

# Gitを普通に使う

## 目次

[目次](#)

[共有したいフォルダを作成しましょう](#)

[この中に共有したいフォルダを入れます](#)

[GitHubに行きます ～リポジトリ作成～](#)

[Git Bashに戻る](#)

[1.Shift + Insert](#)

[2.Windows key + v](#)

[本題に入ります](#)

[git status → git add ファイル名](#)

[git statusで確認](#)

[ファイルをまとめてAddしよう](#)

[git add のコマンド紹介](#)

[git commit -m "メッセージ"](#)

[git push](#)

[Githubへ](#)

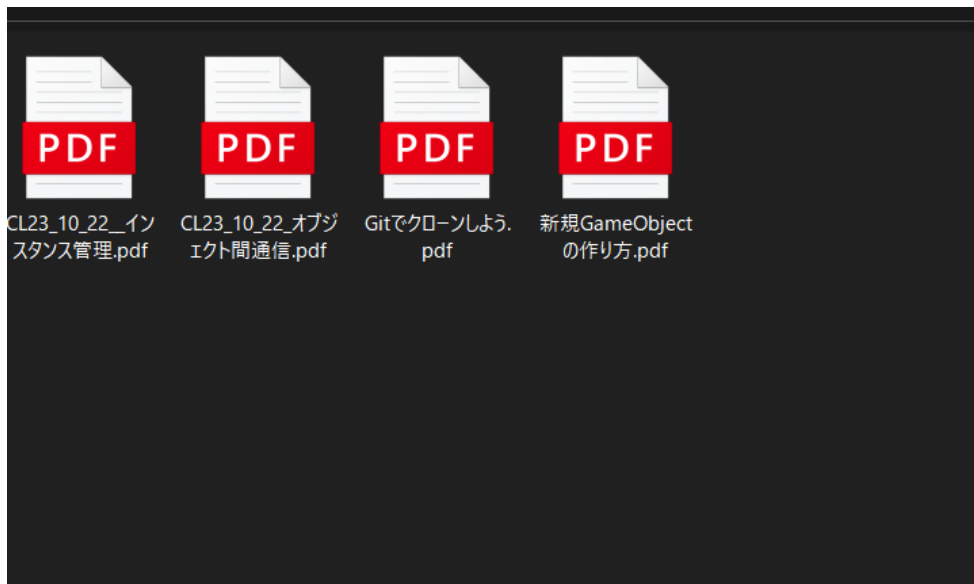
[更新したいときは？](#)

[Gitで管理できるもの](#)

