

(/)

【Windows】Gitの環境構築をしよう！

目安時間：20分

この記事では、WindowsでGitを使うための環境構築方法を説明します。

記事を読みながら実際に手を動かして、ローカル環境にGitの環境を準備していきましょう！

○ 必要なもの・知識

- Windowsが搭載されたPC
- ターミナル操作に関する基礎知識
- テキストエディタ

([HTML & CSS の開発環境を用意しよう！](#)
(<https://prog-8.com/docs/html-env>) を参考にしてください。)

このページはProgateの[Git学習コース](#) (<http://prog-8.com/git/study/1/1#/0>) と [CommandLine学習コース](#) (<http://prog-8.com/commandline/study/1/1#/0>) の知識を前提としています。

この記事はWindowsをご利用の方向けの記事です。macOSをご利用の方は[こちらの記事](#) (<https://prog-8.com/docs/git-env>) をご利用ください。

(L)

1. Gitのインストール

まずは、以下のサイトからGitをダウンロードしましょう。

<https://git-for-windows.github.io/>
(<https://git-for-windows.github.io/>)

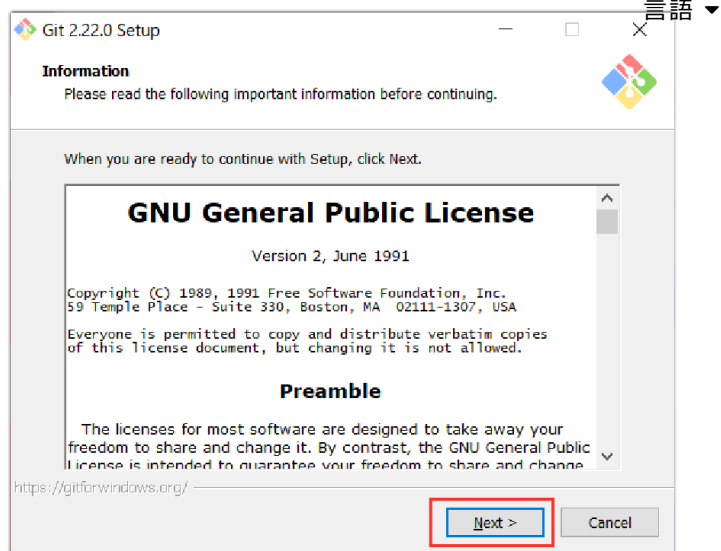


git for windows の画面

ダウンロード完了後、ファイルを展開するとインストール画面が出てきますので、以下の手順でインストールしていきましょう。

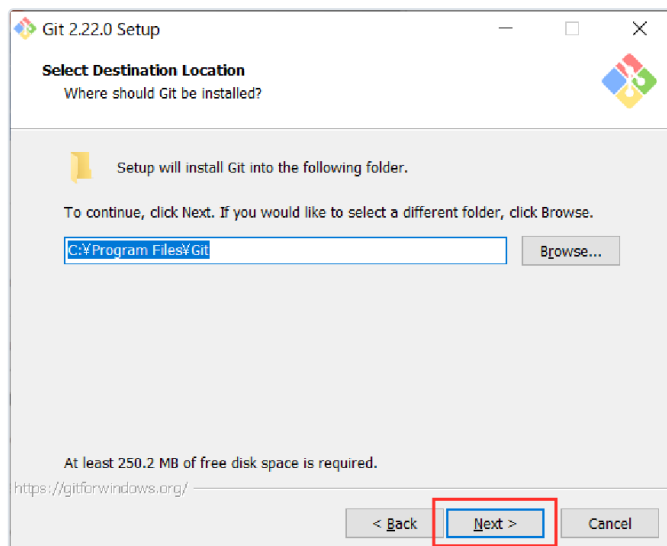
1. 内容に同意したら、Nextをクリックしてください。

(L)



