126b8581e6d1a74.md > 1.md
md2review 1.md > 1.re
rake pdf
```

変換するスクリプト qiitamd.rb です

```
#!/usr/bin/env ruby

while line = gets
```

```

line.chomp!
if md1 = line.match(/(?!\[.??\]\()/)
  if md2 = line.match(/([^\/*\]\()/)/)
##    puts line
##    print md1[0], "\n"
##    print md1[1], "\n"
##    print md2[0], "\n"
##    print md2[1], "\n"
##    print "\n"
##    print md2[0], md1[0], "{\n"
    print md1[0]
    print "images/"
    print md2[1]
    print "\n"
  else
    puts line
  end
else
  puts line
end
end

```

5.5 複数の記事を元に原稿を作ってみる

```

PREDEF:

CHAPS:
- 1.re
- 2.re
- 3.re

APPENDIX:

POSTDEF:

```

として、2.re や 3.re に上のような原稿を当てはめるといいでしょう。

5.6 画像を縮小する

Qiita は画像をアップロードすると、はみ出る画像は横幅 100% でビューするように画面を作ってくれます。紙媒体の場合は多くは 100% だと良くないですね。50% ぐらいの大きさを使うようにするには？

通常の画像をアップロードすると以下のような markdown となります。@<tt>{ ![ファイル名](images/c19f0873-8b39-9405-1486-8cfd43e38d78.jpeg) }

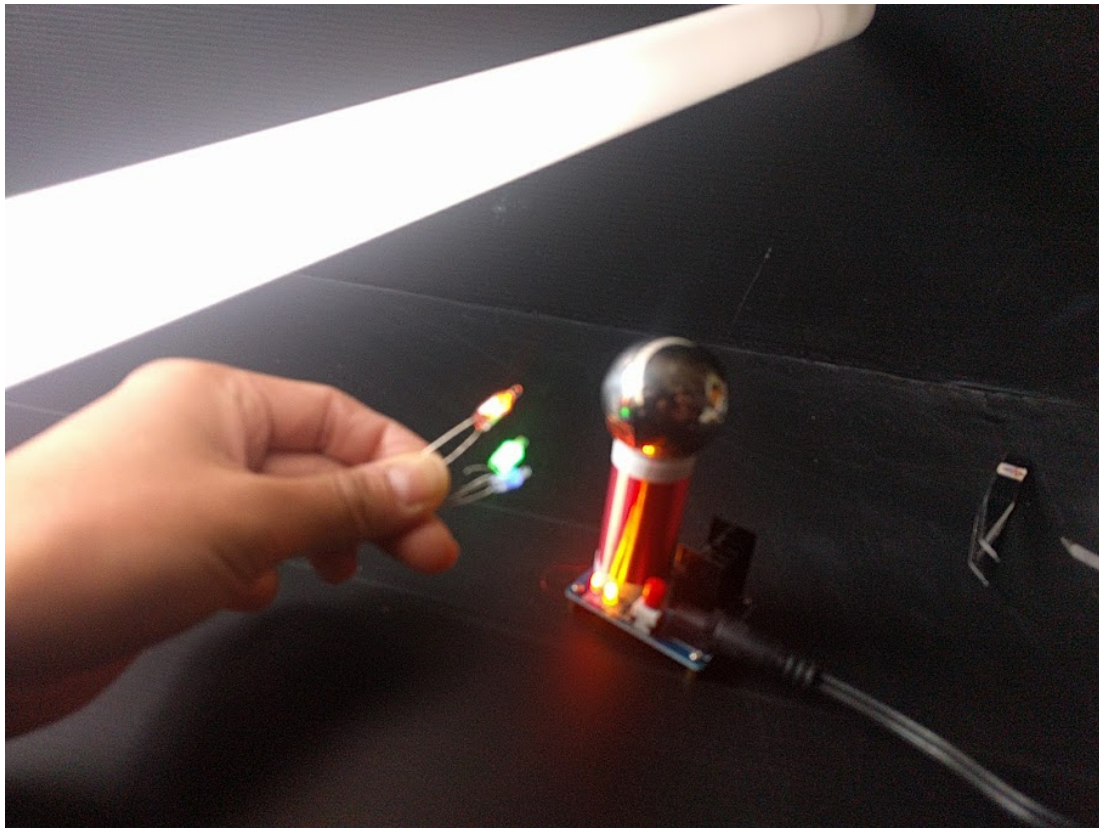


図 5.1 ファイル名

Qiita では、img タグに書き換えると小さくなります。@<tt>{ }

