



# Git 勉強会 2 日目

～基本コマンド & ハンズオン ①～

2022/03/25

# 2日目アジェンダ(今日はこっち)

- Git 基本コマンド ① ~個人開発編~

- init
- clone
- switch (checkout)
- remote
- branch
- status
- add
- commit
- push
- log
- diff
- config

## 3 日目アジェンダ

- Git 基本コマンド ② ~チーム開発編~

- pull
- fetch
- merge
- rebase
- stash
- restore (checkout)
- reset
- revert
- cherry-pick
- blame
- tag
- reflog

# Git勉強会 - 基本コマンド & ハンズオン

## init

### 機能

- Git ローカルリポジトリを作成する
  - .git ディレクトリ(フォルダ)が作成される

### ユースケース

- Git でバージョン管理を始めたい

### コマンド例

```
$ git init
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

-

# Git勉強会 - 基本コマンド & ハンズオン

## clone

### 機能

- リポジトリのクローンを作成する

### ユースケース

- 既に存在するリモートリポジトリ(GitHub/GitLab)のソースコードをローカルに複製したい

### 主なオプション

- b : 複製したいブランチを指定する(b: branch)

### コマンド例

```
$ git clone <リモートリポジトリの URL>
$ git clone <リモートリポジトリの URL> -b <ブランチ>
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

-

## Git勉強会 - 基本コマンド & ハンズオン 機能

- 作業ツリーを異なるブランチに切り替える

### ユースケース

- 既に存在するブランチに移動したい
- 新しいブランチを作成したい

### 主なオプション

- `-c` : ブランチを新規作成して切り替え(c: create)

### コマンド例

```
$ git switch <ブランチ> # git checkout <ブランチ> と同じ
$ git switch -c <ブランチ> # git checkout -b <ブランチ> と同じ
```

### 備考

- `switch` は Git バージョン 2.23.0 でリリース (2019/08/16)

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- [git-switch – Git コマンドリファレンス\(日本語版\)](#)
- [git checkout の代替としてリリースされた git switch と git restore](#)

## 機能

- リモートリポジトリを関連付けする

## ユースケース

- リモートリポジトリ(GitHub / GitLab)を登録(削除)したい
- リモートリポジトリの名前と場所(URL)を確認したい
  - Fork したリポジトリなど、リモートリポジトリが複数ある場合に便利

## 主なオプション

- v : 詳細を表示(v: verbose)

## コマンド例

```
$ git remote add origin <リモートリポジトリのURL>
$ git remote -v
$ git remote remove <リモートリポジトリのURL>
```

## イメージ

## 参考

- [git-remote Documentation](#)
- [【 git remote 】コマンド\(基礎編\)——リモートリポジトリを追加、削除する](#)

## branch

### 機能

- ブランチを作成、削除、一覧表示する

### ユースケース

- ローカルリポジトリにあるブランチを確認したい
- ローカルリポジトリにあるブランチの名前を変えたい
- ローカルリポジトリにあるブランチを削除したい

### コマンド例

```
$ git branch
$ git branch -m <変更前のブランチ名> <変更後のブランチ名> # 今いるブランチ上で git branch -m <変更後のブランチ名> でも OK
$ git branch -d <ブランチ名> # git branch -D <ブランチ名> で強制削除
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- [git-branch – Git コマンドリファレンス\(日本語版\)](#)
- [git branch コマンド - Qiita](#)

# Git勉強会 - 基本コマンド & ハンズオン

## status

### 機能

- ワークツリーにあるファイルの状態を表示する

### ユースケース

- 

### 主なオプション

- `-s` : 短い形式で表示(s: short)

### コマンド例

```
$ git status  
$ git status -s
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- [git-status – Git コマンドリファレンス\(日本語版\)](#)
- [git status -s でちょっと幸せになれる - Qjita](#)

## add

### 機能

- ファイルをインデックスに追加する(コミットの対象にする)

### ユースケース

- 変更したファイルをコミットしたい
- コンフリクトを解決したファイルをコンフリクト解決済の状態に変更したい

### コマンド例

```
$ git add ./src/index.html # ./src/index.html のみインデックスに追加  
$ git add ./src/ # ./src ディレクトリ以下の全てのファイルをインデックスに追加  
$ git add . # 変更済の全てのファイルをインデックスに追加
```

## イメージ

## 参考

- [git-add – Git コマンドリファレンス\(日本語版\)](#)
- [【git add】コマンド——変更内容をインデックスに追加してコミット対象にする](#)

## commit

### 機能

- インデックスに追加した変更をリポジトリに記録する

### ユースケース

- ・

### 主なオプション

- ・

### コマンド例

```
$ git
```

イメージ

参考

- 
-

## push

### 機能

- ローカルリポジトリの変更内容をリモートリポジトリに送信する

### ユースケース

- ・

### 主なオプション

- ・

### コマンド例

```
$ git
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- 
-

## Git勉強会 - 基本コマンド & ハンズオン

### log

#### 機能

- コミット時のログを表示する

#### ユースケース

- 

#### 主なオプション

- 

#### コマンド例

```
$ git
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- 
-

## diff

### 機能

- コミット同士やコミットと作業ツリーの内容を比較する

### ユースケース

- 

### 主なオプション

- 

### コマンド例

```
$ git
```

# Git勉強会 - 基本コマンド & ハンズオン

イメージ

参考

- 
-

# Git勉強会 - 基本コマンド & ハンズオン

## Git勉強会 - 基本コマンド & ハンズオン

Fin.