脱GitHub初心者を目指す人のREADMEマークダウン使いこなし術

[3件の返信](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#comments)



README がキチッと書かれているプロジェクトって、どんなに小さくても立派に見えますよネ。

GitHub の場合、大抵はマークダウン記法で書かれた README.md とか README.markdown とかいう名前のファイルが、HTML に変換 （マークアップ） されて表示されていることはご存知でしょう。

マークダウン記法自体はとても簡単なのですが、GitHub では [GitHub Flavored Markdown](http://github.github.com/github-flavored-markdown/)（略して GFM） という GitHub 用にアレンジされたマークダウン・エンジンが採用されていて、一般のマークダウン・エディタでチェックしてからコミットしても、意図通りの見た目にならないことが多々あります。私 （もちろん GitHub 初心者です！） の場合、README ファイルだけで10回以上もコミットしてしまいました。「マークアップ （レンダリング） を気にして書くマークダウン」 なんて、何ともナンセンスですよネ。

ということで、私の二の舞にならないよう、スマートに README ファイルをコミットする方法を共有してみたいと思います。

マークダウンのこころ

GFM のノウハウにいく前に、そもそもマークダウンの 「こころ」 を知っておくのが良いと思います。マークダウンの発案者 [John Gruber](http://daringfireball.net/projects/markdown/syntax) 本人は、その哲学を次のように語っています。

マークダウンは、簡単に書けるということを重視しながらも、プレーンなテキストとしても読み易くするために、そこに使われる記号は、それ自体で何を意味しているかが分かるよう慎重に選ばれなければならない。

例えば強調を表すシンタックスは、単語をアスタリスクで囲い \*強調する単語\* と表したり、blockquote を表す引用は、電子メールでよく使われる > 引用する文章 といった具合です。

マークダウンのシンタックスには、合計で11個の基本構成要素 （段落、改行、見出し、強調、リスト、引用、コード、リンク、画像、水平線、HTML タグ） が登場します。それぞれのシンタックスは非常に簡単なので、初めての方は一度、学習をしておくとよいでしょう。

お勧めの文献

[Markdown – Wikipedia](http://ja.wikipedia.org/wiki/Markdown)

HTML タグを除く10個の要素がシンプルに解説されています。まずこちらを読んでから、本家の解説を読むのがお勧めです。

[Markdown文法の全訳](http://blog.2310.net/archives/6)

John Gruber による文法解説の日本語訳です。

[Syntax Guideの日本語訳](http://pamgau.net/showdown/)

[Showdown](http://www.ctrlshift.net/project/markdowneditor/) というオンラインツールに付属する Syntax Guide が、本家の解説をさらに詳しく補ってくれます。ツール右上のリストから Syntax Guide を選択すると、右側ペインに表れます。

README.md作成のワークフロー

ローカル側で README.md を書きながらプレビューで確認し、OK となったらリモート側にコミットをプッシュする、というのを1つのワークフローとするなら、その方法は次のいずれかでしょう。

一般的なマークダウン・エディタを使う

[参考情報](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#mdeditor) に挙げたように、世の中にはたくさんのマークダウン・エディタが存在します。しかし、私がハマったように、これらのエディタで OK だったからといって、GFM で OK になるとは限りません。やはりその方言を掴むまでは、GFM 専用のマークダウン・エンジンを探す方が良さそうです。

GFM のマークダウン・エンジンをローカルにインストールする

[github/markup – GitHub](https://github.com/github/markup) に情報がある通り、[redcarpet](https://github.com/vmg/redcarpet) をインストールすればよさそうなのですが、実はコレ、[upskirt](http://fossil.instinctive.eu/libupskirt/index) （現在では [sundown](https://github.com/vmg/sundown) という名前になっている模様） のラッパーだったりして、これらをひも解きながらのインストールは、初心者にはハードルが高いです。

