Git / GitHub

Yanny's Computer 山﨑祐太

目的

本チュートリアルでは，以下の点について学んでいきます.

- Git / GitHubとは何か
- 基本的な使い方
- チーム開発におけるGitのTips

Gitとは

- **分散型のバージョン管理ツール**
- 他にもバージョン管理ツールはあるが，現状**Gitの一択**
- ローカルとリモート(サーバー上)の両方にリポジトリをもつ
- ローカルで作業を行い変更履歴を更新し，リモートに変更を取り込む

GitHub(GitLab)とは

- GitHubは**Gitをより便利に利用できるwebサービス**の名前
- チーム開発やコードの差分チェックなどが便利に！
- 同様のwebサービスにGitLabなどもある

基本的な使い方

1. リモートリポジトリをローカルにclone
2. 作業用のbranchを切る
3. ローカルで作業を行い，変更をcommit
4. ある程度commitが溜まったら(別にcommitは1つでも良い)push
5. pushされたbranchを元のbranchにmerge

1.リモートリポジトリをローカルにclone

```
git clone -b develop https://github.com/yutayamazaki/Gin-Template.git
```

- GitHubなどからリポジトリをローカルに取ってくる
- `-b develop` でdevelopブランチを取ってくるという意味(ブランチは後で解説)
- リモートから取ってきて，ローカルで作業，リモートに変更を取り込むと言う流れになる

2. 作業用ブランチを切る

```
git checkout -b feature/bugfix
```

- `git checkout -b ブランチ名` でブランチを作成する
- とりあえず `feature/変更や機能の名前` というブランチを作成すればok
- 作業をcommitした後にこのブランチを元のブランチにマージする

branch

- ブランチとは本流のmasterから分岐されたもの
- 複数のブランチを作成してそれを本流に結合するという流れで開発する
- これにより複数人や複数チームが並行して別機能の開発を行える

3. ローカルで作業を行い，変更をcommit

```
git add main.py
git commit -m "fix main.py"
```

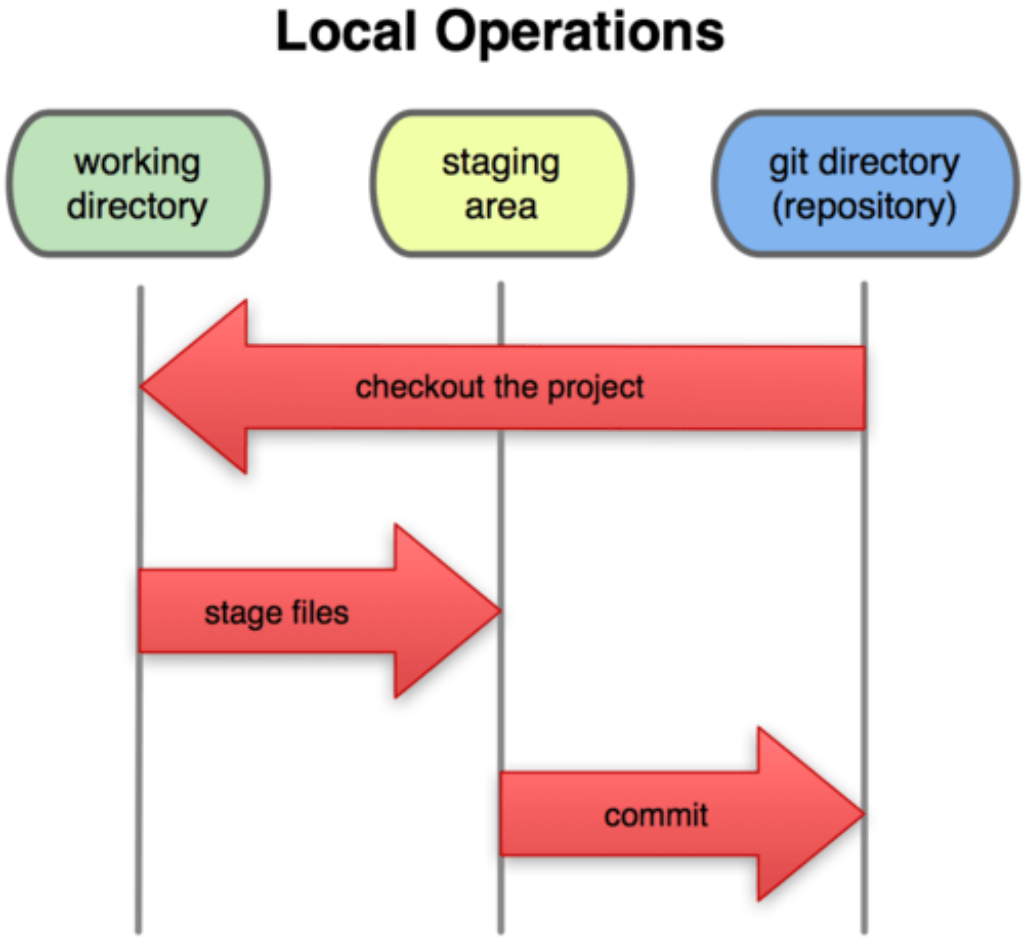
- `git add file名` でステージングエリアに上げる
- `git commit -m "コミットメッセージ"` で変更を記録する

commit

- Gitにおけるバージョン管理の単位
- ひとまとまりの作業を行うたびにcommitを行い，適宜変更を記録していく
- `git commit -m "コミットメッセージ"` でcommitする
- メッセージは `fix function` など「動詞+目的語」で何をどうしたかを書く
  - 参考記事：GitHubで使われている実用英語コメント集
- commitの単位はひとまずは大きすぎなければok

add

- Gitでcommit出来るのはステージングエリアにあるファイルだけ
- ステージングエリアがあることで，commitの単位を調整できる
- 以下の流れでcommitを行う
    i. ファイルに変更を加える
    ii. ステージングエリアに追加
    iii. commit



4. ある程度commitが溜まったらpush

```
git push origin feature/bugfix
```

- 自分の作業ブランチをリモートに送る
- その後GitHubやGitLabでプルリクエストやマージリクエストを作成
- コードレビューや修正を経て元のブランチにマージされる

GitのTips