

Git について

1. Git とはなにか？

共同開発(複数人でプログラムなどをかく等)の時に、便利となる“バージョン管理システム”です。

チームで共有して作業しているファイルでは、編集者の名前を入れておくこともあります。しかし、**どのような変更を行ったか**簡単にはわかりません。

また、チームで共有しているファイルの場合では、**二人で同時に編集してしまった**ために、先に編集した人の変更内容が消えてしまったという経験はないでしょうか。

↑こういったことを簡単に解決できるのが、Git です。

これから広報班でよく使うので、よく覚えて、わからないことは聞くなり調べるなりしましょう。

Git では、ファイルの状態を好きなときに更新履歴として保存しておくことができます。そのため、一度編集したファイルを過去の状態に戻したり、編集箇所の差分を表示したりすることができます。

また、古いファイルを元に編集したファイルで、他人の編集した最新ファイルを上書きしようとする、サーバにアップロードした時に警告が出ます。そのため、知らず知らずのうちに他人の編集内容を上書きしてしまうといった失敗は起こりません。

2. Git のよく使う用語

- ・ローカルリポジトリとリモートリポジトリ

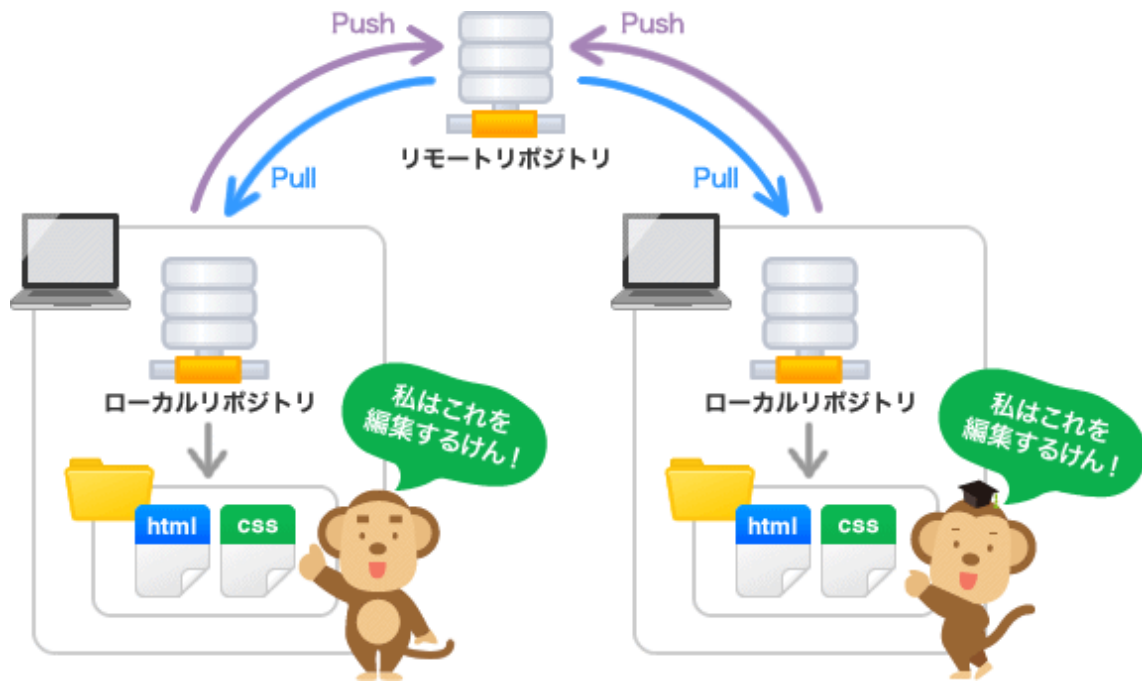
まず、Git のリポジトリは、リモートリポジトリとローカルリポジトリの2種類に分けられます。

リモートリポジトリ:専用のサーバに配置して複数人で共有するためのリポジトリです。

ローカルリポジトリ:ユーザー一人ひとりが利用するために、自分の手元のマシン上に配置するリポジトリです。

リポジトリをリモートとローカルの2種類に分けることで、普段の作業はローカルリポジトリを使って全て手元のマシン上で行うことができます。

自分のローカルリポジトリで作業した内容を公開したい時は、リモートリポジトリにアップロードして公開します。また、リモートリポジトリを通してほかの人の作業内容を取得することもできます。



イメージ図

- コミット

ファイルやディレクトリの追加・変更を、リポジトリに記録するには**コミット**という操作を行います。

コミットを実行すると、リポジトリの内では、前回コミットした時の状態から現在の状態までの差分を記録したコミット(またはリビジョン)と呼ばれるものが作成されます。

コミットの実行時には**コミットメッセージ**の入力を求められます。コミットメッセージは必須となっているため、空のままで実行するとコミットが失敗します。

- プッシュとプル

プッシュ:自分のローカルリポジトリからリモートリポジトリにデータを送ること

プル:リモートリポジトリから自分のローカルリポジトリにデータをもらうこと

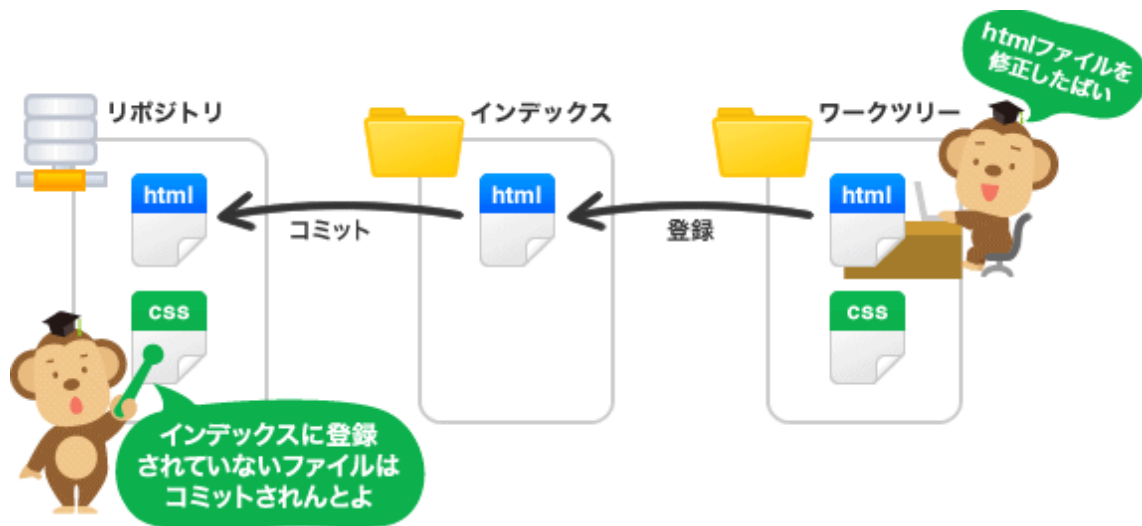
プルは特に何もしないでできますが、**プッシュは2段階の過程のあとにしかできないので、注意です。**(後ほど詳しく説明します)