GMF の Live Preview を使う

GitHub Flavored Markdown – Live Preview （[XSS の問題](https://github.com/github/github-flavored-markdown/commit/6e0b7aae082fc3041328acc4b7882e32ba384170) により、Live Preview は削除されました） は、リアルタイムなプレビュー機能を持つオンライン・エディタなのですが、理解に苦しむほど性能が悪く （仕様が古い、多分）、使えたものではありません。私はコレでもハマりました。

GitHub Wiki のマークダウン・エンジンを使う

結局たどり着いた方法がコレです。ご存知の通り、Wiki もマークダウンを採用しています。GitHub の Wiki は、リポジトリ作成後にあらためて開設しなければなりませんが、一旦開設してしまえば、プレビュー機能はもちろん、GUI ベースのリッチ・エディタやヘルプまで付いた GFM 専用のマークダウン・エンジンが使えるようになります！

ということで、以下、GitHub Wiki を活用した README.md 作成のワークフローを提案してみたいと思います。

参考情報

2011年9月20日 [オンラインマークダウンエディタいろいろ比較](http://www.calmtech.net/2011/09/20/online-markdown-editor/)

本家の Perl スクリプト [Markdown: Dingus](http://daringfireball.net/projects/markdown/dingus) とマークアップ結果が （メールアドレスの難読化以外は） ぴったり一致する [Showdown.js](https://github.com/coreyti/showdown) を用いた [Online Markdown Editor](http://www.ctrlshift.net/project/markdowneditor/) が使い易いと思います（日本語バージョンは [こちら](http://pamgau.net/showdown/)）。

2011年10月2日 [リアルタイムプレビューができるMarkdownエディタ『Mou』](http://veadardiary.blog29.fc2.com/blog-entry-3492.html)

Mac 用です。

2011年10月20日 [リアルタイムプレビューにも対応、Markdown専用のエディター「MarkdownPad」](http://www.forest.impress.co.jp/docs/review/20111020_485035.html)

Windows 用です。

GitHub Wiki を活用したマークダウン作成のワークフロー

提案するのは、GitHub Wiki 上のリッチ・エディタを使ってマークダウンを作成、ローカル側の README.md ファイルにコピペするという方法で、次のようなステップで作業するのがよいと思います。

[GitHub 用マイ・テンプレートを作成する](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#ghtemp)

[GitHub Wiki を開設する](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#ghwiki)

[GitHub マークダウンのヘルプを参照する](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#ghhelp)

[GitHub 独自のマークダウンを活用する](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#ghspec)

[プッシュ後にちょっとだけ修正する](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/#ghedit)

1.GitHub 用マイ・テンプレートを作成する

まずは、基本となる自分用のテンプレートを作っておくのが良いと思います。簡単なチュートリアルも兼ね、ほぼ10個の基本構成要素を入れた [サンプル](https://github.com/tokkonopapa/Markdown) を作ってみましたので、参考にしてください。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 | プロジェクト・タイトル  ======================  ここにプロジェクトの概要を書きます。  行末にスペースを2つ入れると  改行されます。    段落を分けるには、[空行](http://example.com/) を入れます。    使い方  ------  ### インライン ###  インラインのコードは、\*\*バッククォート\*\* (`` ` ``) で囲みます。    ### ブロックレベル ###      function f () {          alert(0);  /\* 先頭に4文字のスペース、                        もしくはタブを挿入します \*/      }    パラメータの解説  ----------------  リストの間に空行を挟むと、それぞれのリストに `<p>` タグが挿入され、行間が  広くなります。        def MyFunction(param1, param2, ...)    +   `param1` :      \_パラメータ1\_ の説明    +   `param2` :      \_パラメータ2\_ の説明    関連情報  --------  ### リンク、ネストしたリスト  1. [リンク1](http://example.com/ "リンクのタイトル")      \* ![画像1](http://github.com/unicorn.png "画像のタイトル")  2. [リンク2][link]      - [![画像2][image]](https://github.com/)      [link]: http://example.com/ "インデックス型のリンク"    [image]: http://github.com/github.png "インデックス型の画像"    ### 引用、ネストした引用  > これは引用です。  >  > > スペースを挟んで `>` を重ねると、引用の中で引用ができますが、  > > GitHubの場合、1行前に空の引用が無いと、正しくマークアップされません。    ライセンス  ----------  Copyright &copy; 2011 xxxxxx  Licensed under the [Apache License, Version 2.0][Apache]  Distributed under the [MIT License][mit].  Dual licensed under the [MIT license][MIT] and [GPL license][GPL].    [Apache]: http://www.apache.org/licenses/LICENSE-2.0  [MIT]: http://www.opensource.org/licenses/mit-license.php  [GPL]: http://www.gnu.org/licenses/gpl.html |

