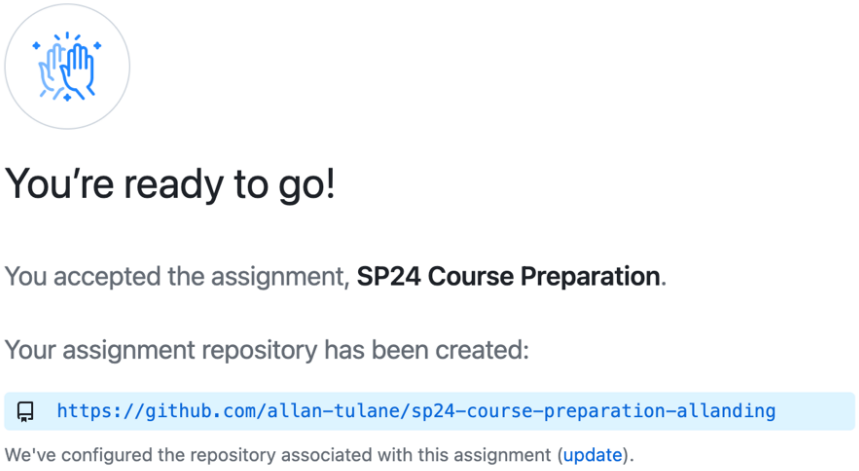
Course Preparation Introduction

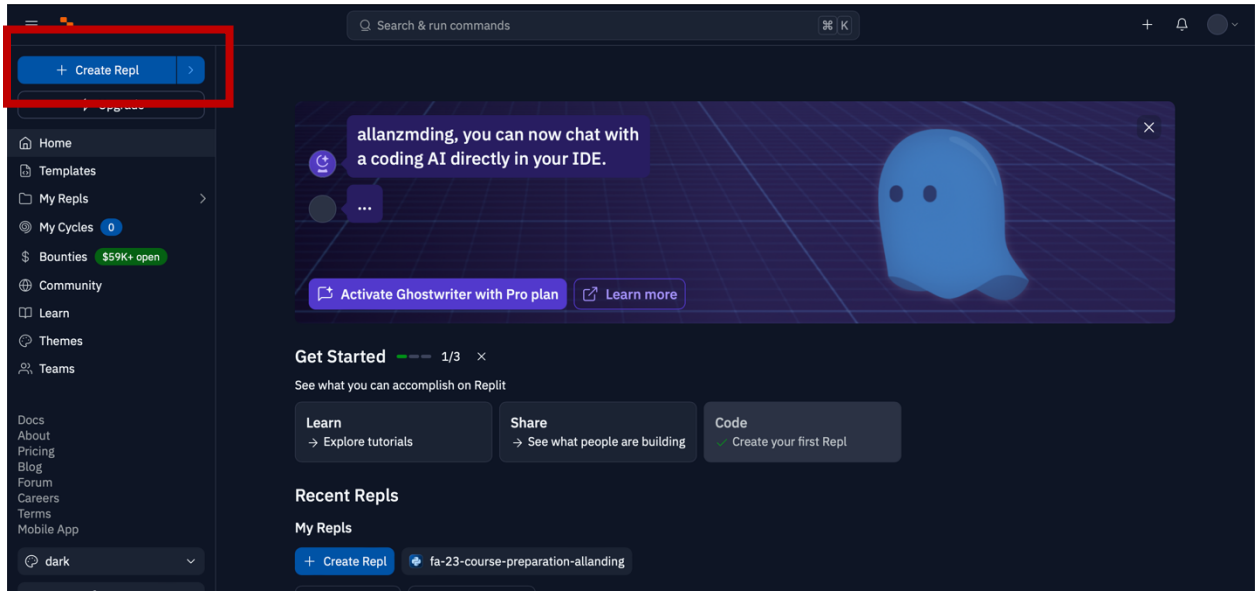
First, click the Google classroom link https://classroom.github.com/a/uz7DadW6 and accept the assignment, which will generate a personal GitHub link for each of your assignments https://github.com/allan-tulane/sp24-course-preparation-[**YOUR-GITHUB-ID**]

For example, this is my one https://github.com/allan-tulane/sp24-course-preparation-allanding, where **allanding** is my GitHub ID.

