

知識情報システム実習A

Git（ギット）入門（2）

知識情報・図書館学類
（知識情報システム主専攻）
大澤文人

osawa[at] slis.tsukuba.ac.jp

今週の授業までにやっておくこと

各自の PC で Git を使えるように準備してください

1. ターミナル (Mac/Linux)

PowerShell や コマンドプロンプト (Windows)

上で, `git --version` と打ったときに,
バージョンが表示されるか確認

2. GitHub にユーザ登録

3. GitHub で設定したユーザ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"
```

Git の基本

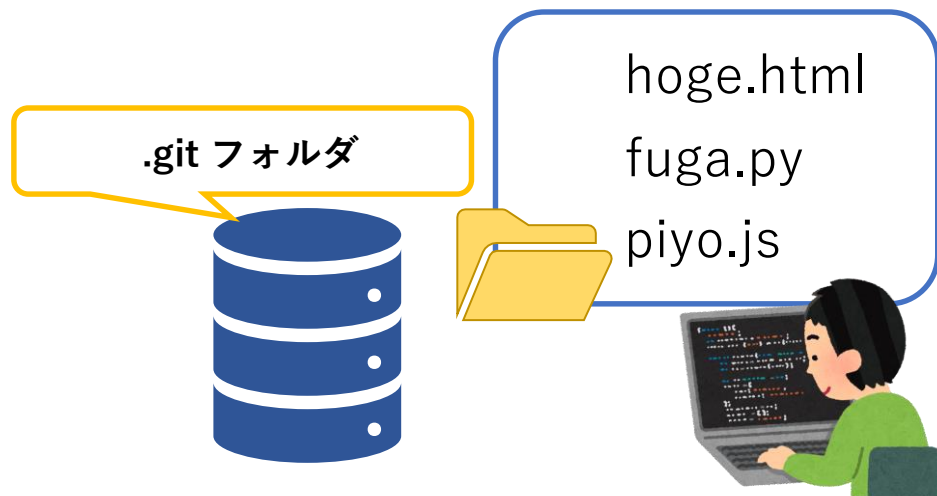
◆ リポジトリ (repository)

- ファイルやディレクトリの状態を記録する場所
- 変更の履歴を管理したいディレクトリをリポジトリの管理下に置くことで、ディレクトリ内のファイル等の変更履歴を記録できる



ローカルリポジトリの作成

1. ローカルPC上のプログラムを作成しているフォルダに移動
2. `$ git init`
というコマンドを入力
→ リポジトリの情報が保存される
.git フォルダが作成される



変更履歴をリポジトリに記録

1. 主専攻実習で作成するシステムの説明ファイル
Readme.txt をローカル PC 上のプログラムを作成するフォルダで作成
2. 変更を記録するファイルを選別（**ステージング**）
 - 作成したフォルダで、以下を実行
`$ git add Readme.txt`
 - フォルダ内のすべてのデータを記録する場合
`$ git add .`
3. リポジトリに記録（**コミット**）
 - ステージングを行った後に以下を実行
`$ git commit -m "コミットメッセージ"`
(最初のコミットメッセージは "initial commit" など)

変更状況の確認

1. Readme.txt を変更

2. 変更状況の確認

\$ git status

- Changes not staged for commit:
modified: Readme.txt ... などと表示される

3. \$ git add Readme.txt を実行

4. \$ git status を実行

- Changed to be committed:
modified: Readme.txt ... などと表示される

5. 新しいコミットを実行

\$ git commit -m "Readme.txt を変更"

6. \$ git status を実行

- nothing to commit, working tree clean などと表示される

変更差分・変更履歴の確認

1. Readme.txt を変更

- 変更差分の確認

\$ git diff

→ 差分が表示

2. \$ git add Readme.txt を実行

- 変更差分の確認

\$ git diff

→ 差分が表示されない

- 変更差分の確認

\$ git diff --staged

→ 差分が表示

3. \$ git commit を実行

- 変更差分の確認

\$ git diff

→ 差分が表示されない

- 変更差分の確認

\$ git diff --staged

→ 差分が表示されない

4. 変更履歴の確認

\$ git log

変更の取り消し

1. Readme.txt を変更

2. 変更状況の確認

```
$ git status
```

➤ Changes not staged for commit:
modified: Readme.txt ... などと表示される

3. Readme.txt を変更を取り消し