内容に同意したら、*Next*をクリック

2. *Next*をクリックしてください。



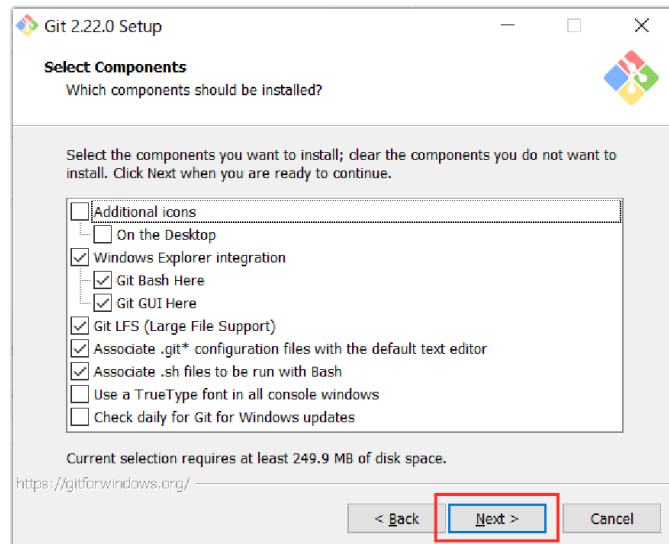
*Next*をクリック

3. Nextをクリックしてください。

言語 ▾

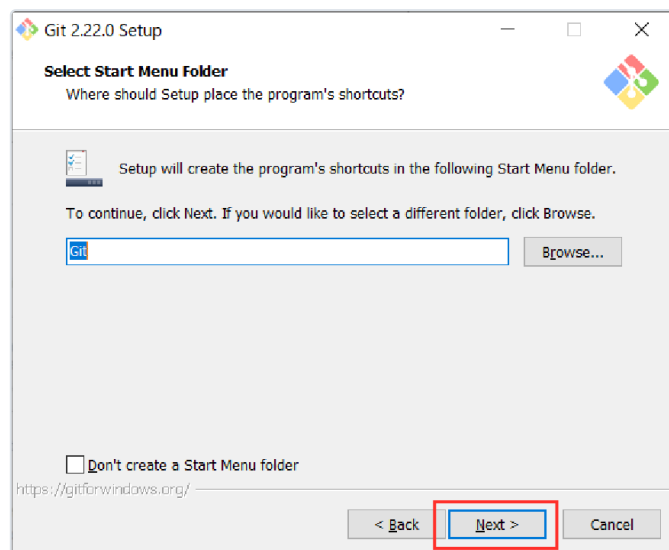


(L)



Nextをクリック

4. Nextをクリックしてください。



Nextをクリック

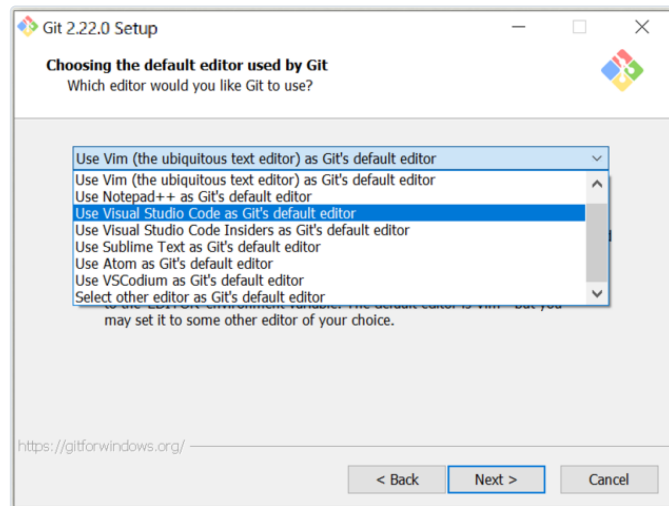
5. エディタの設定をします。使用するエディタの設定を選んでNextをクリックしてください。

(下の画像はVSCodeの例です)

言語 ▾

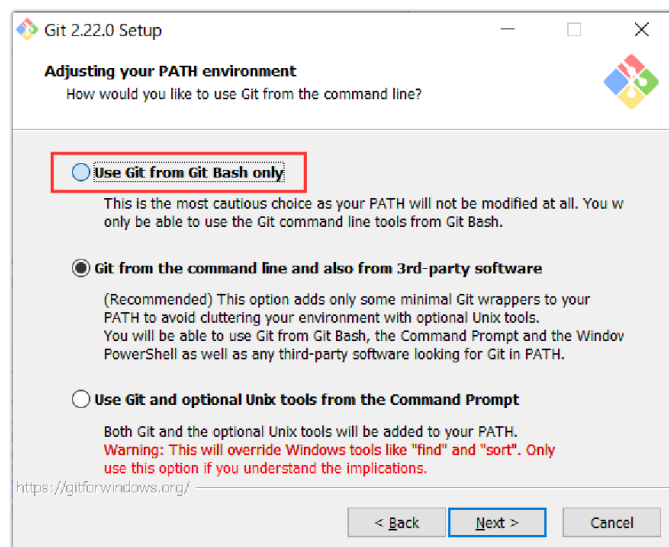


(L)