- ワークツリーとインデックス

みなさんが実際に作業をしているディレクトリのことをワークツリーと呼びます。

そして、Git ではリポジトリとワークツリーの間にはインデックスというものが存在しています。インデックスとは、リポジトリにコミットする準備をするための場所のことです。

このワークツリーからインデックスへのコピーは add というコマンドで実施します。



これらの、add→commit→push の過程を合わせて、アップロードという。

3. 具体的なコマンドと共有までの過程など

1) まず、はじめにファイルができれば

```
$ git add .
```

をする。これにより、ワークツリーからインデックスにデータがすべてコピーされる。
なお、特定のファイルのみをインデックスにコピーするときは

```
$ git add ファイル名
```

でも可能である。

基本的には、前者の方で問題はないです。

```
問題 出力 デバッグ コンソール ターミナル

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program
$
```

↑ この状態から

```
問題 出力 デバッグ コンソール ターミナル

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program
$ cd "C:\Users\Ryuichi\Desktop\Program\test"

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git add .

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$
```

↑ add 後

2) 次に、

`$ git commit -m"コミット名"`

をする。これにより、インデックスからリポジトリにデータが送られる。

このコミット名は日本語でも可能だが、スラッシュは使わない方が良い。また、できる限り具体的なものにします。

(Xファイル更新 → ○理工白門祭実行委員会 html 更新)

```
問題 出力 デバッグ コンソール ターミナル

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program
$ cd "C:\Users\Ryuichi\Desktop\Program\test"

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git add .

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git commit -m"test01の更新"
[main c9f2391] test01の更新
1 file changed, 1 insertion(+), 1 deletion(-)

Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$
```

↑ コミット後 今回は"test01"という名のファイルを更新した

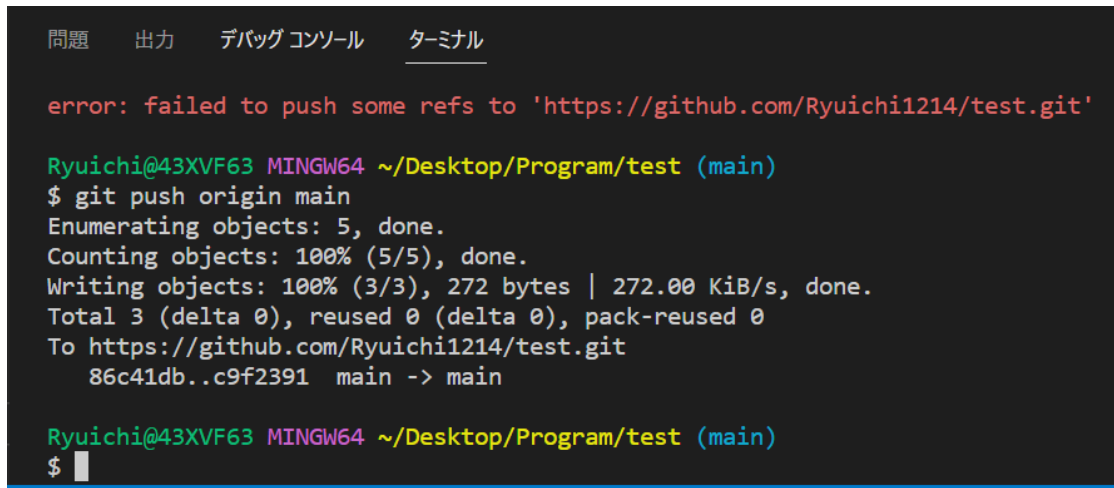
3) そして、

```
$ git push origin main
```

または

```
$ git push origin ブランチ名
```

これにより、リモートリポジトリに共有が完了した。

A terminal window with a dark background and light-colored text. At the top, there are tabs labeled '問題', '出力', 'デバッグ', 'コンソール', and 'ターミナル', with 'ターミナル' being the active tab. The terminal shows an error message in red: 'error: failed to push some refs to 'https://github.com/Ryuichi1214/test.git''. Below this, the user's prompt is 'Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)'. They enter '\$ git push origin main'. The terminal shows the progress of the push: 'Enumerating objects: 5, done.', 'Counting objects: 100% (5/5), done.', 'Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.', 'Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', and 'To https://github.com/Ryuichi1214/test.git 86c41db..c9f2391 main -> main'. The prompt returns to 'Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main) \$'.

↑プッシュ後 このメッセージが出れば完了

4. エラーが出たときは？

まずは、エラーの原因を探る。基本英語で表示されるので、落ち着いて読めば案外読めたりする。わからなければ、そのままコピーして Google 先生に頼ろう。意外と答えが見つかる。