```

```

この解決方法は、GitHub などでも同様です。タグをいちいち書き換えれないといけないのでメンドクサイですね。markdown の拡張でできないかと思ったのですがそういうのは無さそうです。

仕方がないので、Qiita 上では 100% で表示されるけれども、PDF に変換した時に任意の大きさになるようにする。

「画像を初期化するスクリプト」<http://qiita.com/nanbuwks/items/9b00e8012e328de6e440#%E7%94%BB%E5%83%>を使うと、

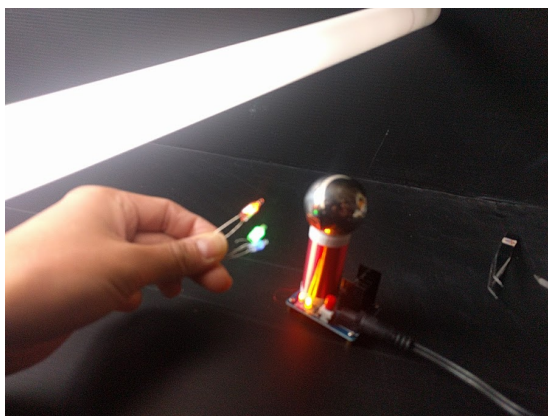


図 5.2 ファイル名

```
[scale=0.5]![ファイル名](https://qiita-image-store.s3.amazonaws.com/0/139524/c19f0873-8b39-9405-1486-8cf43e
```

と書けば、Qiita 上では大きく、紙媒体に印刷したときには 50% で印刷される。けれども `[scale=0.5]` というのが表示されてしまう。

少し書式を変更。

```
[] ( scale=0.5 )![ファイル名]( https://qiita-image-store.s3.amazonaws.com/0/139524/c19f0873-8b39-9405-1486-8c
```

とした。

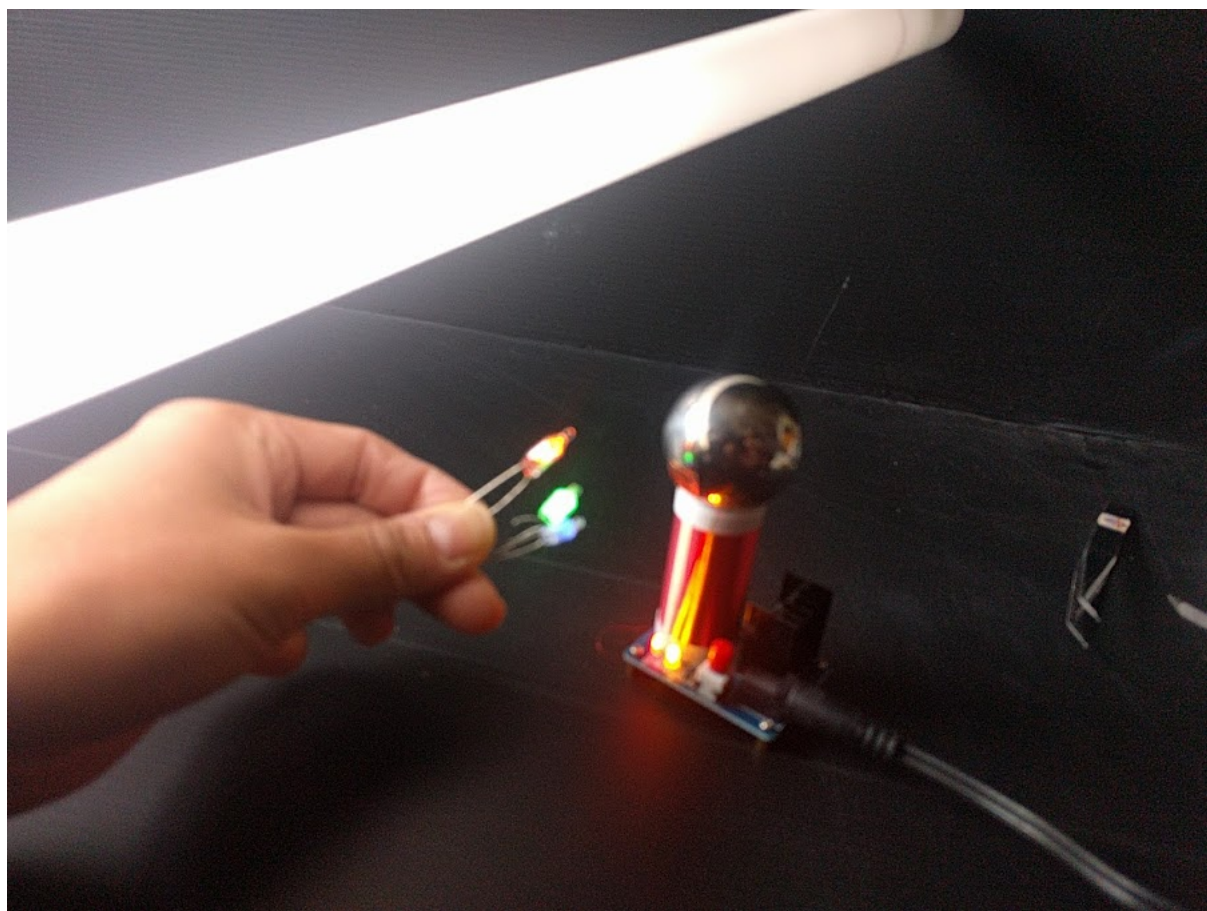


図 5.3 ファイル名

しかしながら md2review でエラーが起こる

