

# 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 4, 2019

# もくじ

目標

Git の構造

Git のコマンド

分散型バージョン管理

なぜ Git を使うのか

今日から始める Git 生活

便利な Git コマンド

リンク集

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

- ▶ Git 初心者が，「ファイル名に日付をつけてバックアップ」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
- ▶ `pull request` を使ったチームでの開発
- ▶ その他色々ややこしいコマンド

- ▶ 作業ツリー
- ▶ ステージングエリア
- ▶ ローカルリポジトリ
- ▶ リモートリポジトリ

を表す図を追加

作業ツリー

block

ステージングエリア

block

ローカルリポジトリ

block

リモートリポジトリ

block

# Git のコマンド

```
git add
```

block

```
git commit
```

block

```
git push
```

block

```
git pull
```

block

# なんでこんな難しいんや：分散型バージョン管理

クラウドサービスと Git の  
差異を表す図

commit  
block

# なぜ Git を使うのか

`git_tutorial.md` から資料を作成する



# 今日から始める Git 生活

1. GitHub でアカウント作成
2. Education plan の作成
3. 研究用リポジトリの作成
4. コードを保存しているディレクトリを登録
5. ファイル編集 → `add` → `commit` → `push`
  - ※ 見返してわかりやすい「commit メッセージ」をつける
6. GitHub のページを確認し，コードが変更されていることを確認

# 便利な Git コマンド

```
git checkout .
```

block

```
git stash
```

block

その他, `git revert` `git reset`

git コマンドへのオプション, `.gitignore`

```
git commit --amend
```

block

```
git reflog
```

block

`git_tutorial.md` から資料を作成する