コラボレイティブ開発特論

産業技術大学院大学
中鉢 欣秀

2016 年度

コラボレイティブ開発特論

産業技術大学院大学
中鉢 欣秀

第 3 章 Git/GitHub の基本操作

コラボレイティブ開発特論

- 第 3 章 Git/GitHub の基本操作
  - ローカルリポジトリ
  - リモートリポジトリ
  - Git と GitHub の基本操作
  - 演習課題

Git のローカルリポジトリの作成

ローカルリポジトリ

- ソースコードや各種のファイルを保存し，開発に利用する
- 「my_enpit」というディレクトリを作成し，初期化する

コマンド

```
1 mkdir ~/my_enpit
2 cd ~/my_enpit
3 git init
```

Git の設定ディレクトリ

隠しフォルダ「 .git 」

- Git ソースコードの履歴情報や，各種の設定を Git が保存するディレクトリ
- このフォルダは 変更してはならない

確認方法

```
1 ls -a
2 find .git
```

コラボレイティブ開発特論

- 第 3 章 Git/GitHub の基本操作
  - ローカルリポジトリ
  - リモートリポジトリ
  - Git と GitHub の基本操作
  - 演習課題

Hub コマンドが利用可能

- enPiT 環境の Hub コマンド
  - github/hub
- Git への GitHub 操作機能追加
  - 通常の Git の機能に加えて，GitHub 用のコマンドが利用できる
  - コマンド名は「git」のまま（エイリアス設定済み）
- 確認方法

```
1 git version
2 alias git
```

Hub コマンドによるリモートリポジトリの作成

作業内容

- コマンドライン操作で，GitHub にリポジトリを作成する
- Hub コマンドの機能である `git create` を利用
- 初回既動時にはパスワードか聞かれる

コマンド

```
1 git create
```

リポジトリの確認方法

確認方法

- Web ブラウザで GitHub を開き，「 my_enpit 」ができていることを確認

コマンドラインで確認

```
1 git remote -vv
```

コラボレイティブ開発特論

- 第 3 章 Git/GitHub の基本操作
  - ローカルリポジトリ
  - リモートリポジトリ
  - Git と GitHub の基本操作
  - 演習課題

マニュアル等

- Git - Documentation

commit ログの書き方

- Writing good commit messages · erlang/otp Wiki

ステータスの確認

リポジトリの状態を確認する

- `git status` は，頻繁に利用するコマンド
- リポジトリの状態を確認することができる
- この表示の読み方を理解することが重要

コマンド

```
1 git status
```

ファイルの追加とステータスの確認

作業内容

- テキストエディタで README.md を作成
- ステータスの変化を見る

コマンド

```
1 emacs README.md
2 git status
```

Add/Commit の方法

ステージングエリアを利用する場合

- git add README.mb
- git commit -m 'First commit'

ステージングエリアを省略する場合

- git commit -a -m 'First commit'
  - トラックされていないファイルは commit しないので注意

リモートリポジトリへの公開

push とは？

- ローカルで作成した commit を，リモートのリポジトリにアップロードすること
- origin とは，リモートのリポジトリの内部的な名前
- upstream とは，ブランチ（後述）が紐づいているリポジトリのこと
- 最初にそのブランチを push するときは，`--setupstream` オプションを指定

コマンド

```
1 git push --set-upstream origin master
```

Log の閲覧

コミットログ

- ソースコードに加えた変更の履歴を，commit を単位として閲覧できる

コマンド

```
1 git log
```

コミットのログを詳細に書く方法

エディタを使ったログの記述

- コミットのログや，Pull Request の記述を，より詳しく書くことができる
- `commit` や `pull_request` から `-m` オプションを外すと，エディタが立ち上がる
  - エディタは emacs を起動するようになっている
  - `C-x C-s` で保存，`C-x C-c` で終了

コマンド

```
1 git commit
```

Git の参考資料

コラボレイティブ開発特論

- 第 3 章 Git/GitHub の基本操作
  - ローカルリポジトリ
  - リモートリポジトリ
  - Git と GitHub の基本操作
  - 演習課題

演習課題 2-1

Init/Status/Add の練習

1. 解説した手順に従い，my_enpit リポジトリを作成
2. git status コマンドを実行
3. README.md ファイルを作成しなさい
4. git status コマンドを実行し，変化を見なさい
5. commit しなさい．ログを必ず書くこと
6. git status コマンドを実行し，変化を見なさい

演習課題 2-3

Commit/Log/Push の練習

1. README.md を修正して commit しなさい
2. 新しいファイルを作成して commit しなさい
3. 作業が完了したら，push しなさい
   （`--set-upstream` が必要）
4. コミットが push されていることを Web ブラウザで確認しなさい
5. 作成したファイルを削除して commit して push しなさい
6. エディタを使って，詳細なログを書きなさい
7. その他，自由に commit の作業を試しなさい

課題の提出

提出物

- 下記のものを提出してください
  - GitHub と Heroku アカウント
  - 作成した my_enpit リポジトリの URL

提出先

- [コラボレイティブ開発特論 (2015) アカウント等]