

はじめてのGitとGitHub



Gitのよく使うコマンド集

ローカルリポジトリの新規作成

リポジトリとは、ファイルやディレクトリの状態、変更履歴を記録する場所。
"git init" コマンドでリポジトリを新規作成する。

ローカルリポジトリの作成

>_ ターミナル

```
~ $ git init
```

initializeの略だよ

.gitディレクトリが作成される



俺がファイルを
管理するぜ

ステージングエリアへの変更追加

複数ファイルを変更したときに、コミットするファイルを選択するためにあるのがステージングエリア。

ステージングエリアへの追加

>_ ターミナル

```
~ $ git add [ファイル名]
```

```
~ $ git add .
```

追加するということだよ

コミットの前にaddする



コミットする
ファイルを選
択してね

変更の記録

変更メッセージを付けてリポジトリに記録するのがコミット。コミットすることで変更がリポジトリ内に時系列で記録される。

リポジトリに変更を記録する

>_ ターミナル

```
~ $ git commit
```

メッセージ付きで記録するよ

ひとつの作業ごとに1コミット



どの作業を何の
ためにしたのか、
すぐ振り返られる
ようにしといてね

現在の状況の確認

git statusコマンドでGitリポジトリの状況を表示する。

現在の状況の確認

>_ ターミナル

~ \$ git status

状況を確認するよ

よく使うコマンドだよ



git statusコマンドで現在の状況を確認しながら作業するよ

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
commit 3b4f3aafe
Author: kiyodori <~@~>
Date:   Fri Feb 10
    first commit
```

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
```

変更差分の確認

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

differenceの略だよ

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

リモートリポジトリの登録

git remote addコマンドで、リモートリポジトリを登録できる。

リモートリポジトリの登録

>_ ターミナル

```
~ $ git remote add origin https://github.com/user/repo.git
```

今後はoriginという名前でGitHubリポジトリにアクセスできるよ

リモートリポジトリへ送信

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。

リモートリポジトリへ送信

>_ ターミナル

```
~ $ git push origin master
```

登録したGitHubへ送れるよ

master以外のブランチへ送信

>_ ターミナル

```
~ $ git push origin feature
```

featureブランチを送るよ