使用するエディタの設定を選んでNextをクリック

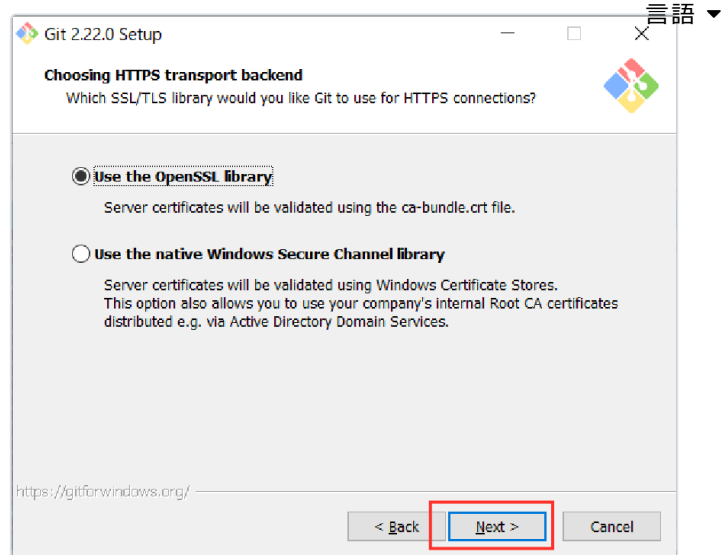
6. Use Git from Git Bash onlyにチェックを入れてNextをクリックしてください。



Use Git from Git Bash onlyにチェックを入れてNextをクリック

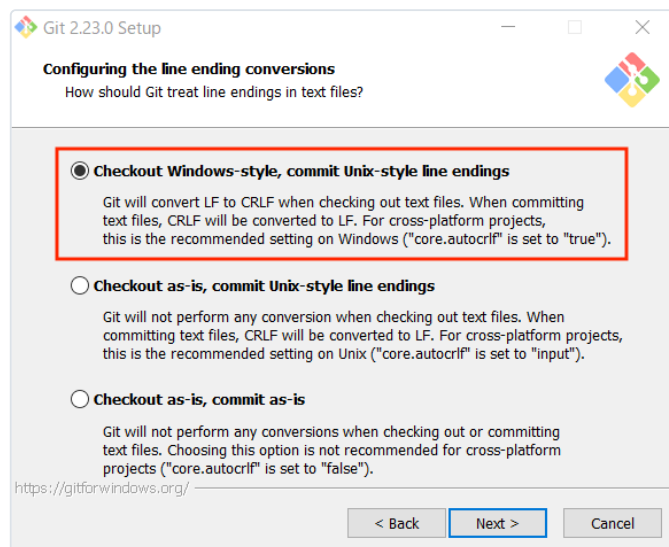
7. Use the OpenSSL library にチェックを入れてNextをクリックしてください。

(L)



Use the OpenSSL library にチェックを入れて
Nextをクリック

8. Checkout Windows-style, commit Unix-style line endingsにチェックを入れてNextをクリックしてください。



Checkout Windows-style, commit Unix-style line endings にチェックを入れてNextをクリック

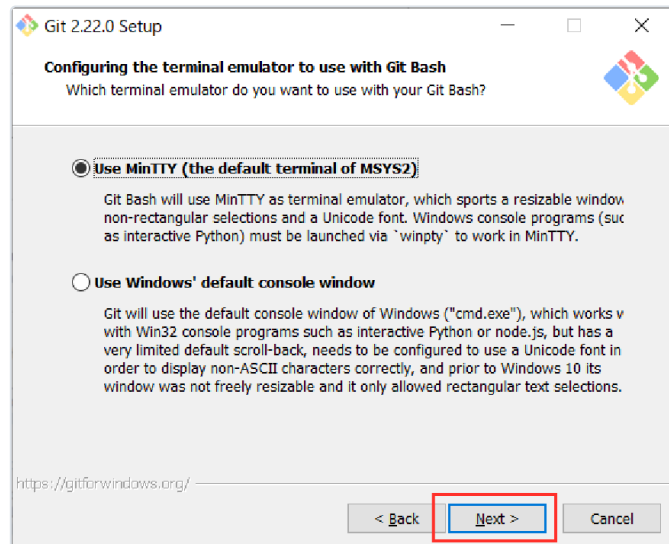
9. Use MinTTY (the default terminal of MSYS2)を選択して、Nextをクリックしてく