```
md2review qiita2review2.md
/var/lib/gems/2.3.0/gems/md2review-1.11.0/lib/redcarpet/render/review.rb:299:in 'remove_inline_markup': un
  from /var/lib/gems/2.3.0/gems/md2review-1.11.0/lib/redcarpet/render/review.rb:191:in 'link'
  from /var/lib/gems/2.3.0/gems/md2review-1.11.0/lib/md2review/markdown.rb:13:in 'render'
  from /var/lib/gems/2.3.0/gems/md2review-1.11.0/lib/md2review/markdown.rb:13:in 'render'
  from /var/lib/gems/2.3.0/gems/md2review-1.11.0/bin/md2review:54:in '<top (required)>'
  from /usr/local/bin/md2review:22:in 'load'
  from /usr/local/bin/md2review:22:in '<main>'
```

```
[] ( scale=0.5 )
```

という書き方は markdown のコメントアウトなので、これでエラーが起こるとは情けないぞ。

```
[ (
)
```

こんなので同様のエラー。仕方がないので通常のコメントはふつーに消去、scale=0.5) は [scale=0.5] にするフィルタを作り、md2review をかける前に適用するようにしよう。

「preprocess.rb」

```
#!/usr/bin/env ruby
## this is not support $...$ in inline htmltag,link,image.
codeBlock=false
incomment=true
while line = gets
  line.chomp!
  if ( codeBlock == false && md1 = line.match(/`"/))
    codeBlock=true
    puts line
  elsif ( codeBlock == true && md1 = line.match(/`"/))
    codeBlock=false
    puts line
  elsif ( codeBlock == false && md1 = line.match(/~\[\\]\(\\ *(scale=.)\\ \\)!\\[.]*\\(\\(.*\\)/))
    md2 = line.match(/~\[\\]\(\\ *(scale=.)\\ \\)!\\[.]*\\(\\(.*\\)/)
    print "[" + md1[1] + "]" + md2[1]
  elsif ( codeBlock == false )
    offset=0
    while md1 = line.index("[(",offset) do
      thereistex=false
      if ( md2 = line.match(/\\$.*?\\$/))
        thereistex=true
      end
      inlinecode="nocode"
      inlinetex="notex"
      for num in offset..md1-1
        ch = line[num]
        if ( ch == "\"" && inlinecode=="nocode" )
          inlinecode="starting"
        elsif ( ch != "\"" && inlinecode=="starting" )
          inlinecode="incode"
        elsif ( ch == "\"" && inlinecode=="incode" )
          inlinecode="ending"
        elsif ( ch != "\"" && inlinecode=="ending" )
          inlinecode="nocode"
        elsif ( ch == "$" && inlinecode=="nocode" && inlinetex=="notex" && thereistex==true )
          inlinetex="intex"
        elsif ( ch == "$" && inlinecode=="nocode" && inlinetex=="intex" )
          inlinetex="notex"
        end
        print ch
      end
      offset=md1+4
    end
  end
end
```