またよく起こるのが、競合という現象。これは、他の人がファイルやデータを編集し、プッシュしたあとに、自分が更にプッシュしようとする時に出るエラーです。

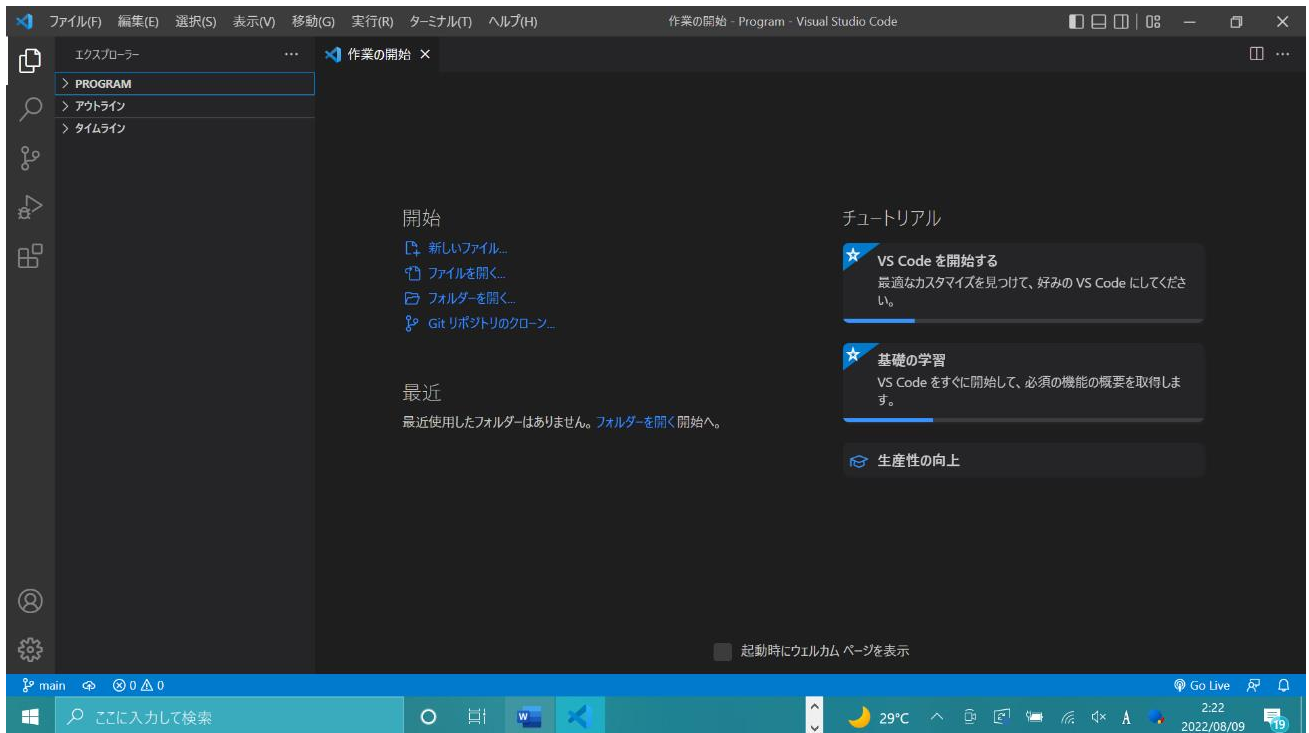
解決方法は意外と簡単で、add したあとに、一度プルし、再度コミット→プッシュすれば良い。多分この後の、実践編でも何回か生じると思います。

5. いざ、実践！

ここまでの実践でわかったことを復習も兼ねて実践してみよう。

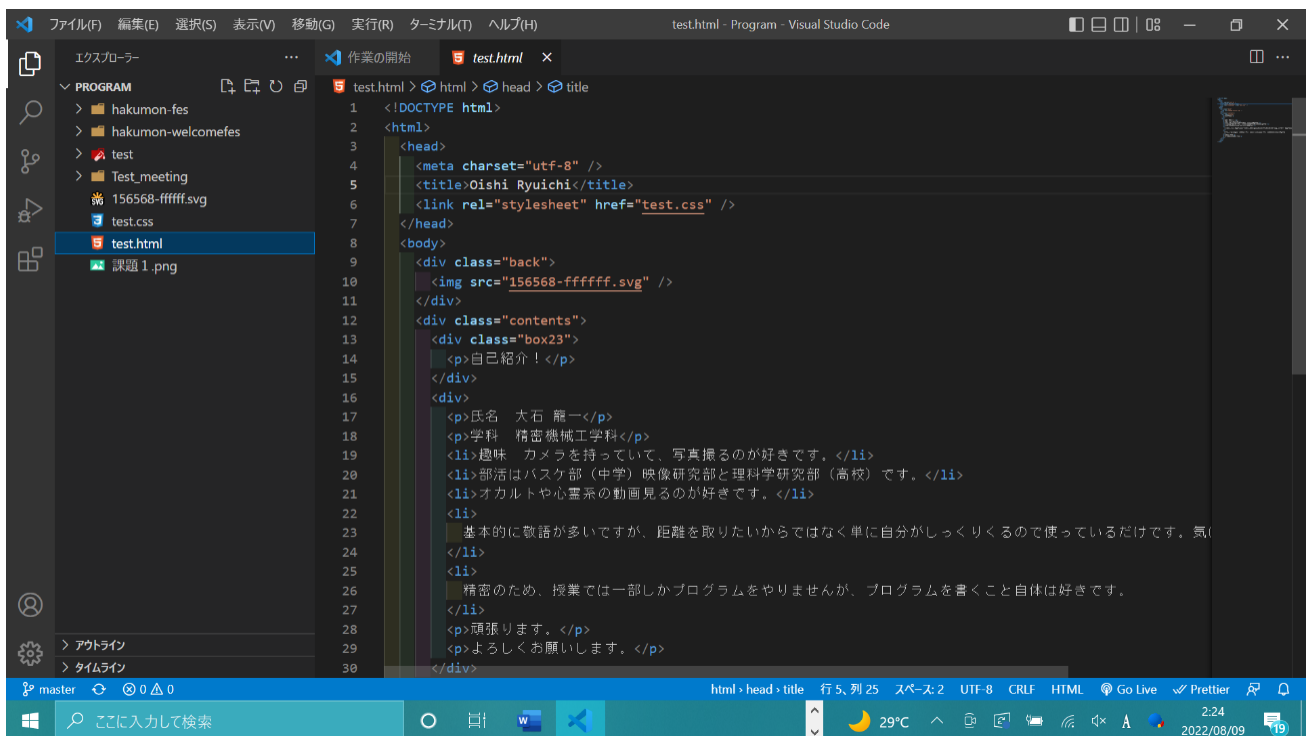
今回は、みんなが以前作ってくれた自己紹介の html が残っていればそれを、ない人は適当な html を Git で共有してみよう。