ださい。

言語 ▾

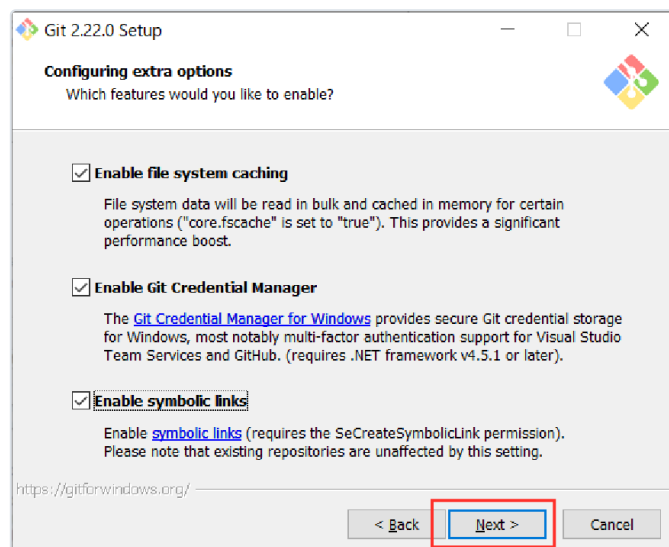


(L)



*Use MinTTY (the default terminal of MSYS2)*を選択して、*Next*をクリック

10. 全てにチェックを入れて、*Next*をクリックしてください。



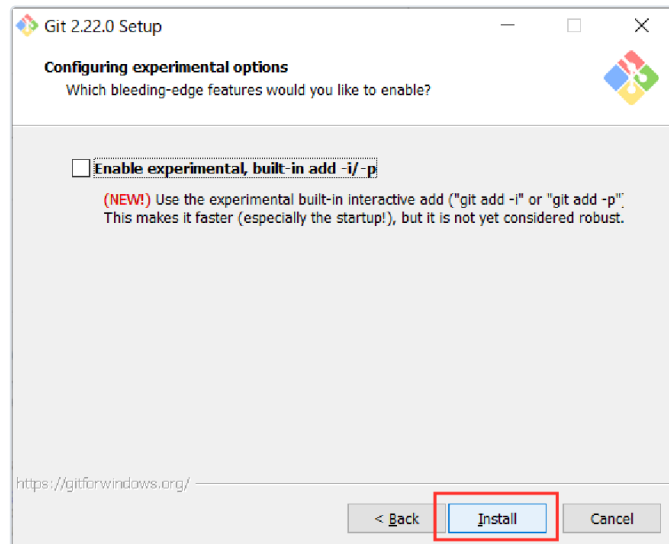
全てにチェックを入れて、*Next*をクリック

11. Installをクリックしてください。

言語 ▼

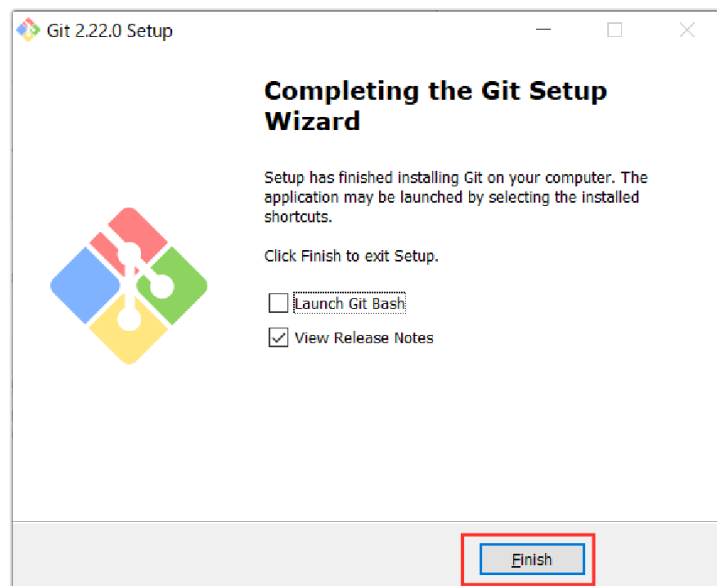


(L)

