

Git勉強資料

情報通信ネットワーク研究室
高橋優作

Gitとは

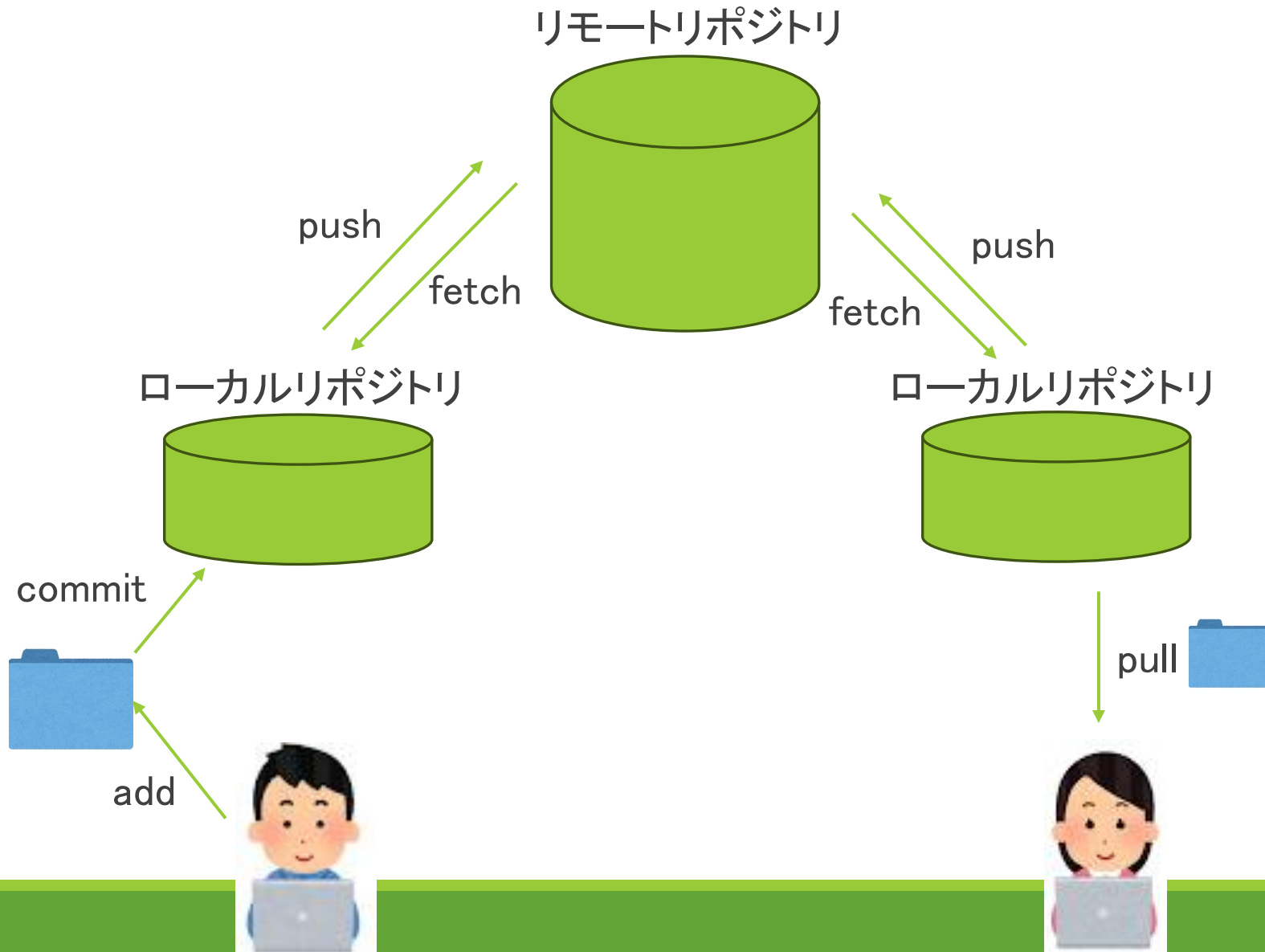
- Git
 - ソースコードやドキュメントの変更履歴を管理するための分散型バージョン管理システム
- GitHub
 - Gitを利用したリポジトリをインターネット上で共有・保存・共同開発するためのプラットフォーム



Gitの特徴

- ファイルの変更 → commit
- 分散型
 - ローカルでも管理可能
 - オフライン作業
- ブランチ(Branch)
 - 開発の分岐を管理(バグ修正の分離, 新機能開発)
- マージ(Merge)
 - 分岐した変更を統合

簡略図



WSL de Git

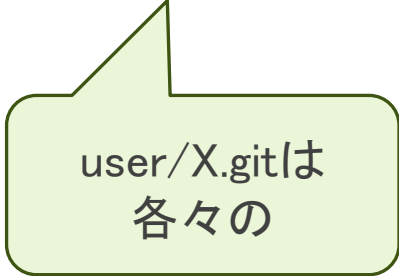
- `$ git --version`
 - もし, インストールされていないなら
 - `$ sudo apt update`
 - `$ sudo apt upgrade`
 - `$ sudo apt install git`
- Gitユーザ情報初期設定
 - `$ git config --global user.name "NAME"`
 - `$ git config --global user.email "EMAIL"`

WSL de Git2

- リポジトリにファイルを追加する
 - `$ cd ~/test`
 - `$ git add .`
 - `$ git commit -m “update file”`
 - `$ git push`
- git add ○○の補足
 - “.”で指定するとすべてのファイルが対象となる
 - ファイル名指定で当該ファイルのみをadd対象とできる