マークダウンでは、11個目の要素である HTML タグも使うことができます。例えば <table> や <dl><dt><dd> などを使いたいこともあるでしょう。その場合、次の3つの注意事項があります。

<u> や <del> などのインライン要素中では、マークダウン記法が使える。

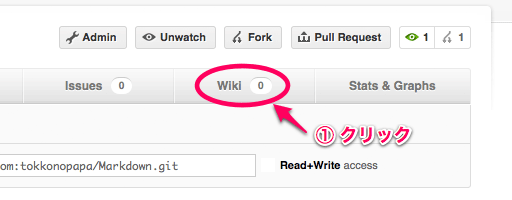
ただし <font> は使えない。

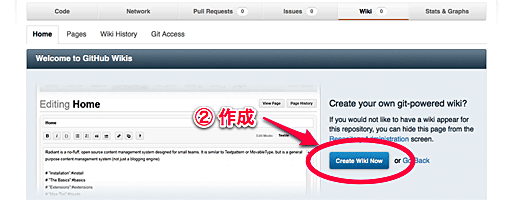
<div> などのブロック要素中では、マークダウン表記は無視される。

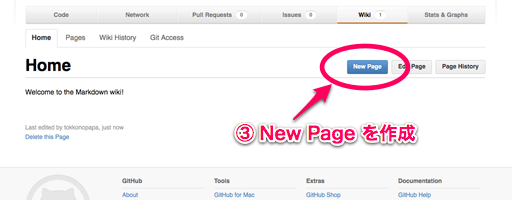
ただし 「マークダウンのこころ」 からは外れてしまうので、Wiki 以外では極力使わないようにしましょう。

2.GitHub Wiki を開設する

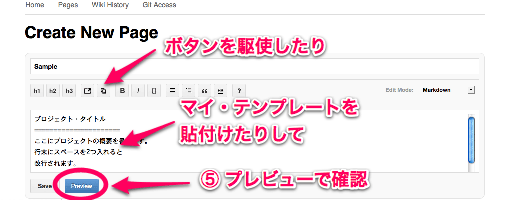
特に難しくはありませんが、Wiki の開設 〜 リッチ・エディタまでの概略を以下に示します。各リポジトリ毎に開設するのも手ですが、無料枠でのリソース消費を抑える意味も含め、マークダウン専用のリポジトリで Wiki を開設、前章のテンプレートを載せておくという手もあるかと思います。



