アジェンダ

- git の目的。複数人で編集する際のメリット。
- gitの基本手順
- gitの課題提出の流れ
- 課題ex05/6
- unixフォルダ構成

gitを使う目的

- リポジトリに変更内容を記録し、ファイル管理（差分・ファイル変更履歴の移動）する
  - リポジトリとは、ファイルの変更履歴の記録したもの
  - コミットした際に、リポジトリに変更内容が記録される
- 複数メンバーで使う時に真価を発揮するが、ここでは個人の利用に限って説明する。

git変更の記録手順

1. shell00等の課題を提出するための、リポジトリを作る。
   - `git clone <課題提出先のURL>`
   - git cloneが終わったら、`ls -a`コマンドで、`.git`が作られていることを確認する。
2. ファイルを追加、または変更・削除する。
   - 例えば`touch filename`コマンドでファイルを作成する,`ls`コマンドで`filename`が作られていることを確認する
   - このタイミングでgitがファイルの変更を検知するので、`git status`コマンドで操作したファイルが表示されることを確認する
3. 記録したいファイルをステージエリアに乗せる
   - .gitのあるフォルダで、`git add .`または `git add <記録したいファイル名>` する
   -
   - この後、ファイル差分を探す。？
4. 記録したいファイルをコミット（修正内容を確認して、確定する▼）する。
   - `git commit -m "ここにコミットメッセージを書いてください`
5. コミットしたファイルを、リモートレポジトリ（ネットワークの向こうのgit リポジトリ）
   - リモートレポジトリ　⇔　ローカルリポジトリ（自分の作業中のディレクトリ

手順2~4を繰り返します。 手順1は最初だけ実施します。 手順５は課題提出前にだけやれば大丈夫です。

gitの事前設定内容

- ユーザー情報（名前・メアド）を確認▼する。コミット時に利用する。
- 隠しファイルの表示方法を確認する。(ls -a または、エクスプローラで隠しファイルの表示をする)

gitで利用するコマンド

- git add .
- git commit -m "<コミットメッセージを書く>"
- git log：gitのコミットの履歴を確認する。
- git status：git で、`un tracking（

- git push：リモートリポジトリに反映する。（課題提出時に使う）

macで利用するコマンド

- `pwd`:print working directory
    - 自分のいるフォルダのパス（＝カレントディレクトリ）がわかる
- `ls`:list segments
    - `ls -a` :隠しファイルを表示する。.gitフォルダを探せる。
- `cd`:change directory
    - フォルダ移動する
    - 例えば`cd ../`で一個上の親のフォルダに移動
- `ls cd`と組み合わせるコマンド
    - `~` :ホームディレクトリに移動
    - `.` :current directory▼
    - `../` : 一つ上のフォルダへ移動

macフォルダ構成

- ~ : ホームディレクトリ e.g. `cd ~`の時に表示されるフォルダ
- .git のあるフォルダ

TIPS

.git：ターミナルを使わずエクスプローラーを利用している場合は、隠しファイルを表示する設定にする。

今回触れないコマンド

- git checkout
    - ファイルを過去の編集内容に戻す
- git innit
    - gitで編集管理したいフォルダを選ぶ。
- git branch
    - ブランチの移動 — git merge
    - ブランチをマージする
- git push
- git pull
- git fork