Git と GitHub についてのメモ

# GitHub

Git を使ったリポジトリホスティングサービスであり、
”ソーシャルコーディング”の発祥の地といえるサイト。
https://github.com/
リポジトリ容量は無制限。公開リポジトリは、いくつでも作成可能＆何人でも開発可です。

## Pull Request について

プログラマが GitHub とどう関わっているのか垣間見て感じたこと
Pull Request を送るのに確認を取る必要はないということ ...。
勝手に Fork して Pull Request したらいい。その気軽さが GitHub の文化だそうです。

GitHub 直伝 Pull Request 活用の 3 つのコツ：海外テック情報局
GitHub 自身のプロジェクトで Pull Request をどのように使いこなしているか、触れられています :

## コミットログについて

基本的に英語文化です。なのでコミットログも英語がベスト。

- git commit するまえに考えるべき 10 のこと
- ChangeLog を支える英語 - よく使われる書き方のまとめ。コミットログを書くにも参考になります。

### ■コミットログで使える Issue 連携文字列

```
fixes #24
fixed #24
fix #24
closes #24
close #24
closed #24
resolve #24
resolved #24
bugfix #24
bugfixed #24
```

( 引用 : http://sue445.hatenablog.com/entry/2012/12/02/000132)

## プロジェクトのライセンスについて

ライセンスについて特に規定はありません。
現に、GitHub 上にはライセンスが明記されていないプロジェクトも少なからず見かけられます。
( ただし個人的には、README や LICENSE ファイルなどにきちんと明記しておくことを推奨します。
ちなみに、MIT や GPL がやはり多いですね。)

## ブランチの運用について

見えないチカラ : A successful Git branching model を翻訳しました - 鉄板ですね。
私もこれを少し簡素にした感じで運用しています。

### ■ GitHub クローン

GitHub では、プライベートリポジトリは有料サービスとなっています。
そのため、無料で同様の機能を使いたいという開発者 ( オープンソースにできないプロジェクトの開発者 ) によって、GitHub クローンが開発されています。

中でも、GitLab が割と使いやすい感じです。
( ※ GitHub 自体はオープンソースではないため、クローンといっても第三者によるフルスクラッチです。その為、細かい機能は色々と違います。また GitHub のほうが個人的には使い勝手が上だと思います。)

## gist

GitHub による (Git を使った ) コードスニペット管理サービス。
https://gist.github.com/
ディレクトリは作成できないですが、単一または複数ファイルを管理できます。
( 公開・非公開が選択でき、いずれも無料。)
ブログなどにコードシンタックスハイライトされたコードを表示できる HTML コードも生成できます。

### gist コマンド

公式ではないようですが、Gist をコマンドラインから操作できるコマンド。
スクリプトをサッと書いてアップできるわけです。
https://github.com/defunkt/gist

元々のバージョンの gist コマンドでは、GitHub の ID/ パスワードをローカルの設定ファイルに記述する必要があるのですが、Fork されたバージョンで OAuth2 でアクセスできるものがありました : https://github.com/dr4g0nnn/gist

また、関係ないのですが、Vim や Sublime Text などのエディタからサクっと操作できる同様のプラグインもあります。

### 二段階認証を設定している場合に Gist を clone するには

Gist はふつう HTTPS で clone しますが、GitHub で二段階認証を設定している場合には
認証パスワードとして、別途ランダムで発行しておいた文字列を入力する必要がでてきます。
そのパスワードを PC 上に記憶させておくこともできますが、
それよりも、SSH で clone する ( 公開鍵認証を使う ) ことを推奨します。

拙作ですが、Gist を SSH で Clone するためのスクリプトがありますので、よければご利用ください。
https://gist.github.com/mugifly/7147811
( これがなくとも手動でもできますが、少し面倒です。)

このスクリプトをパスが通るディレクトリへ入れておくと ...

```
$ sudo cd git-clone-ssh.pl /usr/local/bin/gist-clone-ssh
```

次のように、楽に Gist を SSH で clone できます。

```
$ gist-clone-ssh https://gist.github.com/mugifly/7147811
```

---

# Git

VCS の一つ。Linux を管理するプロジェクトにも使われている (... というか元々その為に開発された )。

## すべてのリモートブランチをローカルに取得する

https://gist.github.com/mnogu/3844699
リモートのリポジトリサービスを移行する場合にも、これを応用すれば丸ごと移行が可能です。

## 簡単な用語

自分用まとめです。間違っている可能性もあります。ご了承を。

### ■ワーキングツリー / working tree

作業場所

### ■インデックス / index

コミットされるファイルを管理する領域

### ■クローン / clone

リモートリポジトリからローカルリポジトリを作成すること ( 複製 )
(Subversion のチェックアウトと異なるのは、リポジトリの履歴なども含めてまるごとを「複製」することである。)

### ■コミット / commit

インデックスに登録された内容をローカルリポジトリに反映すること。(SVN のコミットに近い)

### ■プッシュ / push

ローカルリポジトリからリモートリポジトリへ変更を送信すること。

### ■プルリクエスト / pull request

他人のリポジトリへ差分パッチを送ること。
github では、他人のリポジトリへ直接プッシュすることはできない。以下の手順のようになる。

- 他人リポジトリを clone することで、自分用ローカルリポジトリを作る
- 自分用ローカルリポジトリへコミットしていく ...
- 他人リポジトリを fork することで、自分用リモートリポジトリを Github に作る。
- 自分用ローカルリポジトリから自分用リモートリポジトリへプッシュする。
- 自分用リモートリポジトリから、他人リポジトリにへ " プルリクエスト " を送る
- 他人リポジトリのオーナーが、受け取ったプルリクエストをマージすることで他人リポジトリに反映される。

---