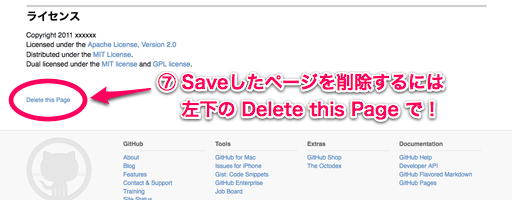






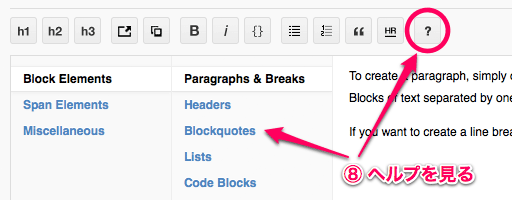




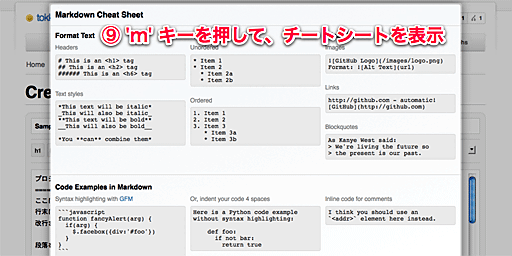


3.GitHub マークダウンのヘルプを参照する

Wiki のリッチ・エディタで 「?」 を押すと、ヘルプを表示させることができます。基本構成要素の10個が、ブロック要素とインライン要素とに分けて参照することができます。



GitHub マークダウンは、README だけでなく、コメント欄でも使うことができます。GitHub のどのページでも良いので、キーボードの ‘m‘ を押してみましょう。使いたいときにいつでも呼び出せる、マークダウンのチートシートを表示させることができます。



余談ですが、’?‘ キーを押すと、ショートカット・キーのチートシートが表示できます。お試しあれ。

4.GitHub 独自のマークダウンを活用する

知ってるとちょっと GitHub 通を気取れる Tips です。

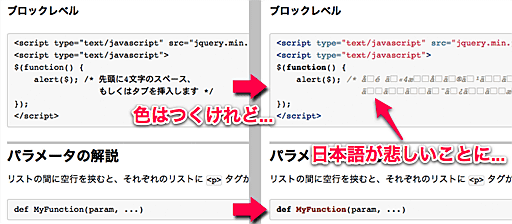
コードブロックのハイライト

コードブロックを 「```lang」 ～ 「```」 で囲むと、指定の言語に合わせてハイライトされます。通常マークダウンでは、コードの先頭にスペース4つか1つのタブを挿入しますが、この記法を使うとブロックが丸ごとそのまま 「<pre><code>」 〜 「</code></pre>」 で囲われるため、余計な空白ができてしまいます。注意しましょう。

また公式ドキュメントには lang に指定可能な言語の解説がないので何とも言えませんが、html、css、javascript、ruby、python などは指定可能なようです。

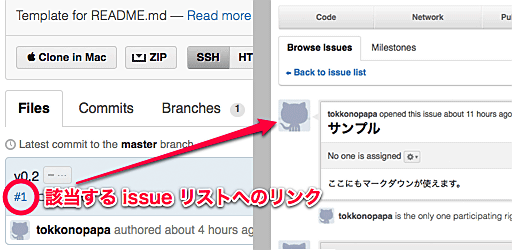
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | ```html  <script type="text/javascript" src="jquery.min.js"></script>  <script type="text/javascript">  $(function() {      alert($); /\* 先頭に4文字のスペース、                   もしくはタブを挿入します \*/  });  </script>  ```    ```python  def MyFunction(param, ...)  ``` |

この機能を使うと、現状ではコメントなどの日本語が化けてしまいます。ユニコード化すれば回避できると思いますが、プレーンなテキストを見たときに何が書いてあるか分からなくなるので、日本語は避けましょう。



issue リストへの自動リンク

「#番号」 で、該当の issue リストへのリンクが自動的に生成されます。この機能は、README や Wiki ページでは働きませんが、コミット時や他のコメント欄では有効です。



5.プッシュ後にちょっとだけ修正する

前章までで、Wiki 上で作成したマークダウンがローカル側の README.md にコピペされコミット、GitHub 側のリポジトリにもプッシュされているものとします。その後ちょっと直したい／プレビューも確認したい、という時には、次のように GitHub 上で修正・確認をすることができます。