```
$ git checkout Readme.txt
```

4. 変更状況の確認

```
$ git status
```

➤ nothing to commit, working tree clean などと表示される

5. ファイルの内容を確認すると変更前に戻っている

GitHub の仕組み

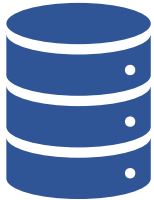
リモートリポジトリ



GitHub のサーバに配置して
複数人で共有するための
リポジトリ

各ユーザが作業するために
自分の手元のマシンに配置する
リポジトリ

ローカルリポジトリ



hoge.html
fuga.py
piyo.js



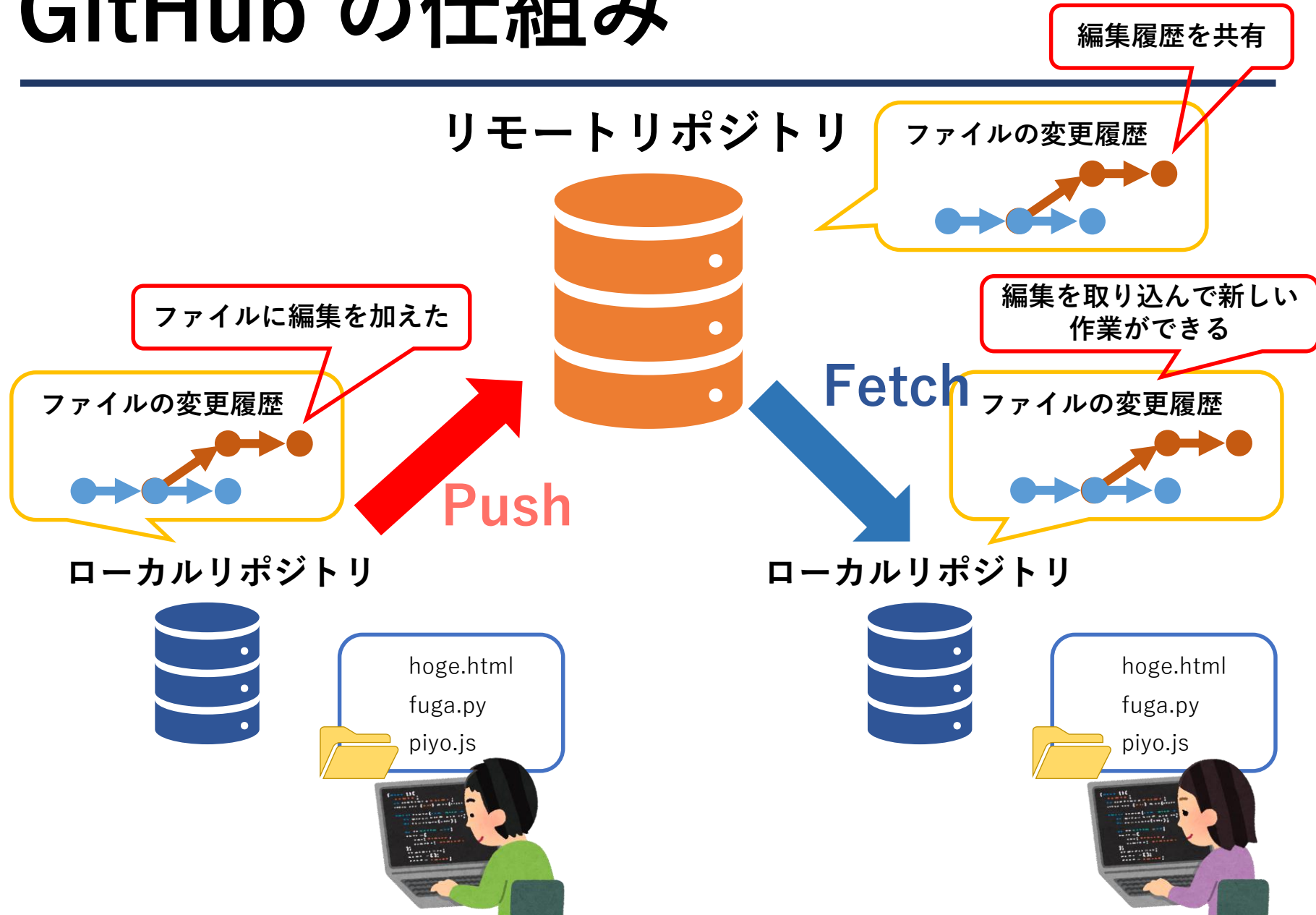
ローカルリポジトリ



hoge.html
fuga.py
piyo.js



GitHub の仕組み



GitHub の仕組み

各自の作業履歴を
リモートリポジトリで共有する
ことで、チームの作業を
整合させながら開発できる

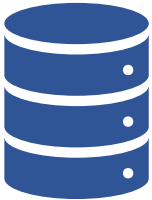
リモートリポジトリ



Push

Fetch

ローカルリポジトリ



hoge.html
fuga.py
piyo.js



Fetch

Push

ローカルリポジトリ



hoge.html
fuga.py
piyo.js



GitHub の仕組み

リモートリポジトリ



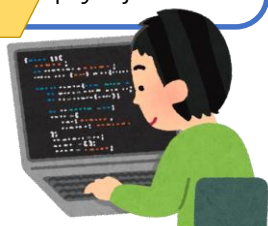
git push コマンド
GitHub へ変更を反映

git fetch コマンド
GitHub から記録を取得

ローカルリポジトリ



hoge.html
fuga.py
piyo.js



ローカルリポジトリ



hoge.html
fuga.py
piyo.js



git merge コマンド
fetch した記録を反映

GitHub からリポジトリを取得 (Clone)

◆ \$ `git clone` リポジトリ名

- リモートリポジトリのファイルと `.git` フォルダをローカルにコピー

ローカルにコピー

リモートリポジトリ



ローカルリポジトリ



`git clone` コマンド

hoge.html
fuga.py
piyo.js



GitHub へ変更を反映 (Push)

1. GitHub 上でリモートリポジトリを新規作成
 - <https://github.com/ユーザ名> にアクセス
 - 右上のアイコンから “Your profile” を開く
 - “Repositories” をクリック → “New” をクリック
 - リポジトリ名を指定 → Create repository を押す
2. リモートリポジトリを登録 (ローカルのフォルダで以下を実行)
\$ `git remote add origin https://github.com/ユーザ名/リポジトリ名.git`
