



# GitHubについて

## GitとGitHubの基本的なワークフロー

新規プロジェクトから始めよう

Gitのリポジトリを作成する

```
>_ターミナル  
-$git init
```

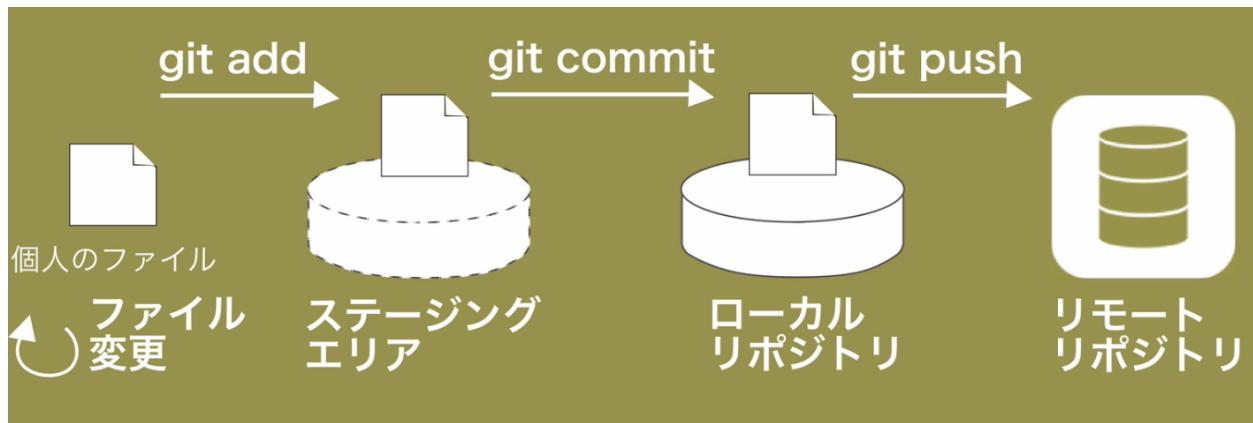
例：

```
>_ターミナル  
  
$cd /Users/[name]/Documents    フォルダーに移動  
  
$cd [フォルダ名]    intro_gitに移動  
  
$ls    ファイルの情報を表示する  
  
$git init    gitのリポジトリを作る  
  
$ls -a    全てのファイルを表示する (aはallという意味)  
  
$ls -a .git/  全ての.gitを表示する
```

Gitの基本的なワークフロー

## 基本的なワークフロー

1. ファイルの変更をステージングエリアへ追加する
2. ローカルリポジトリにコミットする
3. リモートリポジトリにプッシュする



```
>_ターミナル
```

```
$git add exe01.txt  ステージングエリアへ追加する  
$git commit  リポジトリに記録／保管する
```

## Vimエディタの使い方

1. ターミナルでgit commitを入力する
2. Vimエディタが立ち上がる
3. 半角英数字に入力を切り替える
4. 「i」を入力する（挿入モードになる）
5. コミットメッセージを入力する
6. 「esc」を押す（ノーマルモードに戻る）
7. 「:wq」を入力してエンターを押す

(commitメッセージを保存してエディタを閉じる)

## 変更をコミットしよう

### 変更をコミットする流れ

ファイルを変更したら

1. ステージングエリアへ追加
2. リポジトリにメッセージを付けてコミット



### ステージングエリアへ追加

```
>_ターミナル  
-$git add [ファイル名]  
-$git add .
```

### コミットして変更を記録しよう

```
>_ターミナル
```

```
~$git commit
```

## 分かりやすいコミットメッセージを書こう

1行目：変更内容の要約

2行目：空行

3行目：変更した理由

## Code

```
>_ターミナル  
  
$git status 状態を確認する  
  
$git add. ステージングエリアへ追加する  
  
$git commit -v 変更したものを記録する  
  
$git log 履歴を繰り返す
```

## 色々な変更をコミットしよう

- ・新規作成
- ・ファイルの削除
- ・複数ファイルの変更

```
>_ターミナル
```

```
[1]
```

```
$git add exe02.txt  
  
$git commit -v
```

```
$ls  
[2]  
$rm exe02.txt    ファイルを削除する  
$ls  
$git rm exe02.txt  
$git commit -v  
[3]  
$git add .  
$git commit -v
```

## 現在の状況を確認しよう

git statusコマンドでGitリポジトリの状況を表示する

```
>_ターミナル  
-$git status
```

## 変更履歴を確認しよう

git logコマンドでリポジトリにコミットされたログを確認する

### 変更履歴の確認

```
>_ターミナル  
-$git log
```

## 色々なオプション

```
>_ターミナル

#一行で表示する
-$git log --oneline

#ファイルの差分を表示する
-$git log -p index.html

-$git log -n 3 最新のコミット3つまで表示する

-$git log --oneline -n 3 一行列で最新のコミット3つまで表示する
```

## 変更差分を確認しよう

git diffコマンドで、ファイルの変更差分を表示する

### 変更差分を表示する

```
>_ターミナル

-$git diff
```

\* diffはdifferenceの略

### add, commitする前に確認しよう

```
>_ターミナル

#ステージとの差分
-$git diff

#ステージとコミットの差分
-$git diff HEAD
```

# GitHubにプッシュしよう

## リモードリポジトリに登録

git remote addコマンドで、リモートリポジトリをローカルリポジトリに登録できる

```
>_ターミナル  
~$git remote add origin https://github.com/user/repo.gi
```

\*今後はoriginという名前でGitHubリポジトリにアクセスできる

## リモードリポジトリへ送信

git pushコマンドで、ローカルリポジトリの内容をリモートリポジトリへ送信する

```
>_ターミナル  
#登録したGitHubへ送る  
~$git push origin master
```

## master以外のブランチへ送信

```
>_ターミナル  
#featureブランチを送る  
~$git push origin feature
```

## 管理しないファイルをGitの管理から外す

.gitignoreファイルに指定することで、ファイルをGitの管理から外すことができる

### .gitignoreファイルの書き方

```
# #からまる行はコメント  
# 指定したファイルを除外 index.html  
# ルートディスクトリを指定 /root.html  
# ディスクトリ以下を除外 dir/
```

## コミットしてしまったファイルを管理から外す

git rmコマンドで、コミットしたファイルをGitの管理から削除できる

### ファイルごとGitの管理から削除

```
>_ターミナル  
  
#ファイルも  
#ディレクトリも一緒に削除  
-$git rm -r [ディレクトリ名]
```

### ファイルは残る場合

```
>_ターミナル  
  
-$git rm --cached [ファイル名]
```

### ブランチを切り替える

```
>_ターミナル  
  
-$git checkout [ファイル名]
```

### 元の状態に戻す

```
>_ターミナル
```

```
~$git reset HEAD [ファイル名]
```

## 参考

- <https://git-scm.com/book/ja/v2>
- <https://www.atlassian.com/ja/git/tutorials/what-is-version-control>