*Install*をクリック

これで、Gitのインストールが完了します。

Finishをクリックして、終了です。

*finish*をクリックで終了

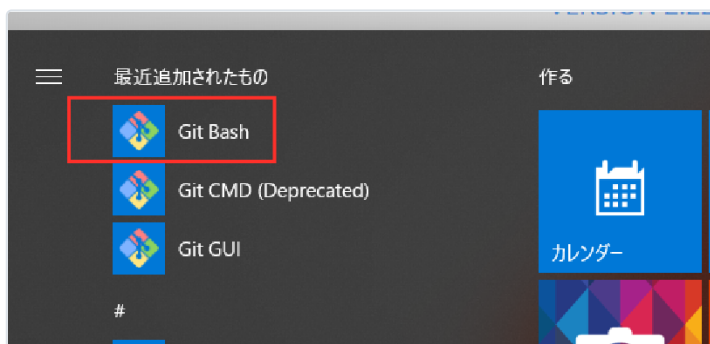
PCに「Git Bash（以下のアイコン）」というアプリがインストールされているはずですので、
これからの作業はGit bashを使用してください

いい！

言語 ▾



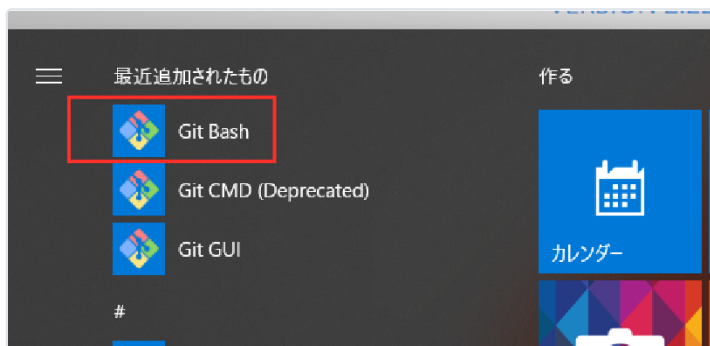
(.).



*Git bash*がインストールされている

2. Gitの初期設定

まずは、Git Bashを起動してください。



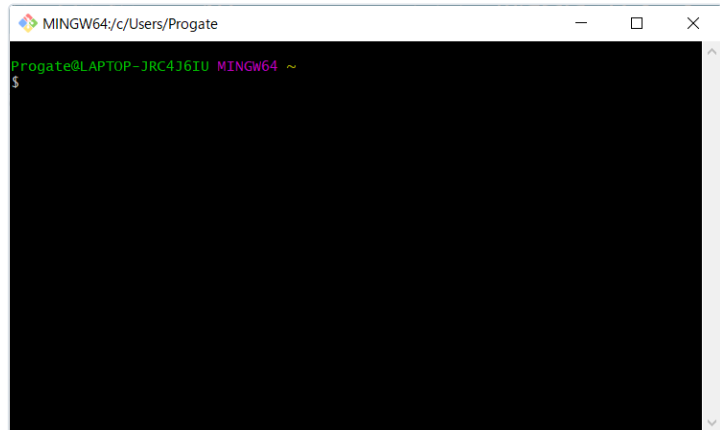
*Git bash*を起動する

起動すると、このような画面になります。

言語 ▾



(L)



Git Bashを起動した画面

では、Git Bashを使用して初期設定を行なっていきましょう。

ユーザー名とメールアドレスを設定する

Gitにユーザー名とメールアドレスを設定することで、コミットしたときに誰が行なったコミットかを記録することができます。

以下の\$マークより後のコマンドを実行して設定してください。

(ユーザー名、メールアドレスは自分で自由に決めてください。)

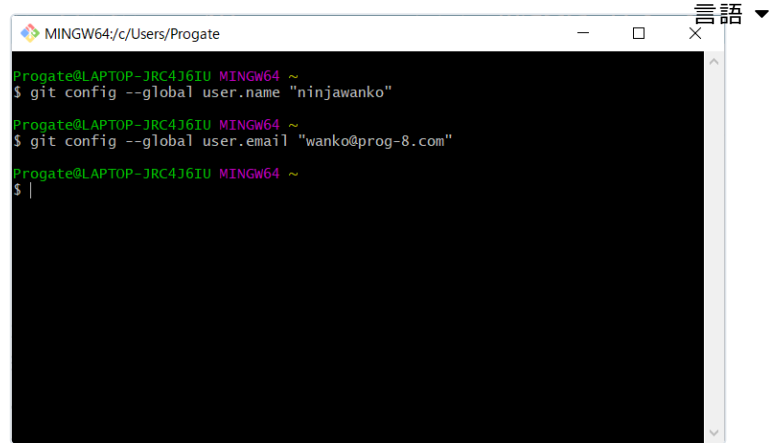
```
$ git config --global user.name "ユーザー名"
```

```
$ git config --global user.email "メールアドレス"
```

