

# 守護神でもできる プログラミング講座

No.9  
D3 村尾響

1

## 今日の目標

- GitとGithubを1人で開発する範囲で使えるようになろう

# OUTLINE

1. Git・Githubとは
2. Git・Githubの下準備
3. Gitを使ってみよう
4. Githubの機能
5. まとめ

## 1. Git・Githubとは

### Gitとは



Gitは、プログラムのソースコードなどの変更履歴を記録・追跡するための管理システム。

(Wikipediaより)

要するに「バージョン管理システム」

<https://git-scm.com/>

# 1. Git・Githubとは

## Githubとは



GitHubは、Gitの仕組みを利用して、世界中の人々が自分の作品(ソースコードやデータなど)を保存、公開することができるようにしたウェブサービスの名称

プログラマの採用項目に含まれることもある。

<https://github.com/>

# 1. Git・Githubとは

できるプログラマならGitとGithubは必須項目！  
Githubを中心に面接をしている会社もある。

<https://findy-code.io/engineer-lab/job-change-github-employment-company>



**GitとGithubをやるしかない！**

## 2. Git・Githubの下準備

### ① Gitのインストール

WindowsはこちらからGitをダウンロード

<https://git-scm.com/download/win>

しばらく待つと  
ダウンロードされるが、  
開始されないときは  
これをクリック



## 2. Git・Githubの下準備

### ① Gitのインストール

WindowsはこちらからGitをダウンロード

<https://git-scm.com/download/win>

しばらく待つと  
ダウンロードされるが、  
開始されないときは  
これをクリック



## 2. Git・Githubの下準備

### ① Gitのインストール

インストールの項目を忘れたので  
チェックしながらやりますorz

## 2. Git・Githubの下準備

### ① Gitのインストール

Macの人は <https://tracpath.com/bootcamp/git-install-to-mac.html> を参考に

ターミナルで

**\$ git --version**

と入力するとインストールされていなかったら  
ポップアップが出るのでインストールする

## 2. Git・Githubの下準備

### ① Gitのインストール

Linux(Linux Mint)の人はターミナルで

```
$ sudo apt install git
```

と入力してインストール

## 2. Git・Githubの下準備

### ② Githubの登録

<https://github.co.jp/> から登録する

<https://qiita.com/koohei/items/361da3c9dbb6e0c7946b> を参考にしよう

わかりやすいです

ユーザー名は採用会社の面接で見られるかもしれません  
~~ふざけた名前はやめましょう~~

## 2. Git・Githubの下準備

### ② Githubリポジトリの作成

<https://qiita.com/kooohai/items/361da3c9dbb6e0c7946b> を引き続き参考に、リポジトリを作成する

### リポジトリとは？

ソースコード(とは限らない)の保管場所

## 2. Git・Githubの下準備

### ③ Gitの環境設定

ターミナルに

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email sample@sample.com
```

とGithubに登録した名前とメールアドレス入力し情報をセットする

### 3. Gitを使ってみよう

今回は以下のことを実践してみる

- Githubリポジトリのクローン
- ローカルリポジトリのコミット
- プッシュ、プル
- ログの確認

### 3. Gitを使ってみよう

- Githubリポジトリのクローン

#### クローンとは

他人のGithubリポジトリを自分のPCにダウンロードすること

Githubで使いたいリポジトリが見つかったら

**\$ git clone リポジトリのURL**

をターミナルに入力する



### 3. Gitを使ってみよう

- ・ ローカルリポジトリのコミット

#### ローカルリポジトリとは



PCの中に作成するリポジトリ(ファイル置き場)

#### リモートリポジトリ

Githubなどの(Web)サーバー上に作成するリポジトリ

### 3. Gitを使ってみよう

適当な場所でターミナルを開き、

```
$ git init
```

でローカルリポジトリを作成する

.gitというフォルダができるので同じ階層に

[index.html](#)を作成、次のスライドをコピーして保存する

### 3. Gitを使ってみよう

#### index.html

```
<!doctype html>

<html lang="ja">
<head>
  <meta charset="utf-8">
  <title>新しいタブ</title>
</head>

<body>
  <h1>見出し</h1>
  <p>本文</p>
</body>
</html>
```

### 3. Gitを使ってみよう

**\$ git add index.html**

と入力してローカルリポジトリにindex.htmlを追加する。続いて

**\$ git commit -m “メッセージを書いてください”**

でindex.htmlをリポジトリに登録する

このときに””内にファイルをなぜ追加したか、どのような変更を行ったかなどのメッセージを記入する

### 3. Gitを使ってみよう

全てのファイルをまとめてaddしたい時は

`$ git add .` または `$ git add *` を用いる

addしたりcommitしたりしたファイルをリポジトリから除外したいときは

`$ git rm --cached ファイル名`

と入力する

ファイルごと消したいときは

`$ git rm ファイル名`

### 3. Gitを使ってみよう

#### ・プッシュ、プル

##### プッシュとは

ローカルリポジトリをリモートリポジトリに反  
映させること

##### プルとは

リモートリポジトリをローカルリポジトリに反  
映させること。プッシュの反対の動作

### 3. Gitを使ってみよう

まずはリモートリポジトリの登録

**2. Git・Githubの下準備**で作成したリモートリポジトリのページを開く

そのURLをコピーして

