非情報系理系院生のための モダンな開発環境づくり

その 1. Git/GitHub を使ったソースコード管理

荒木 亮

阪大院基礎工・後藤研

October 5, 2019

もくじ

目標

ソースコードや LaTeX 文書のバージョンを，Git/GitHub で管理する

- Git に初めて触れる人が，「ファイル名に日付をつけてバックアップ」「ownCloud に全部おいてる」を Git で代替できるようになることをめざす

目標

ソースコードや LaTeX 文書のバージョンを，Git/GitHub で管理する

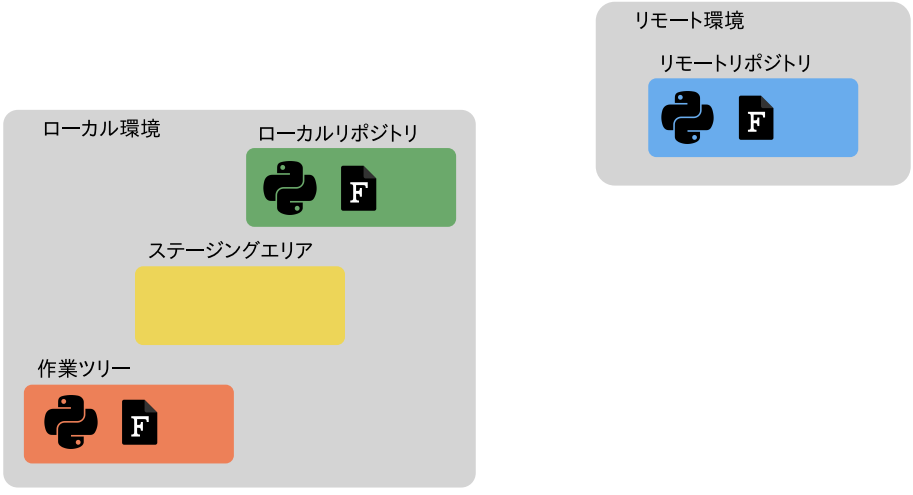
- Git に初めて触れる人が，「ファイル名に日付をつけてバックアップ」「ownCloud に全部おいてる」を Git で代替できるようになることをめざす

説明しないこと

- `branch` を使った同時並行的な開発
- `fork` → `pull request` → `merge` を使ったプロジェクト管理
- `issue`，`Wiki`，`Gist` などの便利な機能