以下は例です。

☒ ご意見箱

(/)



```
MINGW64/c:/Users/Progate
Progate@LAPTOP-JRC4J6IU MINGW64 ~
$ git config --global user.name "ninjawanko"
Progate@LAPTOP-JRC4J6IU MINGW64 ~
$ git config --global user.email "wanko@prog-8.com"
Progate@LAPTOP-JRC4J6IU MINGW64 ~
$ |
```

Gitの初期設定をする

これで初期設定は完了です！

3. Githubの設定

ここからは、Githubというサービスを利用して、リモートリポジトリを作成する方法を見ていきましょう。

「GitコースI」で「ひつじ仙人」が用意していたリモートリポジトリを自分で作成していきます。

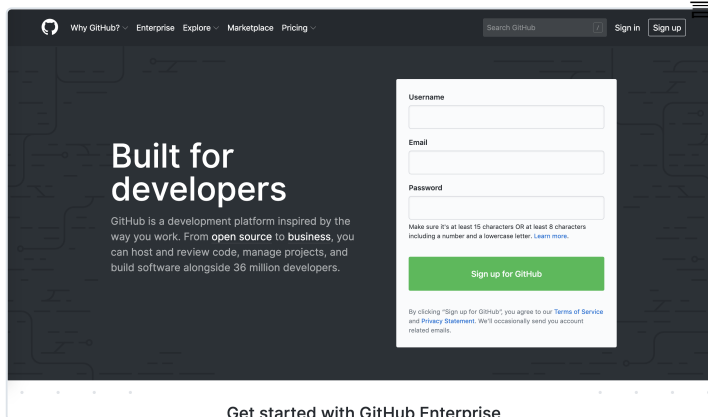
Githubアカウントの作成

Githubを利用するために、アカウントを作成しましょう。

以下のリンクからGithubにアクセスしてください。

Github (<https://github.com/>)

(L)

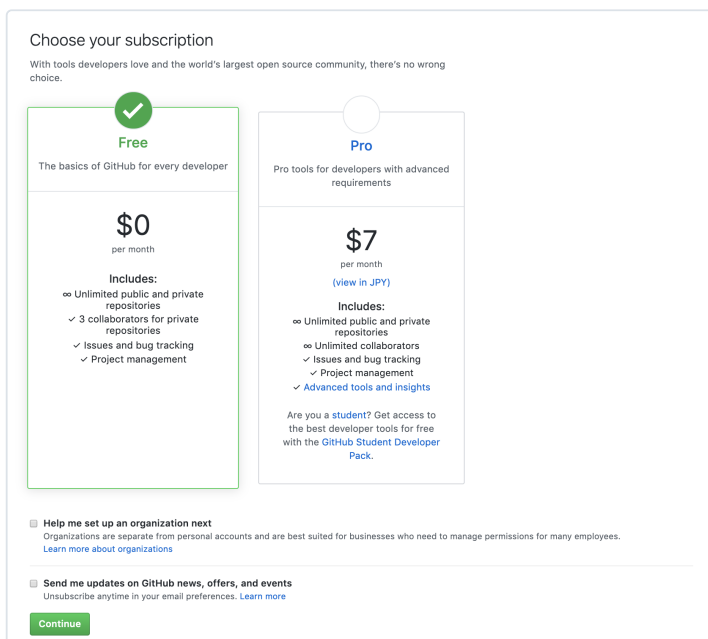


アクセスしたときの画面

新規登録フォームにユーザ名、メールアドレス、パスワードを入力して登録をしましょう。

ユーザー名、メールアドレス、パスワードの登録後、指示に従い進めていくと、プラン選択に移ります。

無料で使う場合は、「free」と書いてある項目を選択します。選択後、一番下の「continue」ボタンを押します。



プラン選択画面

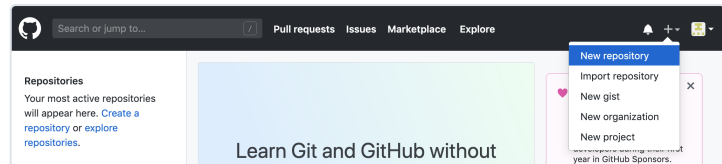
(L)