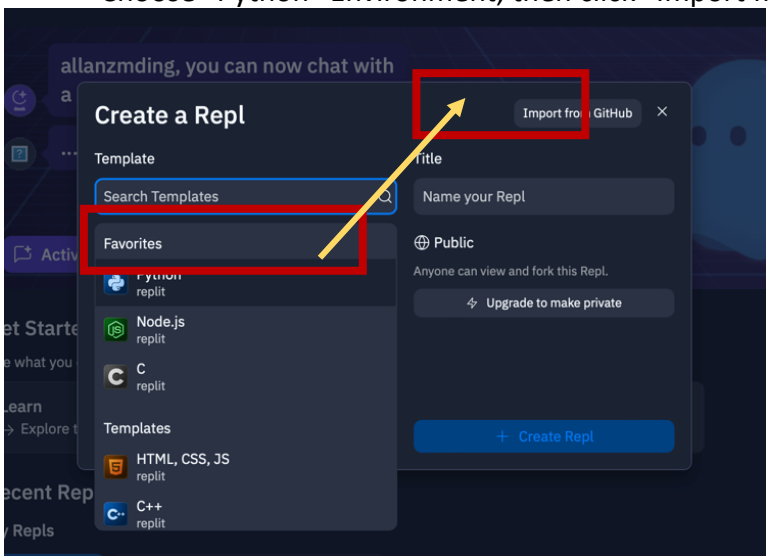


Second, import GitHub to Replit https://replit.com/~ in Python Environment, and complete your assignment in Replit

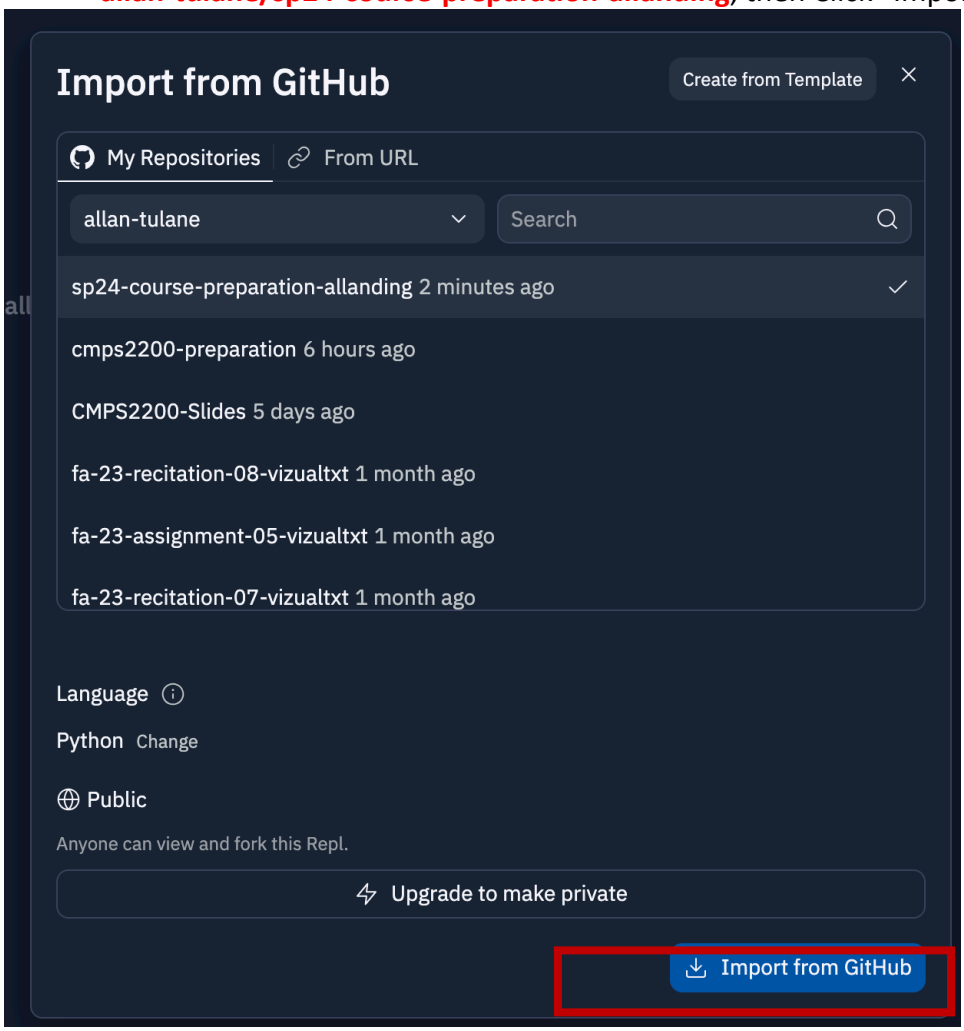
- Click "Create Repl"



