

gitの使い方

作成日: 2017-07-16

created by koh fujiwara(kurenaif)

このような経験はありませんか

- 何が最新のファイルかわからない
 - いつ編集したかもわからない
- 誰が編集したファイルかわからない
- どこが変わったのかわからない
- 編集してバグってどうしようもないので戻したいがctrl+zが有効な範囲を越えている
- なんで編集したのか理解できない
- 定期的にzipに固めて現在日時を付ける運用に嫌気が差した
- ちょっと前までうまく動いていたのに今は動かない泣きたい
- なんかコメントアウトされたコードがたくさんあるけどいつ入れたのかわからない

それ バージョン管理で解決できますよ

バージョン管理ソフト

git, subversion, etc...

実はDropboxとかgoogle Driveとかも履歴を残してくれるので、ちょっとした作業はそれらのオンラインストレージ上でやると結構安心だったりします。

subversionは**古い**でも結構使われてるので必要に応じて使っう必要あり？

ここではgitを使っていきます

git?

git: 分散バージョン管理システム

サーバーにもバージョン管理の情報はあるし、それぞれのPCにもバージョン管理の情報がある。

最悪サーバーがダウンしてもわりとどうにかなる

gitサーバー

オンライン上ですでにあるサービス

publicにするならこれ

- GitHub
- gitbucket

研究室とかで立てるOSSなサーバー

早かったりprivateだったりして良いです

- gitLab (研究室で使ってるのはこれ)
- Bitbucket

[演習0] gitをインストールしよう

<https://git-scm.com/downloads>

`apt-get` や `brew` や `yum` でも入った気がします

[演習1] アカウントを登録しよう

<https://github.com/>

Gitを使ってみよう(Level 1)

これから以下のコマンドを試します.

- clone
- add
- commit
- push

Gitを便利に使うツール

GUIじゃないと辛い時が多いです
おすすめは

- SourceTree
- qgit

その他エディタ備え付けのgitツール

[演習2] 二人チームを作ってください

余った人はプレゼンターと組むことになります.

clone コマンド

リポジトリを自分のところに持ってくるコマンド

```
git clone リポジトリ名
```

[演習3] clone してみよう

```
git clone https://github.com/kurenaif/how\_to\_git
```

[演習4] リポジトリを作ってみよう

リポジトリ名: 貴方のログインネームなど(この中でuniqueが良い)

<https://github.com/new>

して

```
git clone
```

[演習5] ファイルを作ってアップロードしてみよう

1. テキストファイルを作る(intro.txt: I am [name])
2. `git add --all` #リポジトリのすべてのファイルを選択(ちょっと嘘)
3. `git commit -m "first commit"` # first commitという名前で変更
に名前をつける)
4. `git push origin master` # ファイルのアップロード(ちょっと嘘)

特定のファイルを選択したい場合は,

```
git add a.txt b.txt *.c
```

などとする.

さっき作ったgithubのディレクトリのページに移動してみる

Gitを使ってみよう(Level 2)

これから以下のコマンドを試します.

- branch
- diff
- checkout
- merge
- pull
- pull request

branch機能

通常, `master` ブランチは安定版を置いておくもの

- `git clone`して動かないのはつらい
- 動かない状態からみんな編集してみんなつらい
- githubのページにいつてcloneしたら動いてほしい

作りたい機能を実装する時に `branch` を切る
ちゃんと知りたい時: **github flow**で調べよう

```
master) 0 -> 0 -> 0  
new br)           -> 0 -> 0 -> 0
```

[演習6] branchを切ってファイルを編集する

```
git branch hobby # hobbyブランチを作成する
git checkout hobby # hobbyブランチに移動する
# (ここで, intro.txtにあなたの趣味を追記する)
git diff # ここで差分が表示される
git add intro.txt
git commit -m "[add] add my hobby to intro.txt"
git push origin hobby # hobbyブランチをアップロードする
```

ここでgithubのページに行き, hobbyブランチを確認してみる

[演習7] pull requestを発行し, mergeする

[https://github.com/\[アカウント名\]/\[リポジトリ名\]/pulls](https://github.com/[アカウント名]/[リポジトリ名]/pulls)

これでリモート上でmasterブランチにmergeされる
これをローカルに持ってくる.

```
# (ここで, intro.txtを確認する)
git checkout master #ここではまだ趣味は入っていない
# (ここで, intro.txtを確認する)
git pull # ここでリモートのmasterの情報を持ってくる
# (ここで, intro.txtを確認する)
```

[演習6'] branchを切ってファイルを編集する

生年月日ブランチを作ってintro.txtに生年月日を追加してcommitまでしてみよう(pushはしなくて良い)

[演習8] mergeしてみる

わざわざpushしてpull rqしなくてもローカルでマージする方法もある(個人で作業する時用)

```
# (ここで, intro.txtを確認する)
git checkout master
# (ここで, intro.txtを確認する)
git merge (さっき作ったbranch)
# (ここで, intro.txtを確認する)
git push origin master # リモートに反映させておく
```

[演習9] 他人のリポジトリを編集してプルリクを送ってみる

1. branchを切る
2. あなたのintro.txtの中身をを他人のintro.txtの末尾に追加
3. add commit push
4. pull rq
5. 相手がpull rqを確認, 受理
6. ローカルにpullする

[最後の演習] 二人で以下の計算をします

$$MOD = 10^9 + 7$$

$a \leq 10^7$ の非負整数

$b \leq 10^7$ の非負整数

```
fibo(pow(a,b,MOD),MOD)
```

を求める

fibo(number, MOD): number番目のフィボナッチ数列のMODを求める.

pow(x,n,MOD): x^n のMODを求める.

hints

1. チームのどちらかがリポジトリをたててください
2. fiboとpowどっちを実装するのか決めてください
3. branchを切って関数を実装
4. pull rqを発行して, リポジトリをたてた人がマージしてください
5. 最後に出力するブランチを切って出力を実装してください
6. Test

```
a = 1241230  
b = 1230124  
MOD = 1e9+7  
で  
344997604  
が出力されます
```

tips

- 前のcommitに戻したい

```
git reset --hard
```

- git reset hardをミスった

```
git reflog
```

- ./a.outをcommitしたくない
 - .gitignoreに追記
- `git commit` とか打つのがめんどくさい
 - .gitconfigに `git cm` 等にエイリアスを貼る