最後に git pull で、ローカル・リポジトリの更新も忘れずに。

速攻追記！

[kei-s](http://b.hatena.ne.jp/kei-s/) さんからハテぶ経由で、「よければご利用ください」 というコメントとともに、[Github Preview](http://github-preview.herokuapp.com/) をご紹介いただきました。このツールがまたすばらしい！ GFM と同じ redcarpet を使っていながら、Ajax でリアルタイムなプレビュー機能が実装されていて、作業効率が格段にアップします！ 私は速攻で、ブックマークに叩き込みました。kei-s さん、ありがとうございます！！

ちなみにこの Github Preview で、[Github Wiki の Markdown エンジンが腐ってる件について – 永遠に未完成](http://d.hatena.ne.jp/thinca/20110813/1313215131) にダメなケースとして挙っているパターンを再現させてみました。bar の先頭にスペースを4つ入れれば、正しく変換されます。私がハマっていたのは、正にこういうバッド・ノウハウ的な所です。詳しくは述べませんでしたが、もちろん冒頭のテンプレートにもこの手のノウハウは投入してあります。皆さんも試してみて下さい！

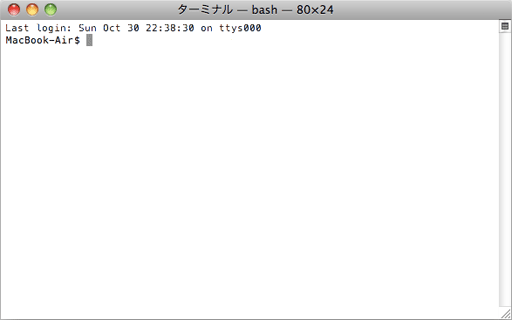
[[](http://github-preview.herokuapp.com/)](http://github-preview.herokuapp.com/)

脱初心者を目指すなら、最後にもう1つ…

個人利用のマークダウンであればどのように書いても自由ですが、GitHub では、いつ誰に fork されてもよいように、1行に記述する文字数にも気を配りましょう。

かつて [VT100](http://ja.wikipedia.org/wiki/VT100) のようなキャラクター端末しかなかった時代は、80桁×24行の表示が一般的でした。それゆえドキュメントは1行の文字数が80桁に収まるように記されたものです。1920桁×1440行といった巨大なモニターさえ手に入る現代でも、プレーンなテキストファイルは、横スクロールしなくても読めるようにするのがココロ遣いと言うものでしょう。

「オッ、コイツできるな」 と思われること請け合いです  。



おまけ

今回の記事を作成するために、マークダウンの本家仕様と方言を幾つか調べていた中で見つけた、興味深い記事を紹介しておきます。本家仕様の曖昧さゆえ、色々とあるようです。

2008年6月24日 [Three Markdown Gotchas](http://blog.stackoverflow.com/2008/06/three-markdown-gotcha/)

書いた意図と違う変換がなされてしまうパターンが3つ指摘されています。

2009年10月15日 [Markdown, One Year Later](http://blog.stackoverflow.com/2009/10/markdown-one-year-later/)

本家仕様と GFM や Stack Overflow のマークダウン仕様の違いが述べられています。

2009年10月23日 [GitHub Flavored Markdown](http://daringfireball.net/linked/2009/10/23/github-flavored-markdown)

John Gruber 本人が、本家仕様と GFM が異なる点を3つほど挙げています。具体的には [GitHub Flavored Markdown – Introduction](http://github.github.com/github-flavored-markdown/) に記述がありますが、現在の GFM 仕様と多少違う （古い） 部分があるようです。

カテゴリー: [プログラミング](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/) | タグ: [GitHub](http://tokkono.cute.coocan.jp/blog/slow/index.php/tag/github/), [tips](http://tokkono.cute.coocan.jp/blog/slow/index.php/tag/tips/) | 投稿日: [2011年11月7日](http://tokkono.cute.coocan.jp/blog/slow/index.php/programming/markdown-skills-for-github-beginners/) |