1) まず、vscode を開く



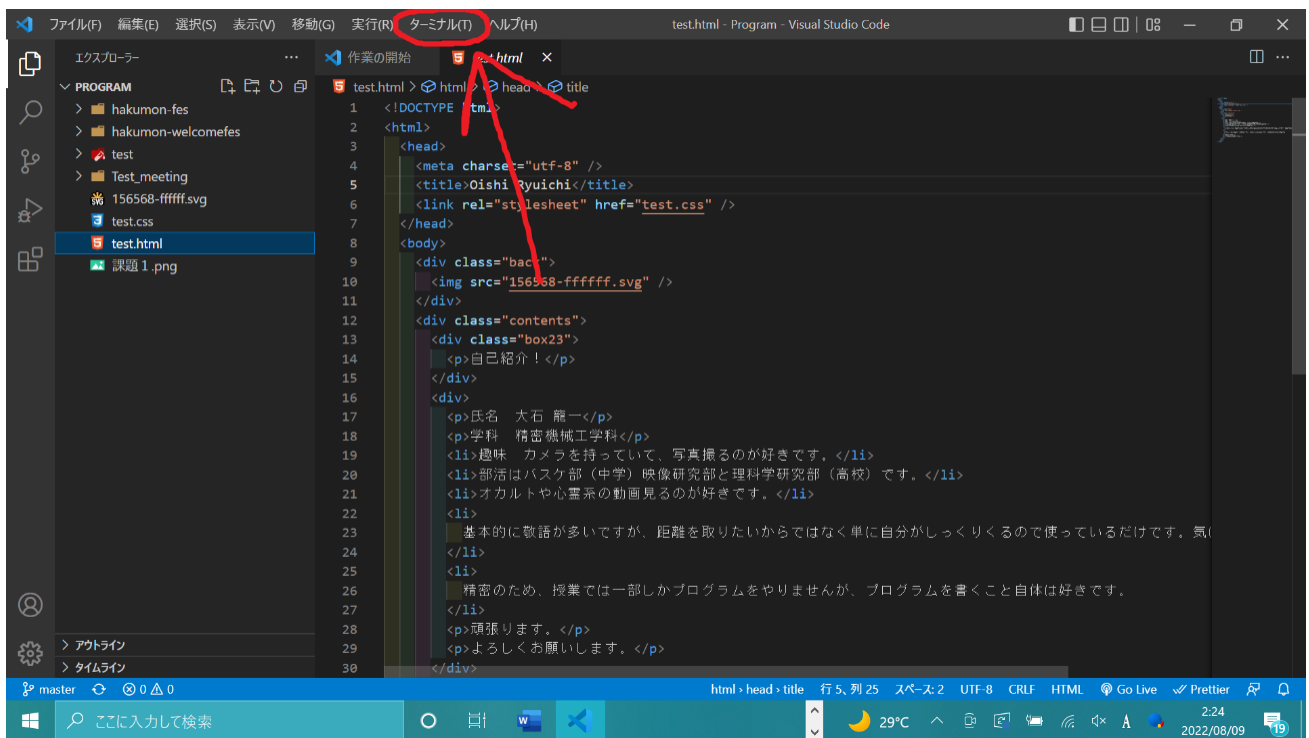
大体の人はこんな感じの画面かな？

2) 自分の作った html を開く

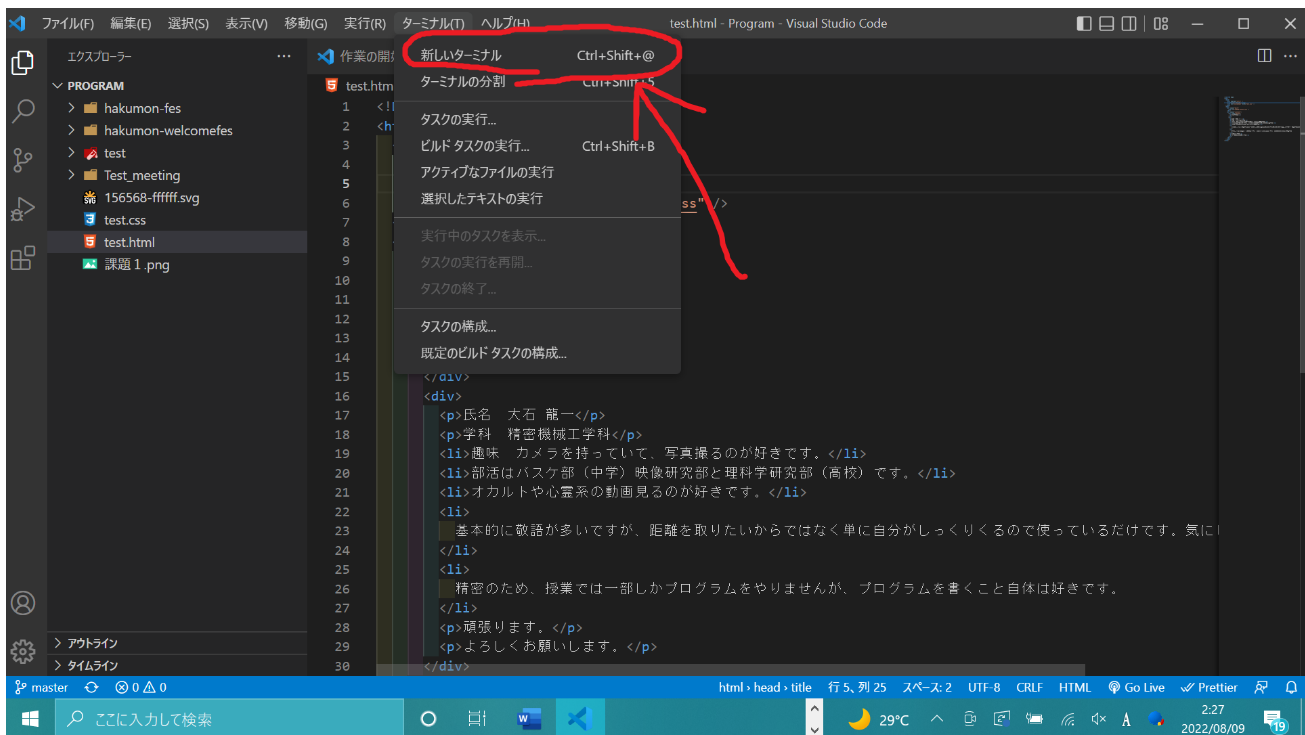


多分こんな感じになると思います

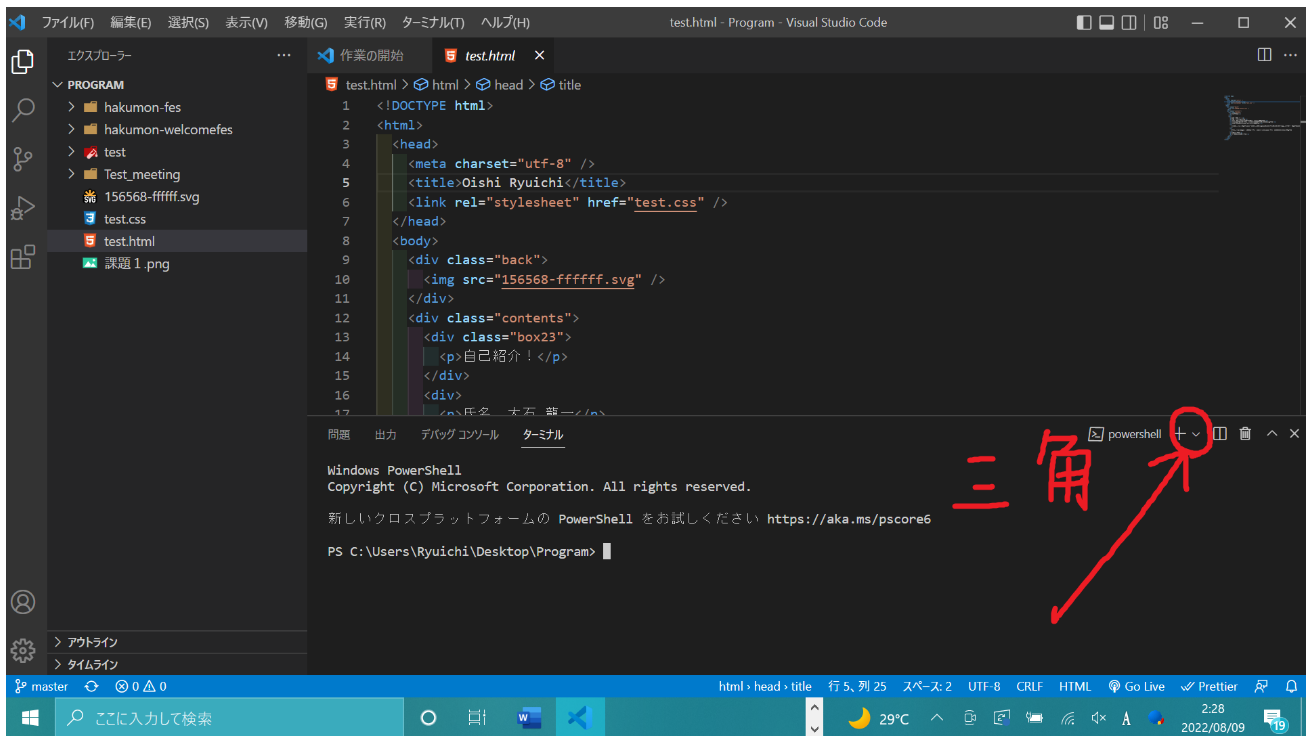
3) 画面中央の上部あたりのターミナル→Git bush を選ぶ



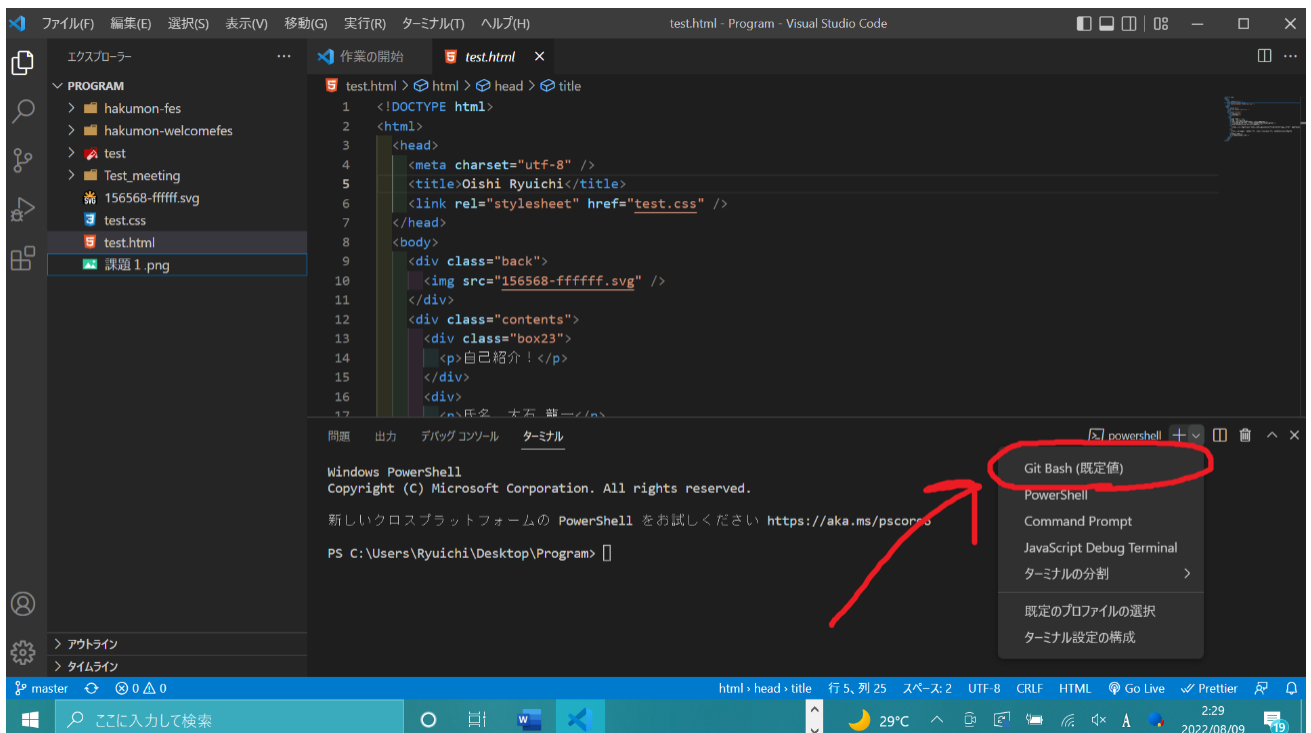
ターミナルから…



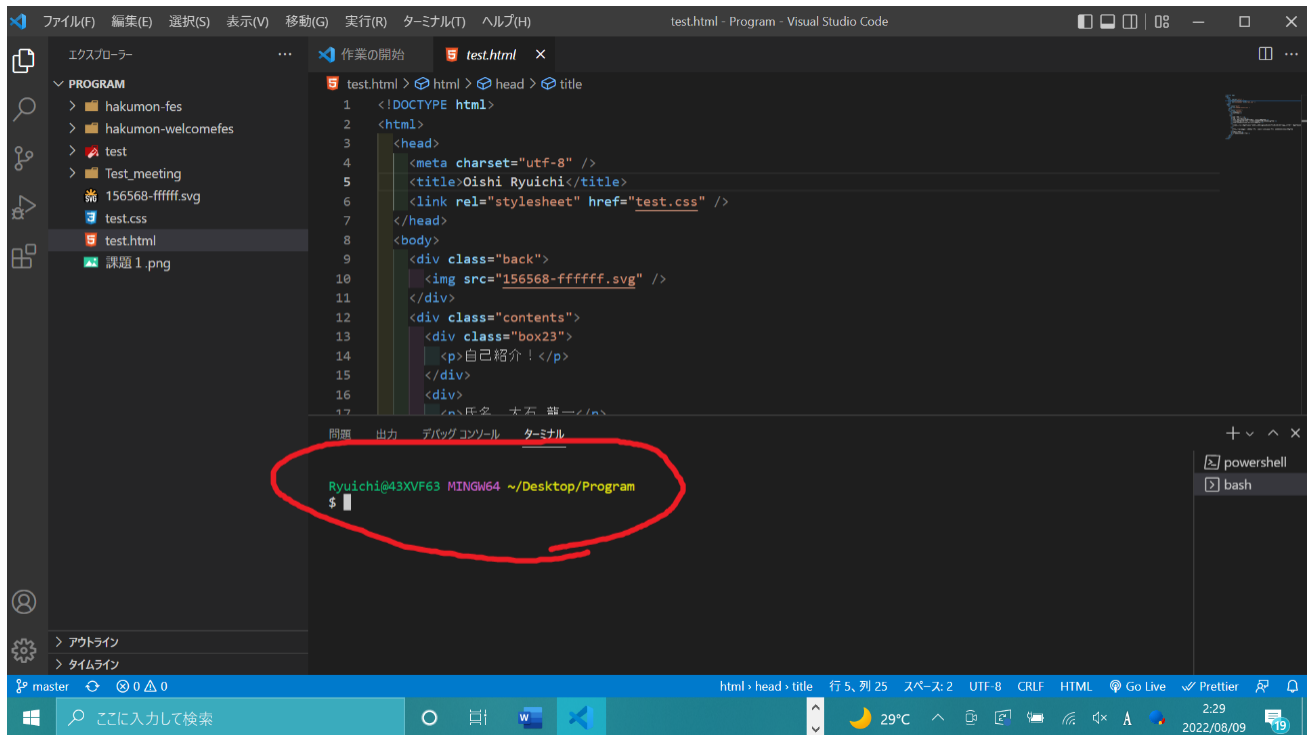
新しいターミナルを選んで



多分最初は PowerShell が起動するので、三角をおして



Git bash を選んで



この表示が出たら完成