```
num=offset-1
if ( inlinetex!="notex" || inlinecode!="nocode")
else
  incomment=true
  while incomment==true do
    if ( line.length < num )
      if ( line=gets )
        num=0
      else
        incomment=false
      end
    end
    ch=line[num]

    if ( ch == ")" )
      incomment=false
    end
    num=num+1
    offset=num
  end
end
end
line=line[offset,line.length]
puts line
else
  puts line
end
end
end
```

制限

汚いコードになってしまったし、

第 6 章

Re:VIEW を apache で動かす

「AWS 上に Re:VIEW 環境を構築する」<http://qiita.com/nanbuwks/items/da9136f1b6f789aaffcf>からの発展で、Re:VIEW を Web 上で動かします。

「組版システム ReVIEW で Qiita から同人誌原稿を作る」<http://qiita.com/nanbuwks/items/ad4ed8b7fbda846ba997>

この仕組みを使って Qiita にある記事を薄い本形式の PDF にするようにします。

6.1 apache と PHP

```
apt-get install apache2 apt-get install libapache2-mod-php5
```

6.2 DocumentRoot に review 環境を展開

```
cd /var/www/html review-init template chown -R www-data:www-data *
```

6.3 This account is currently not available.

コマンド実行でエラーが出たので/etc/passwd ファイルの
として、www-data でシェルログインできるようにします。

```
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

```
www-data:x:33:33:www-data:/var/www:/bin/bash
```

```
su - www-data
```

としてチェック。

・・・としていたのですが、ロケール設定だけの問題だったかも知れません。現在は、ログオンシェルは /usr/sbin/nologin で運用しています。

6.4 review 環境を展開します。

```
review-init book
```

各種フォルダを作ります。

```
mkdir articles mkdir template
```

6.5 テンプレートを作ります。

```
cp book/config.yml template
cd template
vim config.yaml
```

config.yaml を適当に書きます。

書籍タイトル、著者、表紙を作らない、などを指定しておくといいいでしょう。

複数人で執筆するので、Qiita の1つの記事が1章というようになります。

なので Qiita の記事は

1人目

```
## 1 人目の 1 章見出し
内容
### 1 人目の 1-2 章見出し
内容
## 1 人目の 2 章見出し
内容
```

2人目

```
## 2 人目の 1 章見出し
内容
### 2 人目の 1-2 章見出し
内容
## 2 人目の 2 章見出し
内容
```

3人目

```

.
.
.

```

となりますが、これを

```

## 1 人目のタイトル
1 人目の肩書、名前
### 1 人目の 1 章見出し
内容
#### 1 人目の 1-2 章見出し
内容
### 1 人目の 2 章見出し
内容
## 2 人目のタイトル
2 人目の肩書、名前
### 2 人目の 1 章見出し
内容
#### 2 人目の 1-2 章見出し
内容
### 2 人目の 2 章見出し
内容
## 3 人目のタイトル
.
.
.

```

とします。

なので、qiita から取得した markdown は、見出しレベルを変更するスクリプトを通します。

```
sed "s/^#/##/" qiita.md
```

とすれば良いことになります。

qiita.md は qiita から取得した markdown です。qiita から markdown を取得するには wget を使います。こういったものも含めて、処理を行うシェルスクリプトを作ります。

```

## cat qiitaget.sh
URL='ruby -Ku ../../makeurl.rb'
echo $URL
cd images

wget $URL -k -H -r -l 1 -nH -nd -A png,PNG,jpg,JPG,jpeg,JPEG,html -R txt
cd ..