プラン選択後、いくつか質問が出てきますが答えなくても問題ありません。最後に「Submit」ボタンを押して登録を完了することができます！

登録が完了すると、Githubから認証メールが届きますので、そのメール内のURLをクリックすることで、作成したアカウントで作業を始めることができます。

リモートリポジトリの作成

リモートリポジトリを作成するには、Githubのヘッダー右上にある「+」をクリックします。



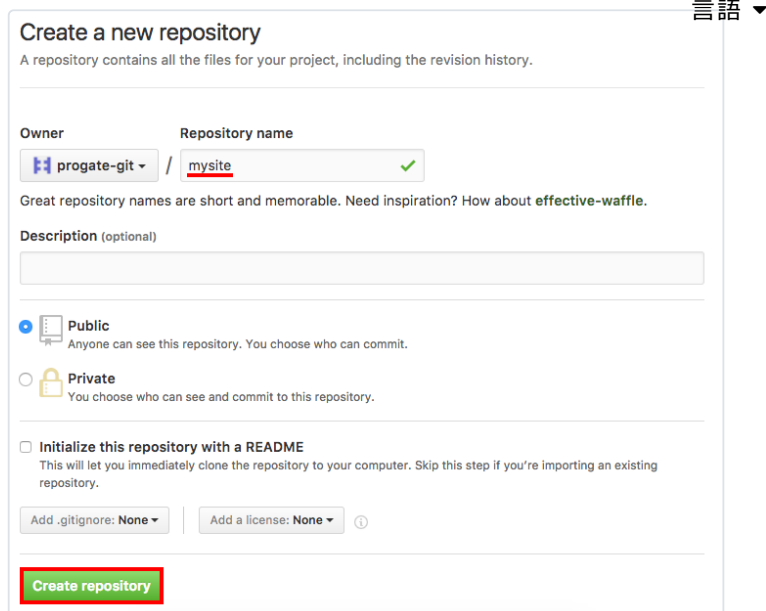
新規リモートリポジトリの作成

「+」をクリックして表示される項目の「New repository」をクリックします。

「New repository」をクリックすることで、以下の画像のページに移ります。

「Repository name」の箇所にプロジェクト名を入力して「Create repository」を押すと、リモートリポジトリを作成することができます。

(L)



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: **progate-git** / Repository name: **mysite** ✓

Great repository names are short and memorable. Need inspiration? How about **effective-waffle**.

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

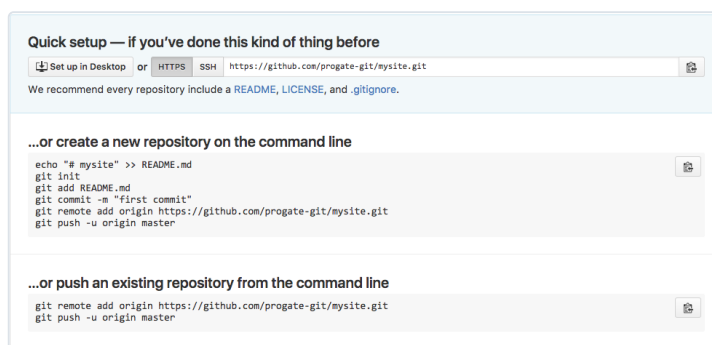
☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** Add a license: **None** ⓘ

