

最低限 git 入門

北海道大学理学部 ひとみさん

令和元年 11 月 29 日

扱うことと扱わないこと

扱うこと

- git とは何か
- リポジトリの作り方
- コミットのしかた

扱わないこと

- 複数人で開発する話
- ブランチ機能

- **Pro Git 2nd Edition**

<https://git-scm.com/book/ja/v2>

git 公式の日本語マニュアル

- **サルでもわかる Git 入門**

<https://backlog.com/ja/git-tutorial/>

とりあえずこれを読めば実用できるようになる

git とはなにか

git とはバージョン管理システムである

バージョン管理システムとは
ファイルの変更履歴を記録する

(記録するためのデータベース → リポジトリ repository)

- いつ変更があったか参照できる
- 過去の状態に戻すことができる

git はリポジトリをローカルとリモートに保存する

git を使うと有用な場合

- **テキストデータを開発する場合**
変更を参照しやすい（どの行を削除/追記したか）
- **複数人で開発する場合**
リモートリポジトリから（他人が行った）変更を取り込める
複数人開発をしたことがないので、この話はないです
- **開発に使う端末が複数ある場合**
リモートリポジトリ経由でデータを同期できる
本来の使い方ではなさそう（私はやっている）

git のインストール

適当なパッケージ管理システム経由でインストール

```
$ sudo apt install git
```

```
$ git --version
```

git version 2.24.0

設定

ユーザー名とメールアドレスを登録

```
$ git config --global user.name "<ユーザ名>"
```

```
$ git config --global user.email "<メールアドレス>"
```

~/.gitconfig に保存

リモートリポジトリを保存する場所をつくる
GitHub¹にリモートリポジトリを置くのがメジャー

GitHub にアカウントを作りましょう。

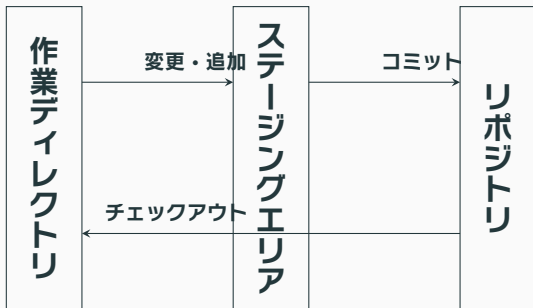
世の中に GitHub しかないわけではないし、自前の git サーバーも作れる

¹<https://github.com/>

git の概要

- リポジトリがリモートだけでなく、ローカルにもある
- ほぼすべての作業をオフラインでできる
- リモートサーバーが死んでも復元できる（分散管理）

3つの状態



git の利用

1. `git init`
ローカルリポジトリの作成
2. `git remote add origin <url>`
リモートリポジトリの追加
3. `git add .`
新しいファイルの追跡、ステージングエリアに追加
4. `git commit -m "comment"`
コミット
5. `git push origin master`
リモートへプッシュ

git の利用

- `git pull`
リモートからのプル
- `git status`
現在の状況の確認
- `git rm`
ファイルの削除 (`--cached` オプションでステージングエリアからのみ除外)
- `.gitignore`
ステージングエリアに自動で追加されない・変更を追跡されないファイルを指定する。
<https://github.com/github/gitignore>

- ・ サルでもわかる Git 入門

<https://backlog.com/ja/git-tutorial/>

とりあえずこれを読めば実用できるようになる

- ・ `$ man git` `$ man gittutorial`

「わからない時は `man` を読め」

- ・ Pro Git 2nd Edition

<https://git-scm.com/book/ja/v2>

git 公式の日本語マニュアル