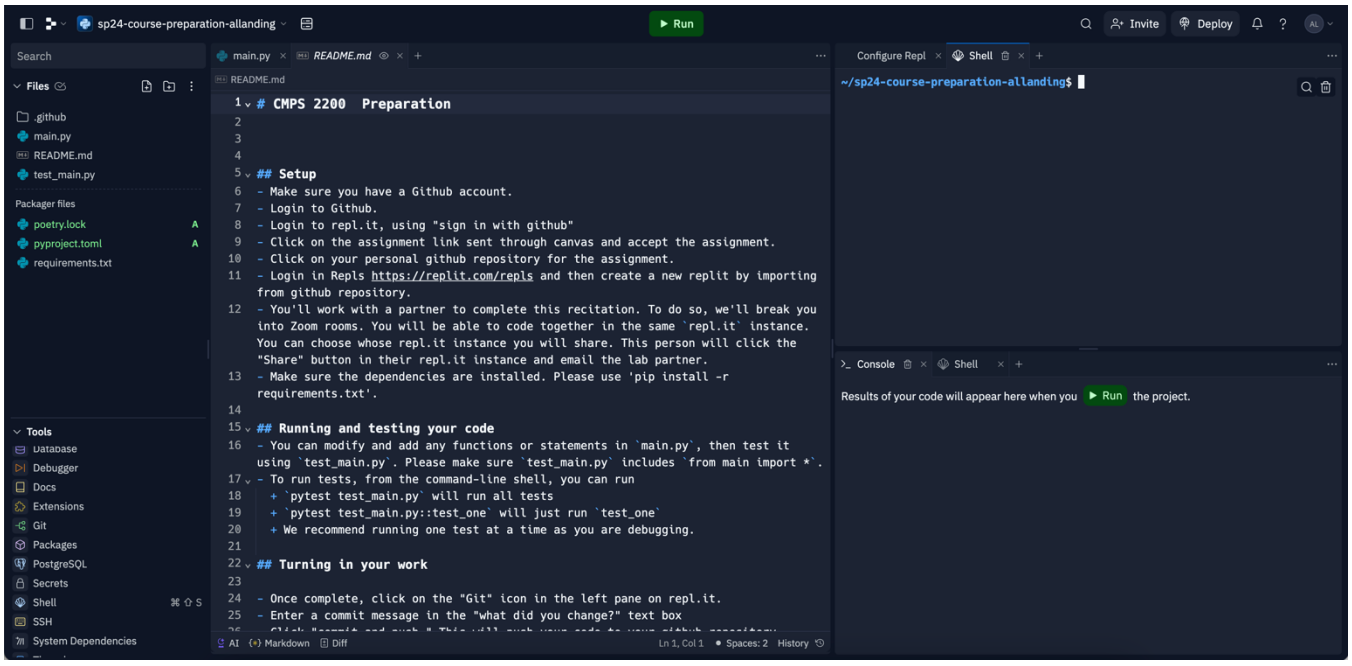
- Choose “Python” Environment, then click “Import from GitHub”



- Input GitHub URL by removing the head part and only using **allan-tulane/sp24-course-preparation-allanding**, then Click “Import from GitHub”



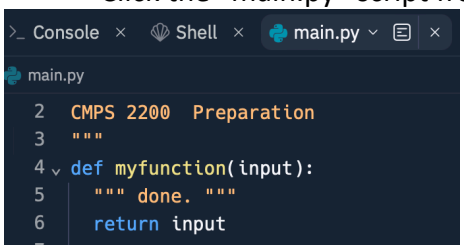
- You will see the Repl Layout as below. Maybe not exactly the same.
- Find the “Shell” under Tools, which will be our terminal to run python commands.