```

```
rm qiita.md
wget $URL.md -O qiita.md
sed "s/^#/#/" qiita.md | ruby -Ku ../../qiitamd.rb | ruby -Ku ../../texblock.rb | ruby -Ku ../../texinlin
head -1 temp.md | sed "s/^#/#/" > $1.md
ruby ../../makeauthor.rb >> $1.md
tail -n +2 temp.md >> $1.md
ruby ../../escapeincode.rb $1.md | ruby ../../preprocess.rb | md2review | ruby -Ku ../../scalemd.rb > $1.r
rake pdf
mv book.pdf $1.pdf
```

この、qiitaget.sh は PHP からコマンド実行関数で呼び出します。

```
## cat compile.php
<html>
<head>
<meta charset="UTF-8">
<title>qiita2review</title>
<style type="text/css">
body {
    color: #000;
    background-color: #ffffff;
}

pre {
    padding: 1em;
    color: #aaffaa;
    background-color: #aaaaaa;
}
</style>
</head>
<body>
<?php

$sectionName = $_GET["section"];
?>
<h1>section <?php echo $sectionName; ?> compile</h1>

<pre>
<?php
passthru("export LC_CTYPE=en_US.UTF-8; export LANG=en_US.UTF-8; cd articles/" . $sectionName . " ; ../../qiitag

?>

</pre>

<form action="<?php echo "articles/" . $sectionName . "/" . $sectionName; ?>.pdf" METHOD="GET" >
<input type="submit" value="View PDF">
</form>
```



```
<form action="make.php" METHOD="GET" >
<input type=hidden name=section value=<?php echo $sectionName; ?>>
<input type=submit value="return">

</form>
</body>
</html>
```

他の php ファイル、qiitaget.sh から呼び出す各種フィルタスクリプトを作ります。

(省略・・・他記事を参照して下さい)

ファイルは www-data をオーナーにします。

```
## chown -R www-data:www-data *
```

ファイル一覧です

```
root@ip-172-30-0-222:/var/www/html# ls -alh
total 80K
drwxr-xr-x 5 www-data www-data 4.0K Apr  6 14:03 .
drwxr-xr-x 3 root      root      4.0K Apr  6 13:58 ..
drwxr-xr-x 4 www-data www-data 4.0K Apr  6 13:48 articles
drwxr-xr-x 5 www-data www-data 4.0K Apr  5 20:07 book
-rw-r--r-- 1 www-data www-data  803 Apr  5 03:48 compile.php
-rwxr-xr-x 1 www-data www-data  680 Apr  6 12:31 escapeincode.rb
-rw-r--r-- 1 root      root      213 Apr  6 14:03 .htaccess
-rw-r--r-- 1 www-data www-data  1.1K Apr  5 05:02 index.php
-rw-r--r-- 1 www-data www-data  235 Apr  4 16:19 makeauthor.rb
-rw-r--r-- 1 www-data www-data  1.4K Apr  5 20:18 make.php
-rw-r--r-- 1 www-data www-data   87 Apr  5 03:28 makeurl.rb
-rw-r--r-- 1 www-data www-data  1.2K Apr  5 05:11 new.php
-rwxr-xr-x 1 www-data www-data  2.0K Apr  5 19:11 preprocess.rb
-rwxrwxrwx 1 www-data www-data  536 Apr  6 12:58 qiitaget.sh
-rwxrwxrwx 1 www-data www-data  458 Apr  3 17:45 qiitamd.rb
-rwxrwxrwx 1 www-data www-data  222 Apr  4 13:41 scalemd.rb
-rwxr-xr-x 1 www-data www-data  1.4K Apr  6 09:11 slash2braceincode.rb
drwxr-xr-x 2 www-data www-data  4.0K Apr  4 18:23 template
-rwxrwxrwx 1 www-data www-data  286 Apr  4 12:20 texblock.rb
-rwxr-xr-x 1 www-data www-data  1.2K Apr  5 11:18 texinline.rb
```

閲覧系は誰でも見せますが、登録系は basic 認証でアクセス制限をかけます。

```
## cat .htaccess
<Files ~ "^(.htaccess|htpasswd)$">
deny from all
</Files>
AuthUserFile /var/www/.htpasswd
AuthName "Please enter your ID and password"
AuthType Basic
order deny,allow
<Files new.php>
require valid-user
</Files>
```

作成したサイトです



図 6.1 Screenshot from 2017-04-07 09-50-56.png

Qiita と **Re:VIEW** を使ってオンラインで記事を寄せあい同人誌を作って 同時に **Web** コンテンツとして公開できるシステムがこんなに便利なわけではない

2017 年 4 月 9 日 初版第 1 刷 発行

著 者 日本 Android の会秋葉原支部ロボット部