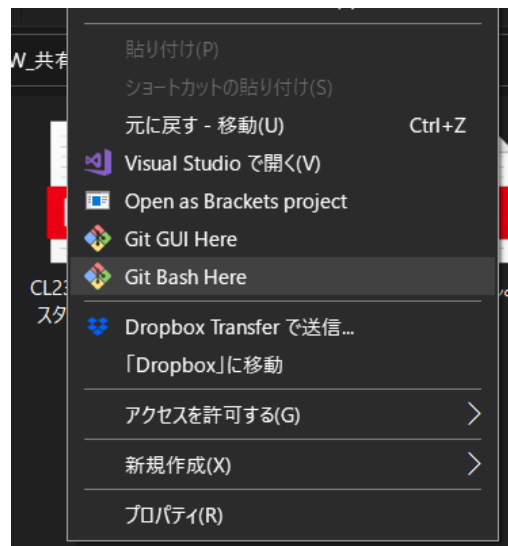
## 共有したいフォルダを作成しましょう



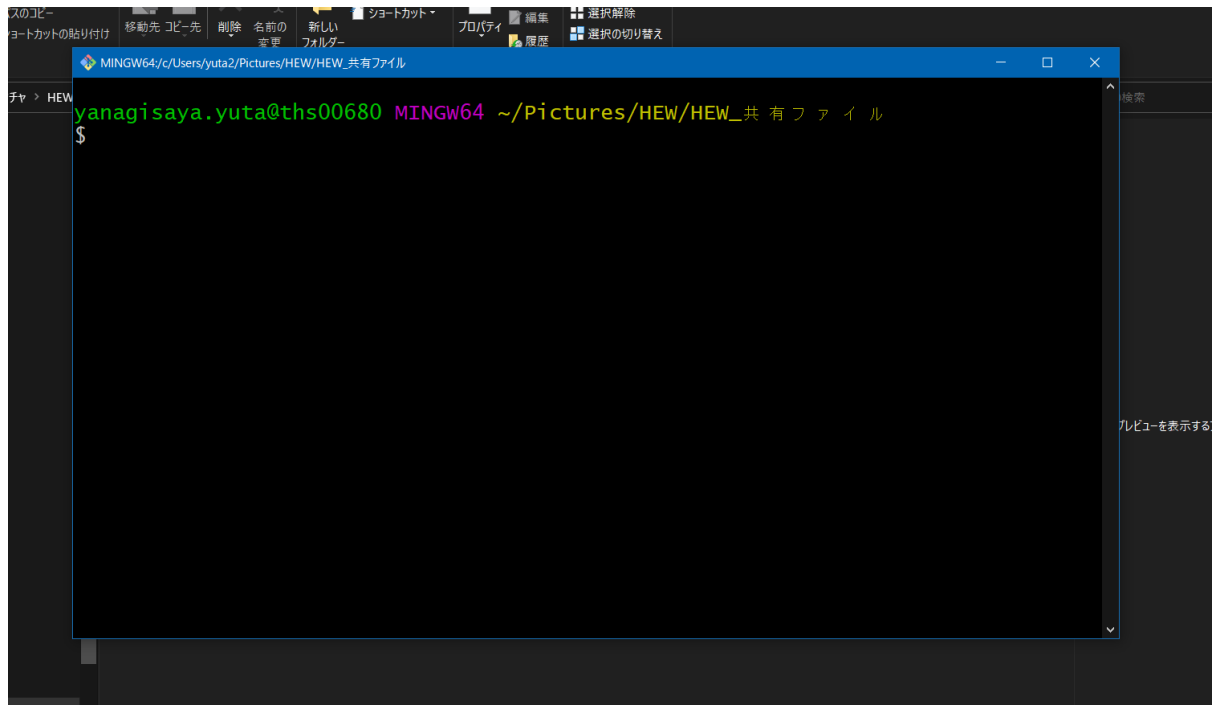
この中に共有したいフォルダを入れます



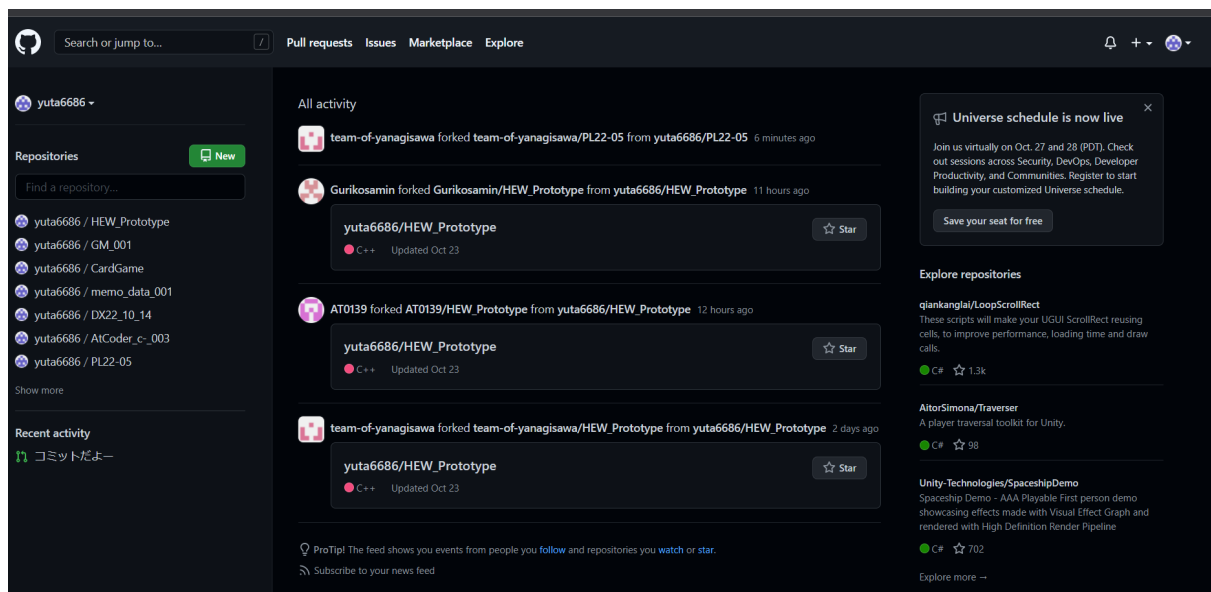
ここで右クリックしてGit Bash Hereを選択



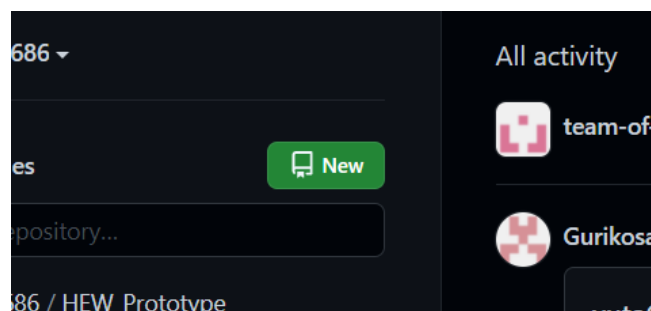
こうなりますね



## GitHubに行きます ～リポジトリ作成～



こんな感じになってます。そしたら、**緑色のNew**を押しましょう



はい。こうなります

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* Repository name \*