(リポジトリの URL を origin とする)
3. ローカルリポジトリをリモートリポジトリに反映
\$ `git push origin master`
4. GitHub のリモートリポジトリのページをリロードして確認

GitHub の記録を取得 (Fetch or Pull)

◆ リモートの変更をローカルに取り込む (Fetch の場合)

\$ `git fetch origin`

- リモートリポジトリの変更履歴の取得だけを行う
- ローカルのファイルに即反映はしない
- リモートの変更をローカルファイルに反映する場合は

\$ `git merge origin/master`

というコマンドを実行する

◆ リモートの変更をローカルに取り込む (Pull の場合)

\$ `git pull origin master`

- リモートリポジトリの変更履歴の取得を行った後に、それをローカルファイルに即反映する
- Fetch と Merge を同時に行うような感じ

**Fetch の方が安全なので、
初心者は git fetch を使うのがおすすめ**

参考 URL

- ◆サル先生の Git 入門

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

- ◆初心者でもWindowsやMacでできる、Gitのインストールと基本的な使い方

<https://www.atmarkit.co.jp/ait/articles/1603/31/news026.html>

- ◆いまさら Git for Windows のインストール、GitHubに接続してみた。

<https://qiita.com/manabu-watanabe/items/ecf1b434baf305adaa00>

- ◆【Mac】Gitの環境構築をしよう！

<https://prog-8.com/docs/git-env>

- ◆Linux/Unix環境でのGit入門

<http://www.aise.ics.saitama-u.ac.jp/~gotoh/IntroGitOnLinux.html>

参考動画

- ◆ 【Git入門】 サルでも分かるGit入門の前に！Git使い方高速入門編【入門は5分で十分だと思います】

<https://www.youtube.com/watch?v=i1L3A0SLDyg>

- ◆ 【GitHub入門】 初心者向け！GitHubでチーム開発するための基本操作を解説！

<https://www.youtube.com/watch?v=Dz95iUNt-fg>

- ◆ 【Git入門講座 合併版】 この動画1本でGitとGitHubの基礎をゼロからマスター！【初心者向け】

<https://www.youtube.com/watch?v=WHwuNP4kaIU&t=155s>

- ◆ 【GitHub入門】 Webアプリを作りながらGitHub開発フローを習得してみよう

<https://www.youtube.com/watch?v=cOTBf8bHsXo>