この色の部分は発展的なトピックを扱う

Git の構造



Figures from: icon-icons.com

Git の構造

作業ツリー

作業をおこなうディレクトリ．普通の意味の「フォルダ」

ステージングエリア

作業ツリーで編集したファイルの変更点を「とりあえず」保管しておく場所

リポジトリ

ファイルを修正履歴も含め保存する場所

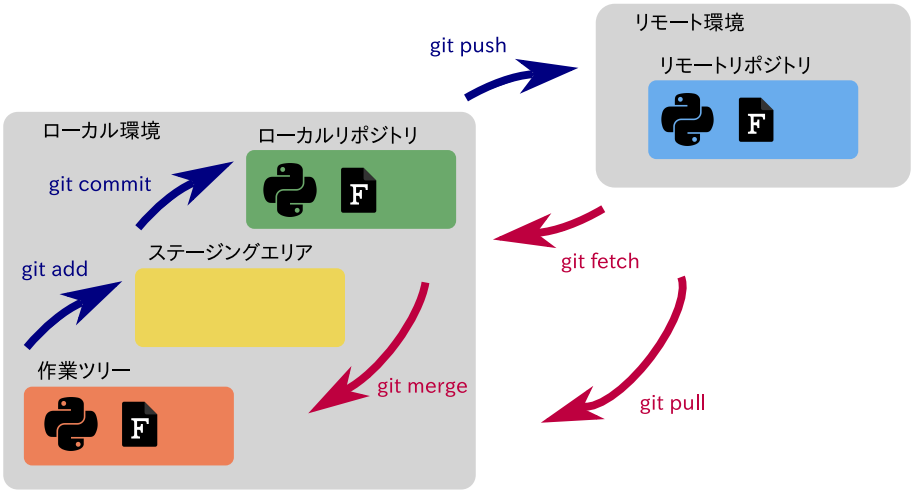
ローカルリポジトリ

ローカルに保存されているリポジトリ
ステージングエリアに登録された変更点の集合を「コミット」として保存する

リモートリポジトリ

サーバで保存，公開されているリポジトリ
ユーザのローカルリポジトリと通信し，情報を更新する

Gitのコマンド



Figures from: icon-icons.com

Gitのコマンド

`git add`

作業ツリーで編集したファイルの変更点をステージングエリアに追加

`git commit`

ステージングエリアにある変更点の集合をローカルリポジトリに記録

`git push`

ローカルリポジトリに記録されたコミット群をリモートリポジトリに送信

`git pull`

作業ツリーをリモートリポジトリの最新の情報で更新

ローカルリポジトリをリモートリポジトリと同期する **`fetch`** と，
作業ツリーとローカルリポジトリを統合する **`merge`** コマンドを一つにしたもの

分散型バージョン管理

Figures from: ownCloud，サルでも分かる Git 入門

自動更新（クラウドサービス）

ローカルの変更箇所を 自動で
リモートに送信

`commit` （Git/GitHub）

変更箇所を 任意の段階でまとめ，
コメントをつけてリモートに送信

- 編集履歴がわかりやすい
- 昔の状態の確認や巻き戻しが簡単
- 編集が競合したとき，解決が簡単

Git はファイル情報をリポジトリ内の
`.git/`以下で管理している

なぜGitを使うのか

- 自動で保存されてしまうクラウドサービスだと…

  「前の状態に戻したいけど，いつのを見ればいいんだろう？」
- 古い状態を記録しておけないと…

  「今は使っていない関数/サブルーチンだけど，また使うかもしれないしとりあえずコメントアウトして残しておこう」
- 共同作業で競合がおきてしまうと…

  「今からこのファイル編集するから触らないでね！」

  「最新のファイルって誰が持ってる？ /どこに置いてる？」

なぜGitを使うのか

- 自動で保存されてしまうクラウドサービスだと…

  「前の状態に戻したいけど，いつのを見ればいいんだろう？」
- 古い状態を記録しておけないと…

  「今は使っていない関数/サブルーチンだけど，また使うかもしれないしとりあえずコメントアウトして残しておこう」
- 共同作業で競合がおきてしまうと…

  「今からこのファイル編集するから触らないでね！」

  「最新のファイルって誰が持ってる？ /どこに置いてる？」

Gitを使うべきでないファイル

- 一度作成したらもう編集しないデータ（実験結果，計算データ）

  →（普通の意味での）バックアップで対処

今日から始めるGit生活

1. GitHubでアカウント作成
2. Github Educationの申請
   ※ 有償アカウントと同等の機能が利用できる
3. 研究用リポジトリの作成
   ※ Private（非公開）に設定する
   ※ プロジェクト単位でリポジトリを作成：全て一つの場所で管理しない
4. リポジトリをローカルに `clone`
5. フォルダにファイルを追加 → `add` → `commit` → `push`
   ※ 見返したときわかりやすいcommitメッセージをつける
6. GitHubでリモートリポジトリを確認し，コードを確認

ssh接続やエディタでGit関連の拡張機能を設定しておくと便利

便利なGitコマンド

`git commit --amend`

一旦ローカルリポジトリに登録したコミットを修正

`git stash`

現在の作業内容を退避し，ステージングエリアの状態を回復
退避した状態は呼び出せる

`git checkout .`

作業ツリーの変更点を破棄，ステージングエリアの状態を回復
ローカルリポジトリに登録された変更点を破棄：**`git reset --hard`**

`git reflog`

Gitのコマンドログを確認

リンク集

読みやすい順

- サルでもわかる Git 入門
- いつやるの？ Git 入門 v1.1.0
- Github 勉強会
- Pro Git book（日本語版）
- こわくない Git

For VSCode ユーザー

- 君には 1 時間で Git について知ってもらう (with VSCode)
- VSCode での Git の基本操作まとめ

何か疑問・質問があれば slack の `#50_github` まで！