yuta6686 /

Great repository names are short and memorable. Need inspiration? How about [psychic-train](#)?

Description (optional)

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

ここで必要なのは名前

リポジトリをみんなに共有するPublicか

リポジトリを自分だけで使用するprivateかを選択する。

今回はみんなに共有したいのでPublicで行きます

Owner \*      Repository name \*

yuta6686 / HEW\_共有ファイル ✓

Great repository names are short, lowercase, and contain only numbers, lowercase letters, and hyphens. Your new repository will be created as HEW\_-. How about **psychic-train**?

Description (optional)

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

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

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

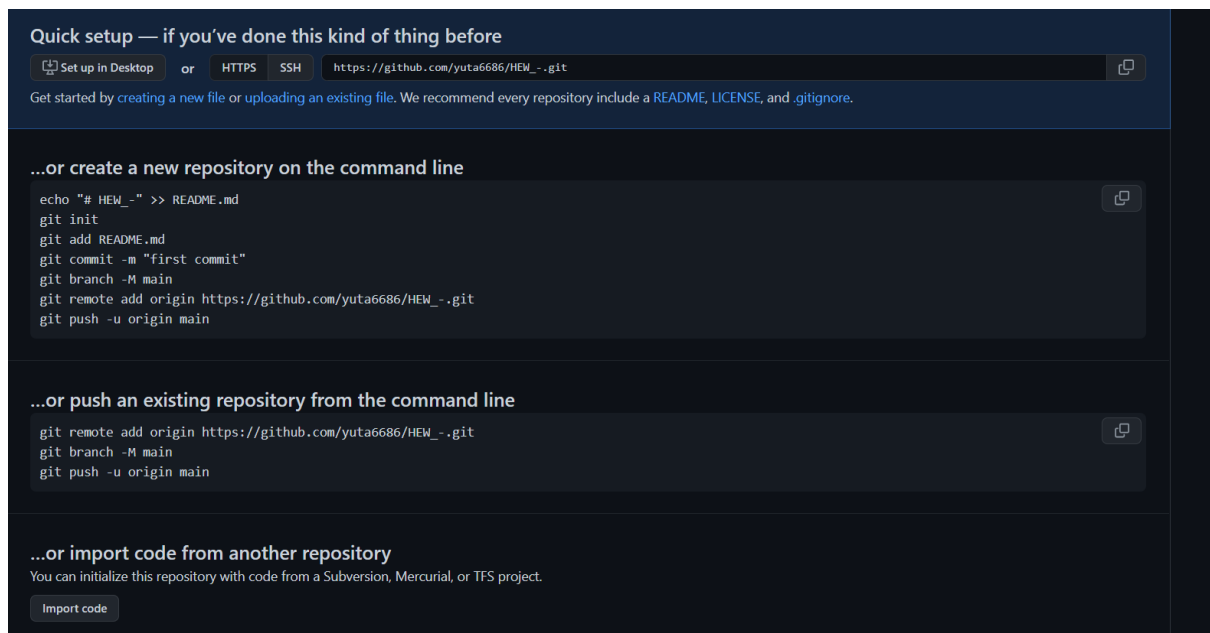
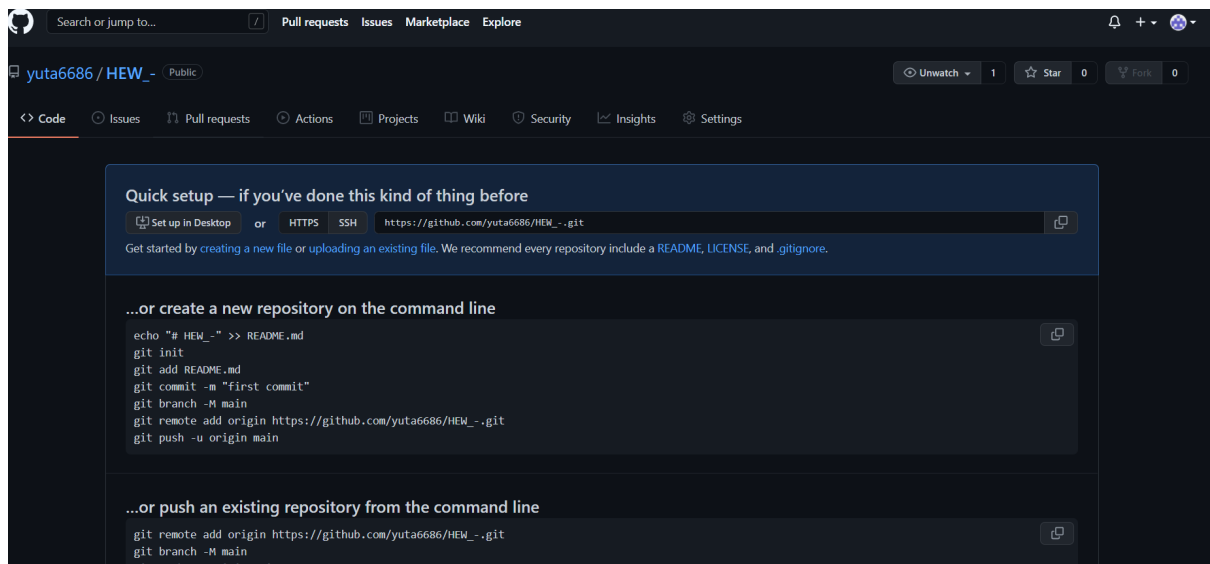
**Create repository**