You can list all files by typing “ls”, which is the UNIX commands

Entering Python Environment by typing “python” and exit with “exit()”

- Click the “main.py” script from **Files**, and you can see a predefined python function



If you want to run this function, you can add one statement after **myfunction** as print(myfunction('Hello World')), then go to Shell and type “python main.py”

- Click “test_main.py” from **Files,** where we will use “pytest” command to test if the function output matches what we want. The statement is “assert myfunction('Hello') == 'Hello'”, where “assert” is the key word in python. **There are three test cases in our given script.**

```
Console × Shell × main.py × test_main.py
test_main.py
1  from main import *
2
3
4  def test_myfunction():
5      """ done. """
6      assert myfunction('Hello') == 'Hello'
7      assert myfunction(1) == 1
8      assert myfunction(6) == -1
```

The first two are True, but the third one is False. Before you run “pytest”, please install it by going to Shell and typing “pip install pytest” or “pip install -r requirements.txt”, where **requirements.txt** includes all python dependencies.

```
Configure Repl × Shell × +
~/sp24-course-preparation-allanding$ pip install pytest
```

```
Configure Repl × Shell × +
~/sp24-course-preparation-allanding$ pip install -r requirements.txt
```

To run tests, from the command-line shell, you can run

- pytest test_main.py will run all tests
- pytest test_main.py::test_one will just run test_one

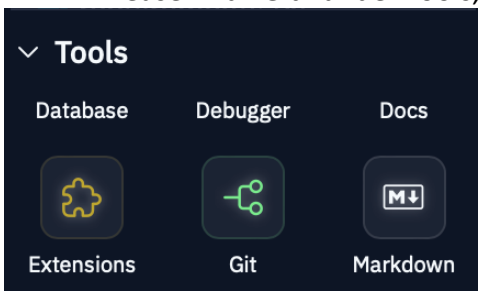
We can see the last case does not pass.

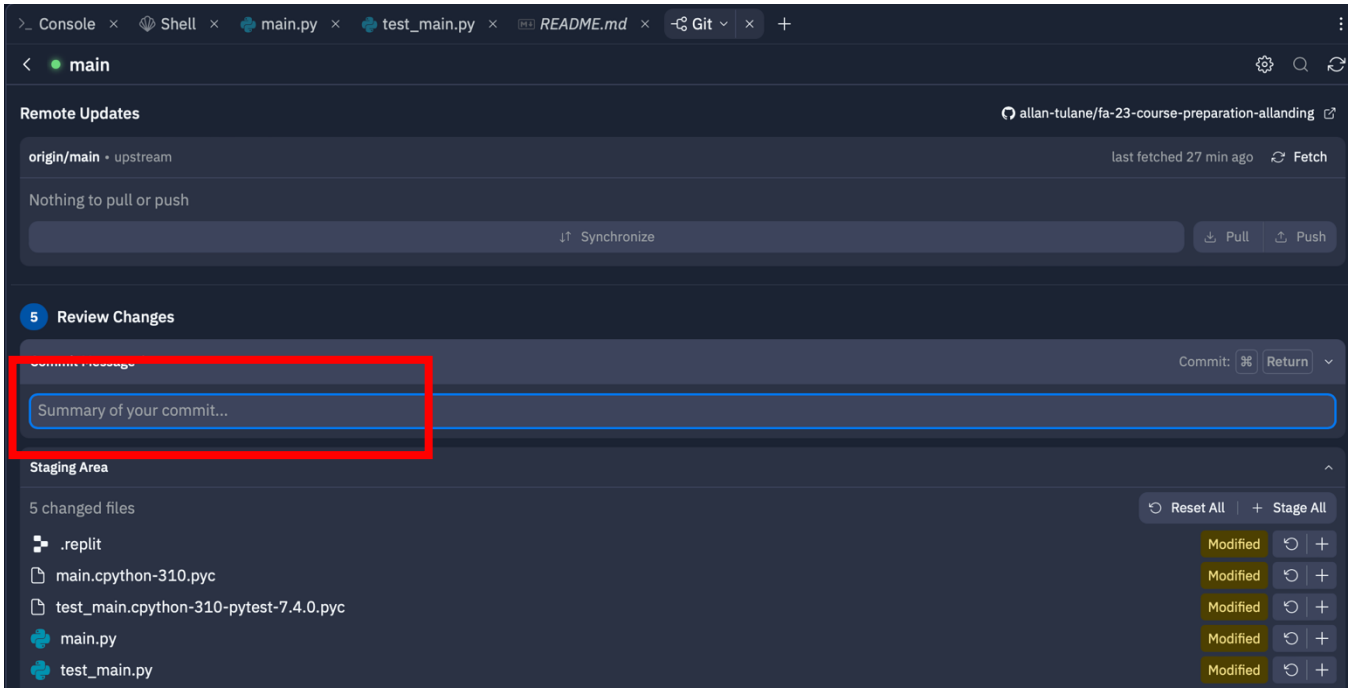
```
~/sp24-course-preparation-allanding$ pytest test_main.py
========================== test session starts ==========================
platform linux -- Python 3.10.11, pytest-7.4.4, pluggy-1.3.0
rootdir: /home/runner/sp24-course-preparation-allanding
collected 1 item

test_main.py F                                                    [100%]

================================ FAILURES ================================
_____________________________ test_myfunction _____________________________

    def test_myfunction():
        """ done. """
        assert myfunction('Hello') == 'Hello'
        assert myfunction(1) == 1
>       assert myfunction(6) == -1
E    assert 6 == -1
E     +  where 6 = myfunction(6)

test_main.py:7: AssertionError
======================== short test summary info =========================
FAILED test_main.py::test_myfunction - assert 6 == -1
========================== 1 failed in 0.12s ==========================
```