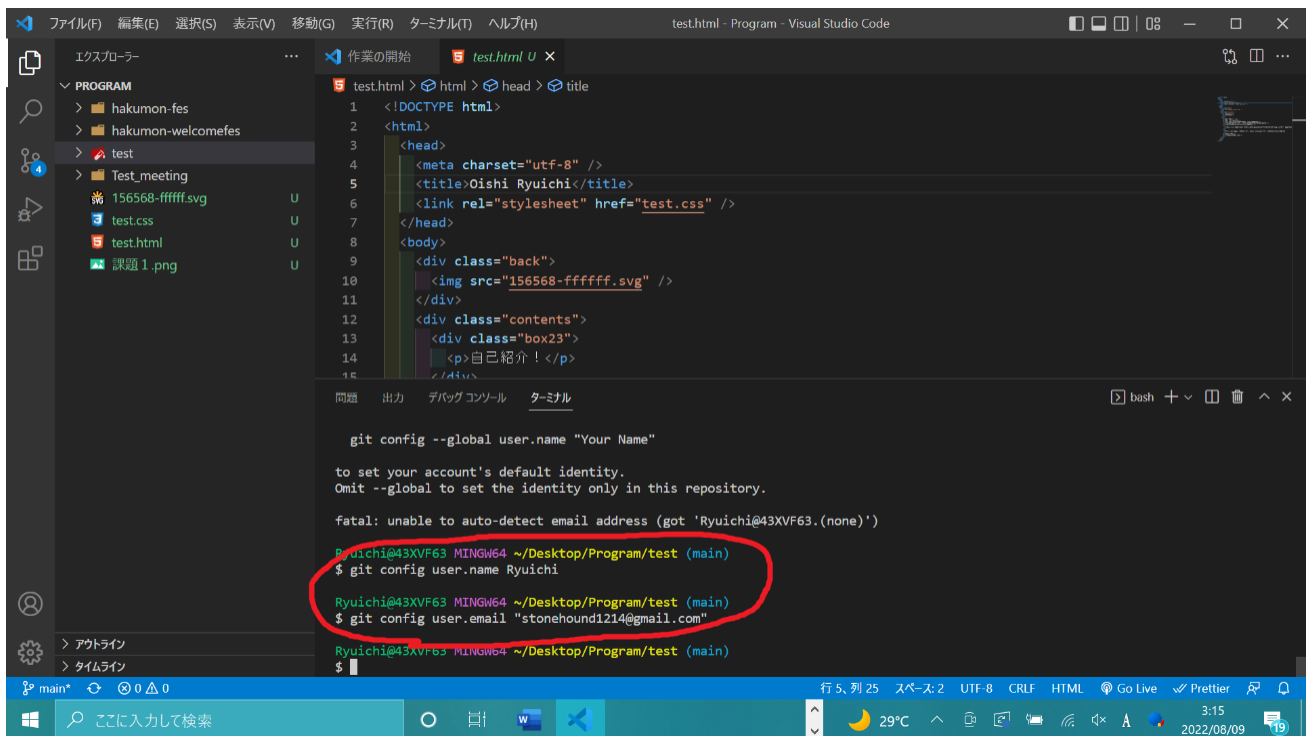
4) Git にメールアドレスと名前を教える

\$ git config user.name ユーザー名

\$ git config user.email メールアドレス

を使って、まずは自分が誰なのかを明かす。

このときの、ユーザー名とメールアドレスは、Git Hub の登録時と同じ物を使うこと。



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with files like 156568-fffff.svg, test.css, test.html, and 課題1.png. The main editor shows a file named test.html with HTML code. The terminal window at the bottom shows the following commands and output:

```
git config --global user.name "Your Name"
to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'Ryuichi@43XVF63.(none)')
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git config user.name Ryuichi
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git config user.email "stonehound1214@gmail.com"
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$
```