名前をつけたら。Create repositoryを押しましょう

☐ **Choose a license**  
A license tells others what they can and can't do with your code.

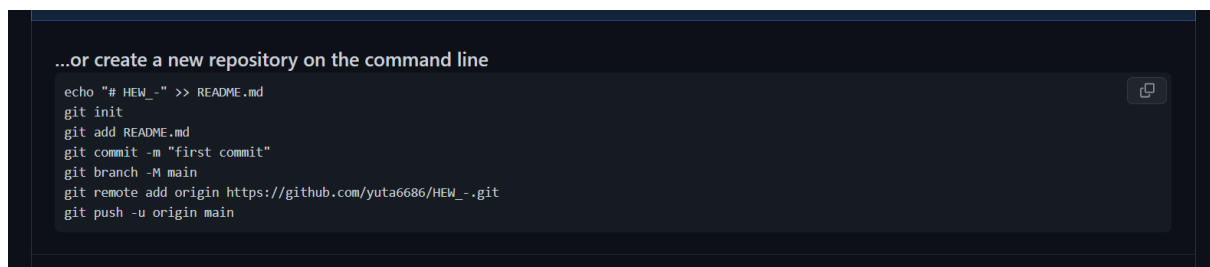
**Create repository**

作成できたら、こんな画面になってますね

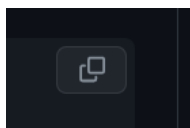


なんだこれ。何が書いてあるんだ。

まあ、とりあえず



ここの



これを押してコピーしましょう！

## Git Bashに戻る

ここでさっきコピーしてきたやつを貼り付けます。

でも、CTRL + V が使えない！！！！

はい。2つやり方があります

### 1.Shift + Insert

1をmsiノートPCでやる場合、DeleteキーとInsertキーが合体してます。

なので、Shift + Fn + Insertを押しましょう

ちなみにDelete/Insertはmsiだと右奥にあります

### 2.Windows key + v

2はクリップボードの機能を使用しています。

便利なので普段使いしましょう。

ペーストするといっぱいやってくれます。

```
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (master)
$ echo "# HEW_" >> README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/yuta6686/HEW_-.git
git push -u origin main
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (master)
$ git init
Reinitialized existing Git repository in C:/Users/yuta2/Pictures/HEW/HEW_共有ファイル/.git/

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (master)
$ git commit -m "first commit"
[master (root-commit) dee36fa] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (master)
$ git branch -M main

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git remote add origin https://github.com/yuta6686/HEW_-.git

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

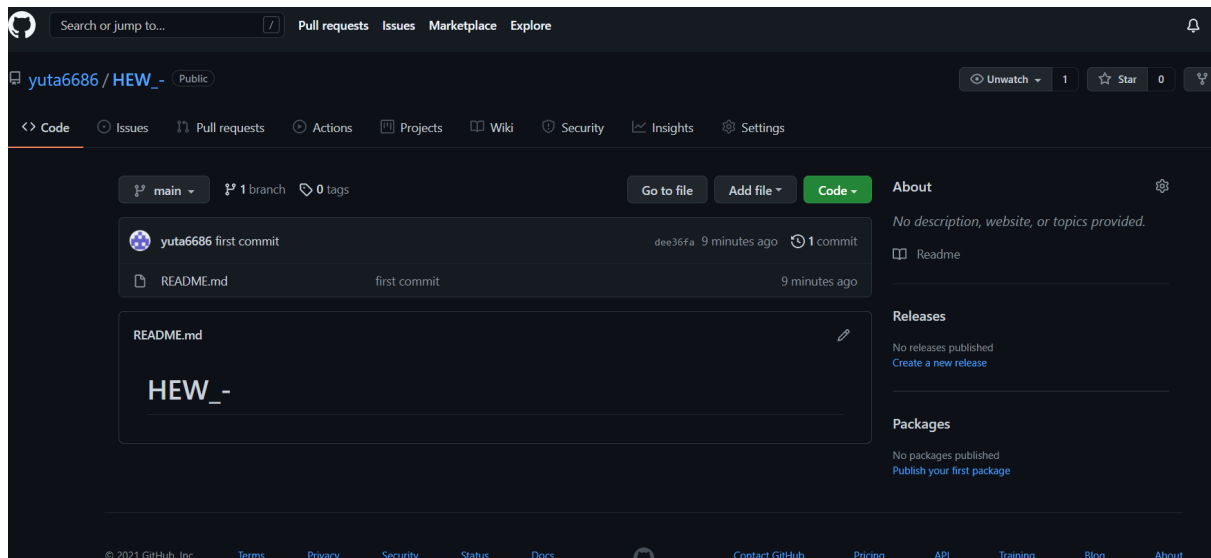
初期化→ README.mdファイルの追加→Commit → ブランチ作成→リモートの設定→プッシュ

↑これは覚えなくてもいいです。

勝手に色々やってくれました。

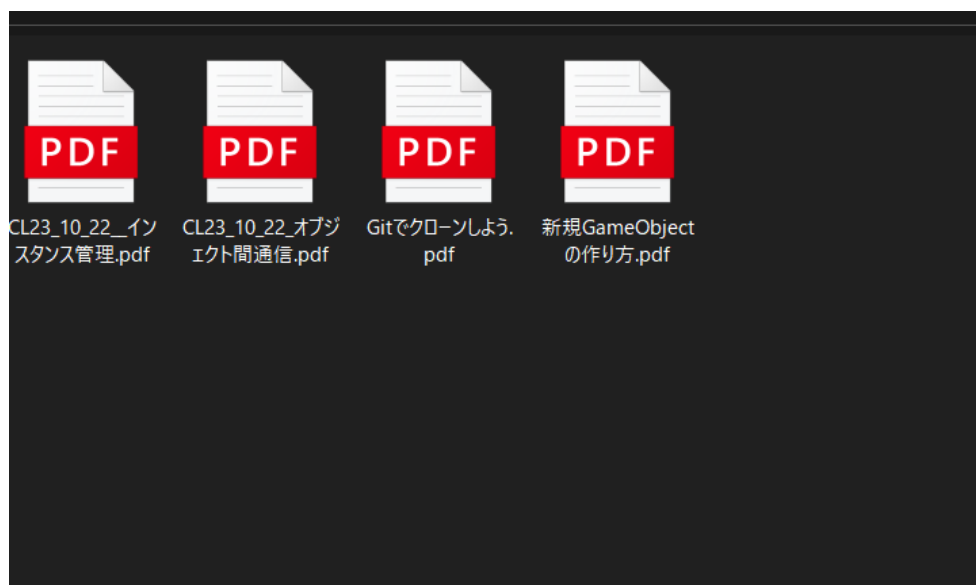
それでは、Git Hubに行ってみましょう

再読み込みすると、、、



リポジトリが作成されています

あれ、でも、こいつらは？



## 本題に入ります

新しく追加したいファイルはさっきのコピペではAddされません  
なので、自分でやります

gitのデータ管理は

Add → Commit → Pushで行います。（Pullもある）

わからない人はProgateやり直してください。

Git Bash に戻って

"git status" コマンドを打ちます



## git status → git add ファイル名