Create repository

新規リモートリポジトリの詳細を入力

リモートリポジトリの作成が完了すると、以下の画像のページに移ります。



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/progate-git/mysite.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# mysite" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/progate-git/mysite.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/progate-git/mysite.git
git push -u origin master
```

リモートリポジトリが作成される

4. Githubにプッシュする

Githubで作成したリモートリポジトリにプッシュしてみましょう。

Gitを使う準備をする

言語 ▾

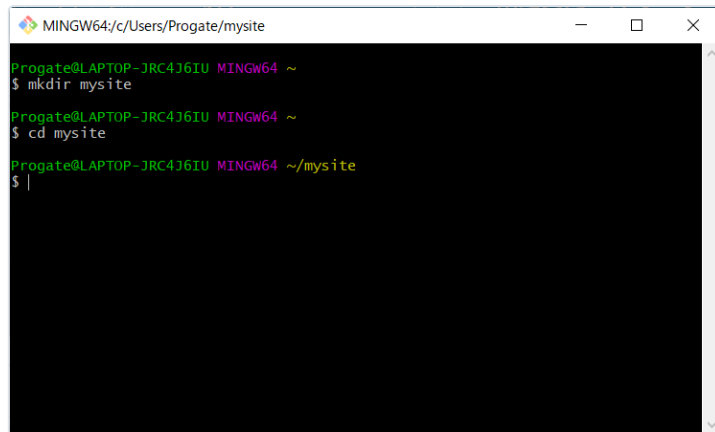


(\).

まずはGitで管理するディレクトリを作成し、移動します。以下のコマンドを Git Bash上で実行してください。

```
$ mkdir mysite
```

```
$ cd mysite
```



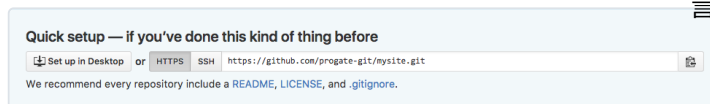
コマンド実行後

そのフォルダをGitで管理できるように、以下のコマンドを実行します。

```
$ git init
```

`git init` を実行した後、下のhttpsから始まる1行をコピーしてください。
(下の画像の右にあるボタンをクリックすると簡単にコピーできます。)

(/)

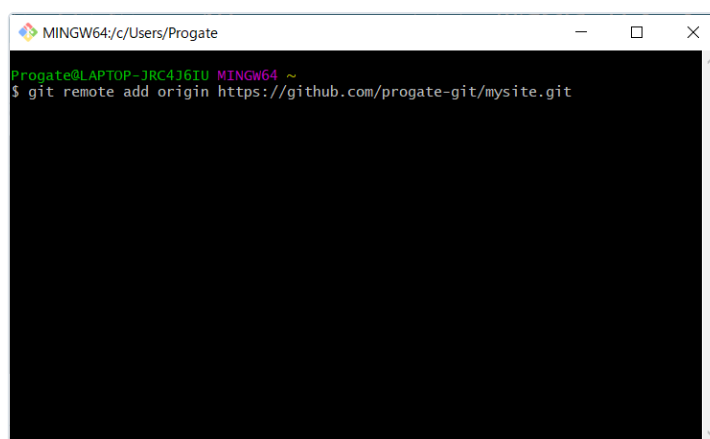


コピーする

コピーできたら <URL> の部分コピーしたものに置き換えて実行してください。

```
$ git remote add origin <URL>
```

このコマンドは、プッシュ先のリモートリポジトリを指定するコマンドです。



リモートリポジトリを設定する

プッシュする

次に、プッシュするためにファイルを作成してコミットしましょう。

```
$ touch index.html
```

```
$ git add index.html
```



```
$ git commit -m "Create index.html"
```

言語 ▼



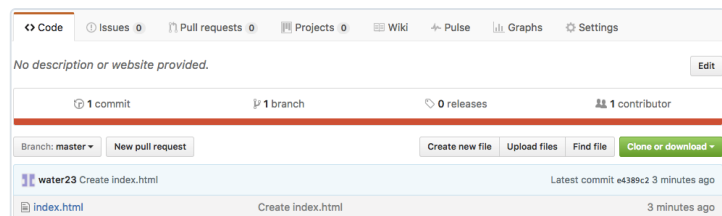
(L)

ここまでできたらプッシュします。

```
$ git push origin master
```

プッシュできているかをGithubで確認してみましょう。

プッシュができていればGithub上にindex.htmlが作成されているはずです。



プッシュできているか確認する

5. うまくいかない時のチェックリスト

うまくいかない場合は、以下のような間違いがないかチェックしてみましょう。

- 記事で指定したコードが貼り付けられていない
- 入力したファイル名やソースコードなどに誤字脱字がある
- ファイルが保存されていない

- 記事で指定している箇所にフォルダが配置されていない 言語 ▼
- コマンドの内容や実行している場所が違う



(L)

これでGithubを用いてプロジェクトを管理できるようになりました！

GitとGithubを用いることで、共同開発やバージョン管理をスムーズに行うことができます。これから開発を始めるときには、ぜひ活用してみてください！

SNSでシェアしよう



[コース一覧へ \(/courses\)](/courses)