5) つぎに、Git のリモートリポジトリの設定をする。

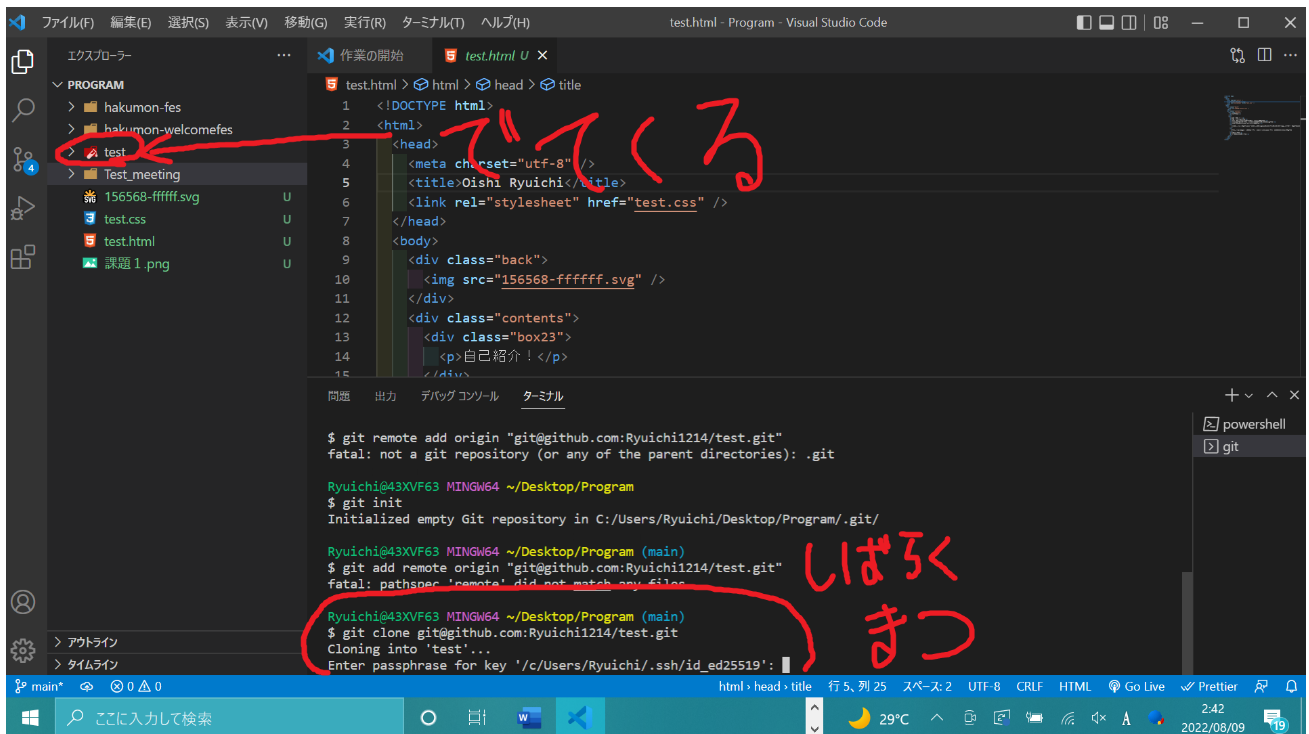
\$ git remote add origin URL

URL と書いたところに、[git@github.com:Ryuichi1214/test.git](https://github.com/Ryuichi1214/test.git) を貼り付ける