```
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "CL23_10_22_\343\202\244\343\203\263\343\202\271\343\202\277\343\203\263\343\202\271\347\256\241\347\220\206.pdf"
    "CL23_10_22_\343\202\252\343\203\226\343\202\270\343\202\247\343\202\257\343\203\210\351\226\223\351\200\232\344\277\241.pdf"
    "Git\343\201\247\343\202\257\343\203\255\343\203\274\343\203\263\343\203\227\343\202\210\343\201\206.pdf"
    "\346\226\260\350\246\217GameObject\343\201\256\344\275\234\343\202\212\346\226\271.pdf"

nothing added to commit but untracked files present (use "git add" to track)
```

### git statusで確認

赤くなってるのが、まだ追加されていないファイル。

### ファイルをまとめてAddしよう

本来なら、

```
"git add
CL23_10_22_\343\202\244\343\203\263\343\202\271\343\202\277\343\203\263\343\202\271\347\256\241\347\220\206.pdf"
```

と打てば

```
CL23_10_22_\343\202\244\343\203\263\343\202\271\343\202\277\343\203\263\343\202\271\347\256\241\347\220\206.pdf
```

はAddされます。

けど、ファイル名を日本語にしてるせいでめんどくさくなってます。

じゃあ、pdfファイルをまとめてAddしてしまおう

ということで、"git add \*.pdf"というコマンドを打ち込みます

```
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git add *.pdf

yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "CL23_10_22_\343\202\244\343\203\263\343\202\271\343\202\277\343\203\263\343\202\271\347\256\241\347\220\206.pdf"
    new file:   "CL23_10_22_\343\202\252\343\203\226\343\202\270\343\202\247\343\202\257\343\203\210\351\226\223\351\200\232\344\277\241.pdf"
    new file:   "Git\343\201\247\343\202\257\343\203\255\343\203\274\343\203\263\343\203\227\343\202\210\343\201\206.pdf"
    new file:   "\346\226\260\350\246\217GameObject\343\201\256\344\275\234\343\202\212\346\226\271.pdf"
```

これでPDFファイルは全部Addできました。

他にも、

## git add のコマンド紹介

git add .           現在のディレクトリ内のファイルをすべて追加

git add -A           git で管理していないファイルも全部追加

git add -u           変更した部分だけ追加

などなど。いろんなコマンドがあります

調べてみてね

続いてコミットしていきます

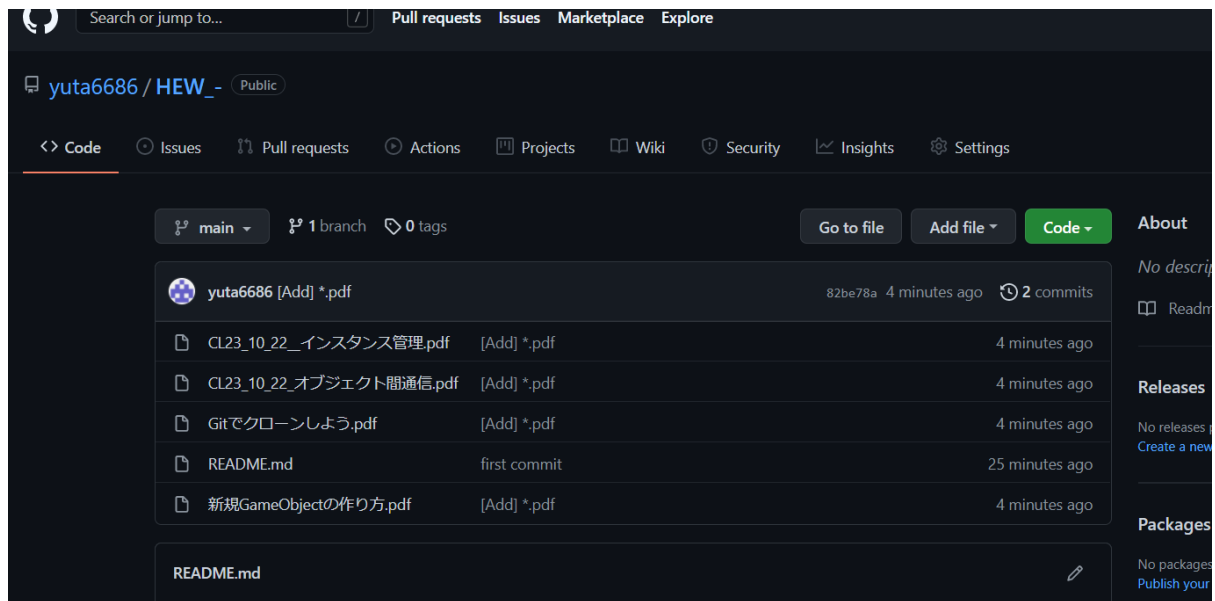
## git commit -m "メッセージ"