If we change the last test case “assert myfunction(6) == 6”

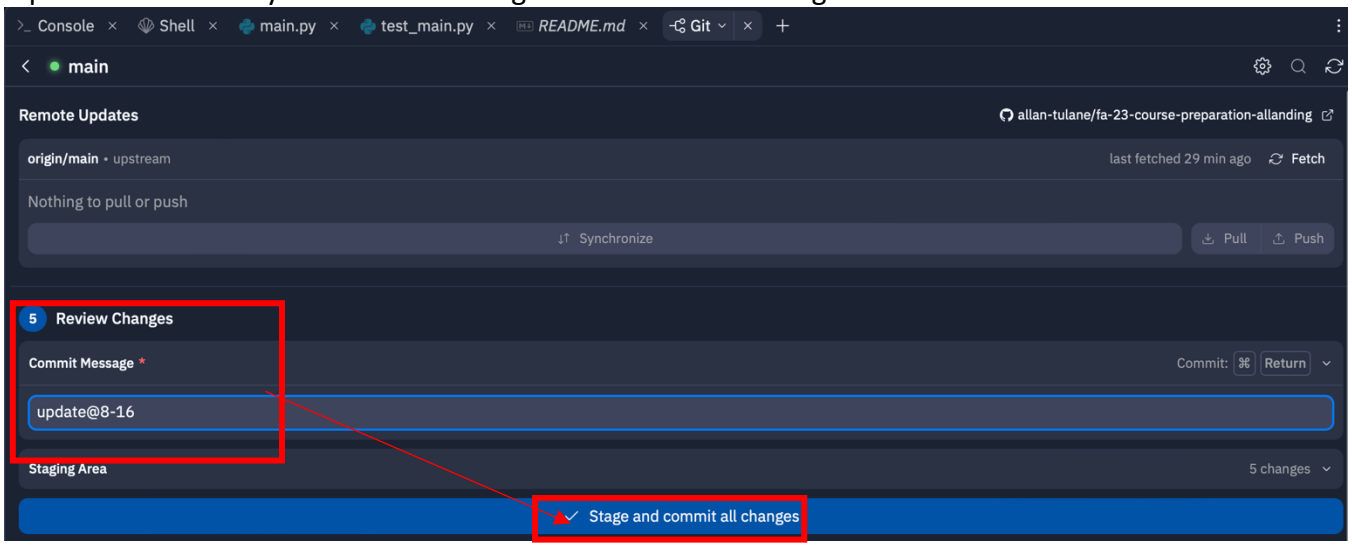
```
~/sp24-course-preparation-allanding$ pytest test_main.py
============================= test session starts ==============================
platform linux -- Python 3.10.11, pytest-7.4.4, pluggy-1.3.0
rootdir: /home/runner/sp24-course-preparation-allanding
collected 1 item

test_main.py .                                                           [100%]

============================== 1 passed in 0.01s ===============================
~/sp24-course-preparation-allanding$
```

- After you implement all tasks and test all cases, you need to commit all updates back to GitHub. Please find “Git” under Tools, and click it.





Input some summary and click the “Stage and commit all changes” butt



- click “push ? commit” button and confirm. **Then You are done. You can check your GitHub URL to see all updates.**