WSL de Git3

- 新規リポジトリとしてGitHubに追加する
 - `$ cd ~/test`
 - `$ git init`
 - `$ git add .`
 - `$ git commit -m "Initial commit"`
 - `$ git branch -M main`
 - `$ git remote add origin https://github.com/user/X.git`
 - `$ git push -u origin main`



user/X.gitは
各々の

WSL de Git4

- 認証

- ① HTTPS

- \$ git push
- UsernameとPasswordを求められる
GitHub上でSettingsからPersonal access tokensを設定

- ② SSH

- \$ ssh-keygen -t ed25519 -C "EMAIL"
- \$ cat ~/.ssh/id_ed25519.pub
- catで得たSSH鍵をGitHubのSSH Keysに登録
- \$ git remote set-url origin git@github.com:user/X.git

user/X.gitは
各々の

GitHub上の注意点等

- public と private について
 - 他人に公開するためにはリポジトリをpublicにする必要有
 - 見せたくないファイルは.gitignoreに追加すること
- 履歴が残る
 - commitすると何をどう変更したのかが履歴として残る
 - 後から上書きをしても履歴は残ることに注意

GitHubを使う最大の利点

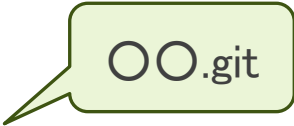
- リモートリポジトリを複数端末から作業可能
 - つまり、複数人で一つのリポジトリを改良可能
- 勝手に変更されてしまう？
 - 変更の競合や意見の衝突で変更が自由だとリポジトリ破壊が起きるのではないか
 - GitHubではpull requestという機能があり、他ブランチでの変更を承認してからmainに統合することが可能

複数人 de GitHub 1

- リポジトリをcloneする

- `$ git clone <repository-url>`

- `$ cd <repository-url>`



〇〇.git

- Gitの状態を確認

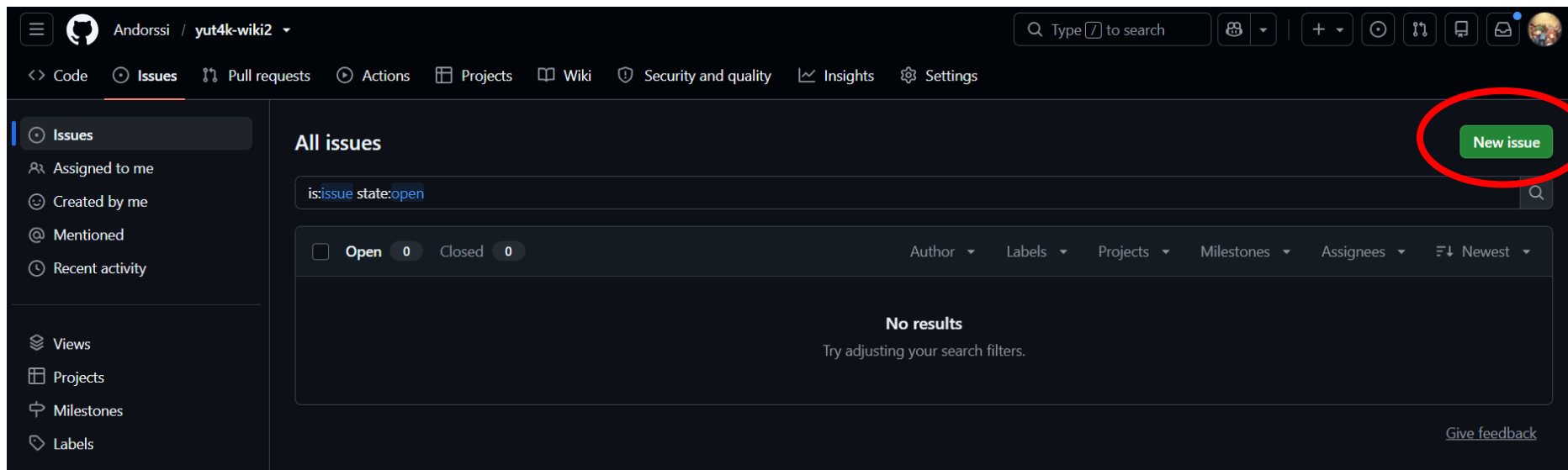
- `$ git status`

- `$ git remote -v`

- `$ git log --oneline -3`

複数人 de GitHub2

- Issue作成
 - Issueはプログラムや設計に関する問題点を一覧で管理することが可能
 - New issueから各々が作成する
 - 作成されたIssueは「# ○」という形でタグ管理される



複数人 de GitHub3

- 作業用ブランチを作成
 - mainブランチで全員が作業することは危険
 - 別ブランチで変更, テストをし, pull requestという形で承認を得てからmainに統合(Merge)する
- 実際にやってみる
 - `$ git switch main`
 - `$ git pull`
 - `$ git switch -c <New Branch Name>`
 - `$ git status`
 - `$ git branch`

最新のmainに更新

ブランチを作成し移動

ブランチ一覧を表示

複数人 de GitHub4

- commit, pushは先ほどと同様
 - \$ git add XXX
 - \$ git commit -m “refactorXXX”
 - \$ git push -u origin <Branch Name>
- mainブランチで作業しているとこの段階でmainが更新されてしまう
 - 別ブランチにpushする分にはmainが変わらないので問題ない

最後に

- GitHubでの一通りの作業する流れを紹介した
 - 実際にリポジトリを作成し、作業してみる
 - 一人でもブランチを分けて作業することは有効