```
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル (main)
$ git commit -m "[Add] *.pdf"
[main 82be78a] [Add] *.pdf
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "CL23_10_22_\343\202\244\343\203\263\343\202\271\343\202\277\343\203\263\343\202\271\347\256\241\347\220\206.pdf"
create mode 100644 "CL23_10_22_\343\202\252\343\203\226\343\202\270\343\202\247\343\202\257\343\203\210\351\226\223\351\200\232\344\277\241.pdf"
create mode 100644 "Git\343\201\247\343\202\257\343\203\255\343\203\274\343\203\263\343\201\227\343\202\210\343\201\206.pdf"
create mode 100644 "\346\226\260\350\246\217GameObject\343\201\256\344\275\234\343\202\212\346\226\271.pdf"
```

## git push

```
yanagisaya.yuta@ths00680 MINGW64 ~/Pictures/HEW/HEW_共有ファイル
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.83 MiB | 6.53 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/yuta6686/HEW_-.git
   dee36fa..82be78a  main -> main
```

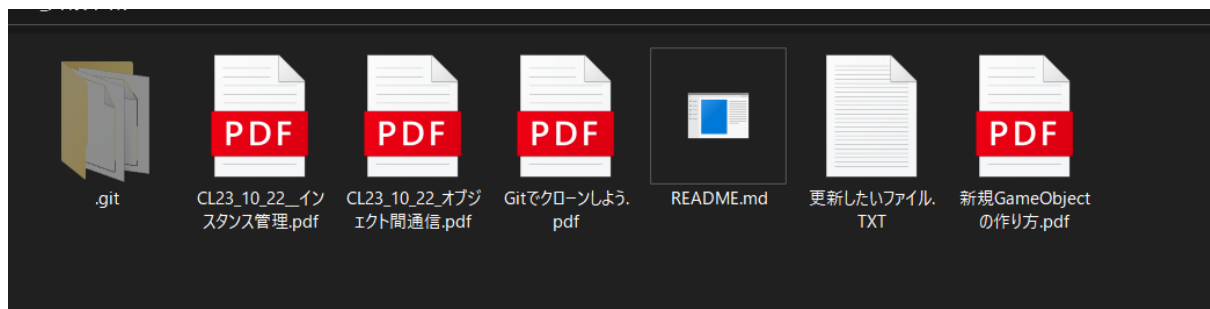
## Githubへ



しっかり作られていますね。

あとは、このサイトのURLを共有すれば完了です。

## 更新したいときは？



新しく”更新したいファイル.TXT”というファイルが作成されたとしましょう  
基本的なやりかたは同じで

1. ここでGit Bash を開く
2. `git add 更新したいファイル.TXT`
3. `git commit -m "なんかメッセージ"`
4. `git push`

これだけでできます。

## Gitで管理できるもの

基本どんなファイルでもいけます。

.xml.pdf.txt,.....blend.mb.....cpp.h.py.html.css.mp4.

容量に限界があることです