```
$ git remote add origin URL名
```

でリモートリポジトリを登録する

### 3. Gitを使ってみよう

```
$ git push origin master
```

でローカルリポジトリのコミットをGithub上のリモートリポジトリに反映させる

(アップロードのようなもの)

ローカルリポジトリの.gitフォルダ以外すべて削除して

```
$ git pull origin master
```

とするとリモートリポジトリからファイルがダウンロードされ、元の状態になる

## 3. Gitを使ってみよう

### ・ログの確認

```
$ git log
```

と入力するとコミットメッセージ等のログを確認することができる

```
$ git log --name-status
```

と入力して変更したファイルを表示できる

logのオプションはいろいろあるので確認してみよう

## 4. Githubの機能

### リポジトリのフォーク

他人が作成したリポジトリを自分のアカウントにコピーする

気になったリポジトリのURLを開いて右上のForkボタンを押すだけ。自分のアカウントに追加される

## 4. Githubの機能

### 課題管理

GithubのIssuesのページで開発中のプロジェクトで問題になっているところを管理できる

↓こんな感じ

<https://github.com/oblivionmgd/ARMap/issues>

## 4. Githubの機能

### Github.io

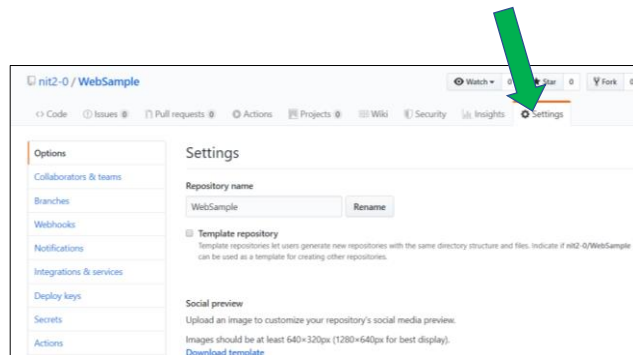
HTML、CSS、JavaScriptのファイルのみで動作するリポジトリが対応

対象のリポジトリをHttpサーバー化してサイトを公開することができる

RubyやPHPなどのサーバーサイドのコードは扱えない

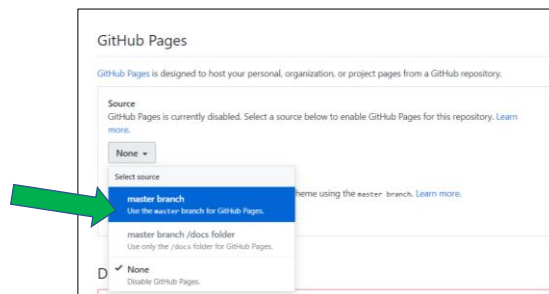
## 4. Githubの機能

先ほどpushしたGithubのリポジトリのページを開いて、**Settings**タブを開く



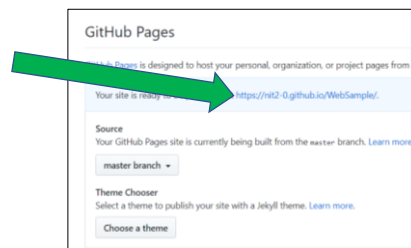
## 4. Githubの機能

スクロールしていったら**GitHub Pages**を探し、**Source**を**None**から**master branch**に変更する



## 4. Githubの機能

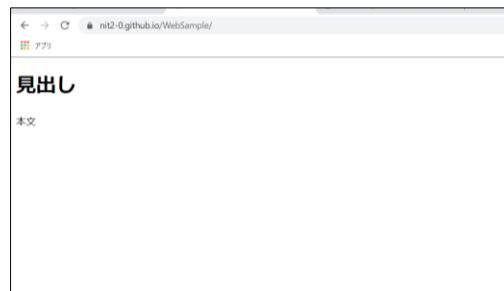
更新された後にもう一度GitHub Pagesに戻ると、先ほど更新したSourceの上にURLが出現するのでこれをクリックする



## 4. Githubの機能

以下のようなページが出てくるのを確認する

404エラーになったときはGithub上でファイルを適当に変更してcommitすると解決するらしい...





## 5. まとめ

- Gitを用いるとソースコードのバージョン管理が楽になる
- GithubはGitの機能を用いてソースコードなどを公開することができるサイトである
- Githubには静的Webページを作成できる機能がある