6) 更に、

\$ git clone URL

で、自分のローカルリポジトリにデータをコピーする。



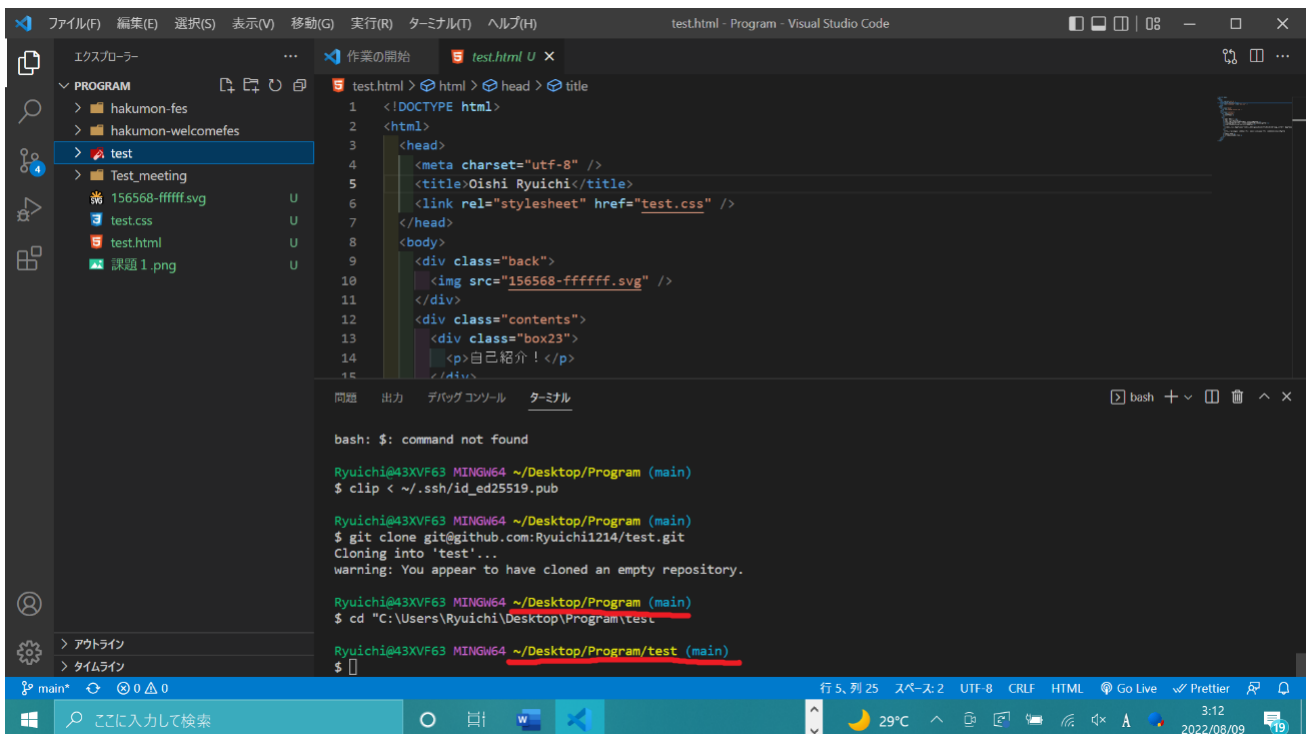
しばらく待つと、test という名のファイルが追加される

7) test のファイルに自分の html をアップロードする。

まず、

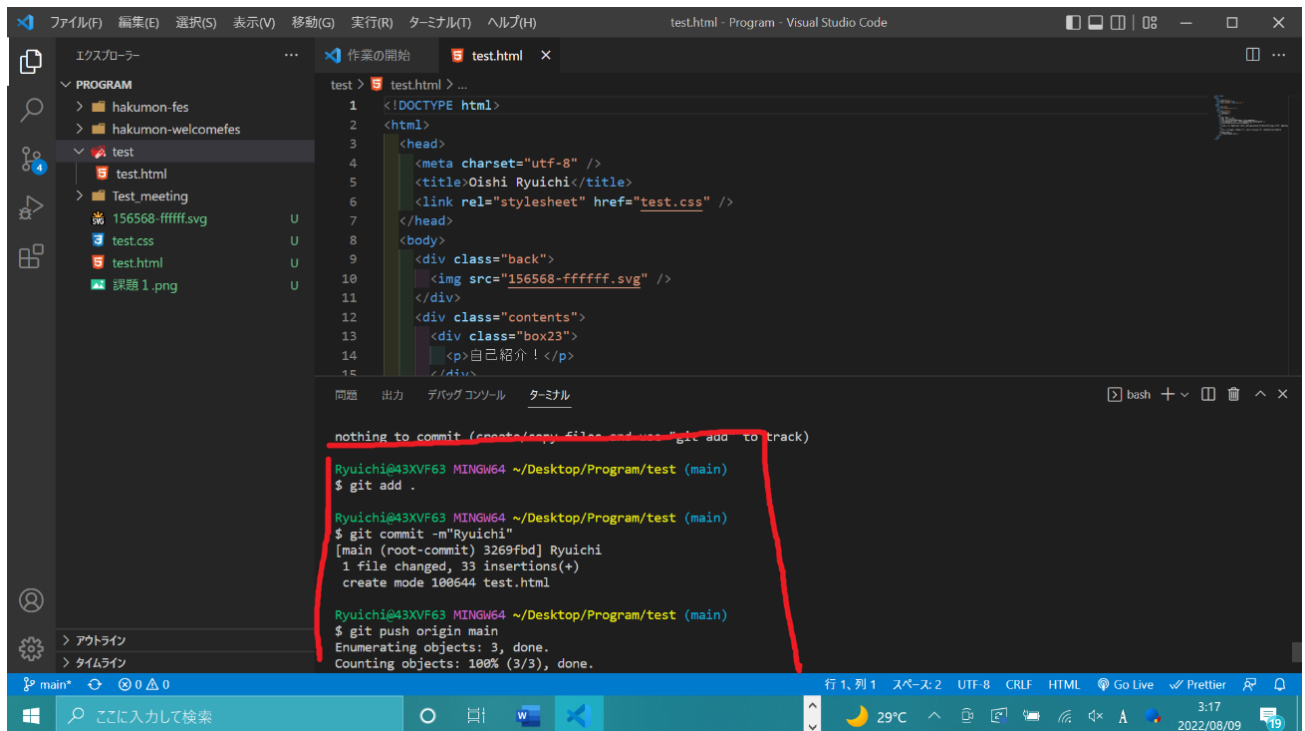
\$ cd”パス“

を打ち込んで、場所を移動する。



こうなったら ok

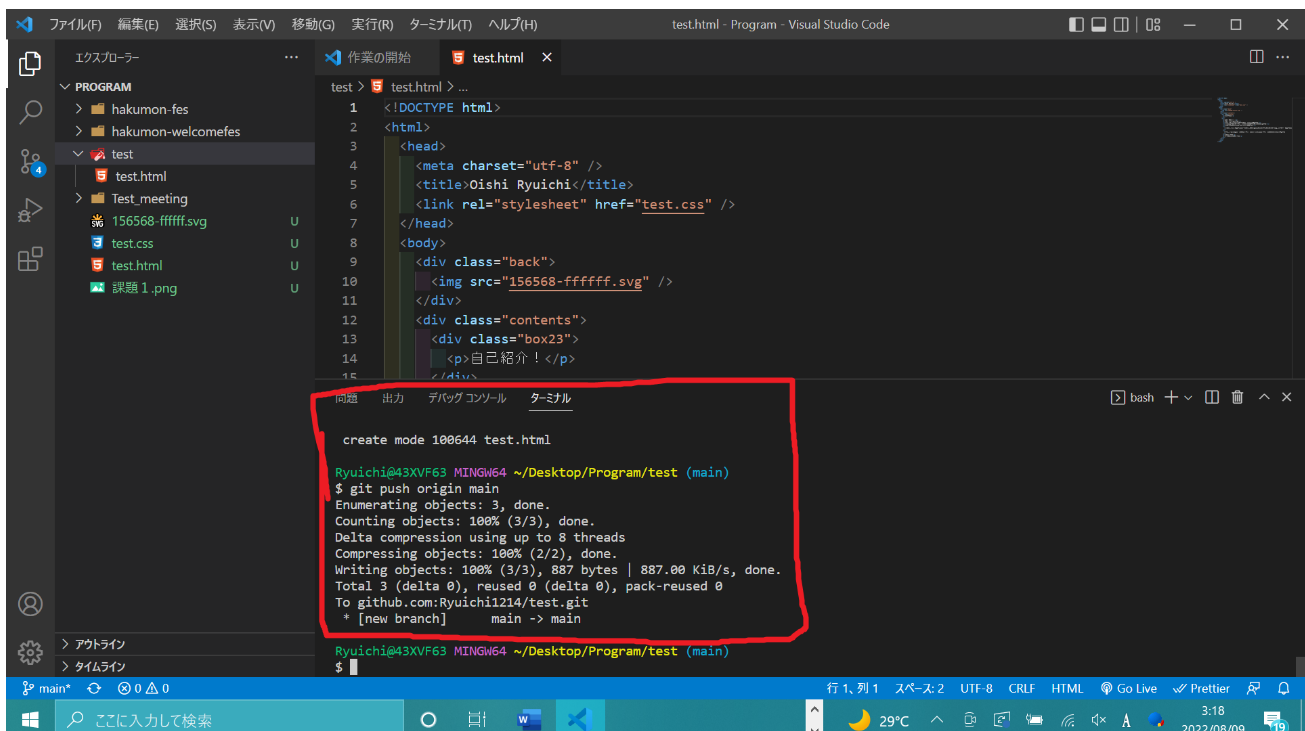
ここからは、さっき教えた手順でアップロードする。



```
test > test.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>Oishi Ryuichi</title>
6 <link rel="stylesheet" href="test.css" />
7 </head>
8 <body>
9 <div class="back">
10 
11 </div>
12 <div class="contents">
13 <div class="box23">
14 <p>自己紹介！</p>
15 </div>
16 </div>
17 </body>
18 </html>
```

```
nothing to commit (create/copy files and use "git add" to track)
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git add .
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git commit -m "Ryuichi"
[main (root-commit) 3269fbd] Ryuichi
1 file changed, 33 insertions(+)
create mode 100644 test.html
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
```

Add して commit して…



```
test > test.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>Oishi Ryuichi</title>
6 <link rel="stylesheet" href="test.css" />
7 </head>
8 <body>
9 <div class="back">
10 
11 </div>
12 <div class="contents">
13 <div class="box23">
14 <p>自己紹介！</p>
15 </div>
16 </div>
17 </body>
18 </html>
```

```
create mode 100644 test.html
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 887 bytes | 887.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Ryuichi1214/test.git
 * [new branch]      main -> main
Ryuichi@43XVF63 MINGW64 ~/Desktop/Program/test (main)
$
```

Push して完了

8) エラーが出たら、一度 pull してから、commit→push

分かんなかったら聞いてね。

9) 自分のアップロード終わったら他の人のプログラムを覗いてみよう。