



가상환경 설치하는 방법

[유어클래스 - Archives - DE Docs - PYTEST설치방법 참고]



Pytest-submit: #1.Install

section3 와 section 5 에서는 pytest 를 이용해 테스트가 진행되고, 코드스테이츠 서버와 연동하는 pytest-submit 플러그인을 통해 채점과 과제제출이 진행됩니다.

! 과제제출은 pytest-submit, Pull request 2가지 모두 진행해주셔야합니다.

! `$ pip install`

```
git+https://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@github.com/aib-contents/pytest-submit-plugin
```

복사해주세요!

Step 1 - Install - 자신이 사용할 Conda 환경 생성 및 실행

- ! conda 환경을 이용하실 땐, 실행되어있는 모든 conda 환경을 deactivate 하시고 진행하셔야합니다

```
$ conda create --name hello_conda python=3.8
```

```
~/D/w/2/a/git_flow > ? ... conda create --name hello_conda python=3.8
```

\$ conda activate hello_conda

```
~/D/w/2/a/git_flow > ? ... conda activate hello_conda  
[ ] hello_conda > ~/D/w/2/a/git_flow > ? ...
```

Step 2 - 생성한 **conda** 환경에 **pytest-submit** 설치

```
$ pip install
```

```
git+https://ghp_kEyHMJ6FyAHSLghjHrilb19sc5QHf14  
KxWJW@github.com/aib-contents/pytest-submit-plu  
gin
```



Step 3 - 생성한 **conda** 환경에 **pytest-submit** 설치되었는지 확인

- **!** 버전은 다를 수 있습니다. **pytest-submit** 이 설치되지 않거나, 작동이 정상적으로 진행되지 않는 경우 헬프 데스크를 통해 알려주시기 바랍니다.
- 사용하는 셸에 주의해서 입력해주세요!

유닉스 기반 셸 (Mac, Linux 터미널) 에서 확인방법

```
$ pip list | grep pytest
```

윈도우 체계 (**Git bash**) 에서 확인방법

```
$ pip list | findstr pytest
```

- 스크린샷은 Mac 기반으로 작성되어 있습니다

```
hello_conda ~/D/w/2/a/git_flow pip list | grep pytest
pytest 6.2.4
pytest-json-report 1.4.0
pytest-metadata 1.11.0
pytest-submit 1.3.3
hello_conda ~/D/w/2/a/git_flow
```



Pytest-submit: #2. 과제

제출하기

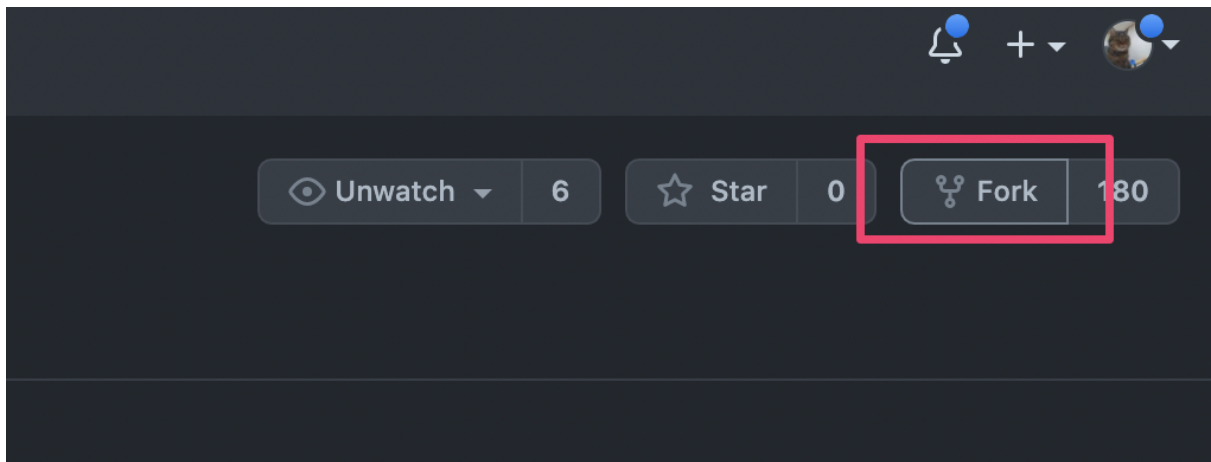
Pytest-submit: #1. Install 카드가 선행되어야 합니다.

Section3 Sprint1 Note1 학습 후 요구되는 사항입니다.

`python -m pytest --submit` 을 통해 과제를 제출할 수 있습니다.

Step 1 - 과제를 개인 레포에 **fork** 해오기

과제 예시: <https://github.com/codestates/ds-pytest-sample>



Step 2 - 과제를 로컬에 **clone** 해오기

! `git clone`시 Terminal에서 인증이 발생하신 분들은 Github access token 페이지를 참고해주세요!

예시 코드: `$ git clone`

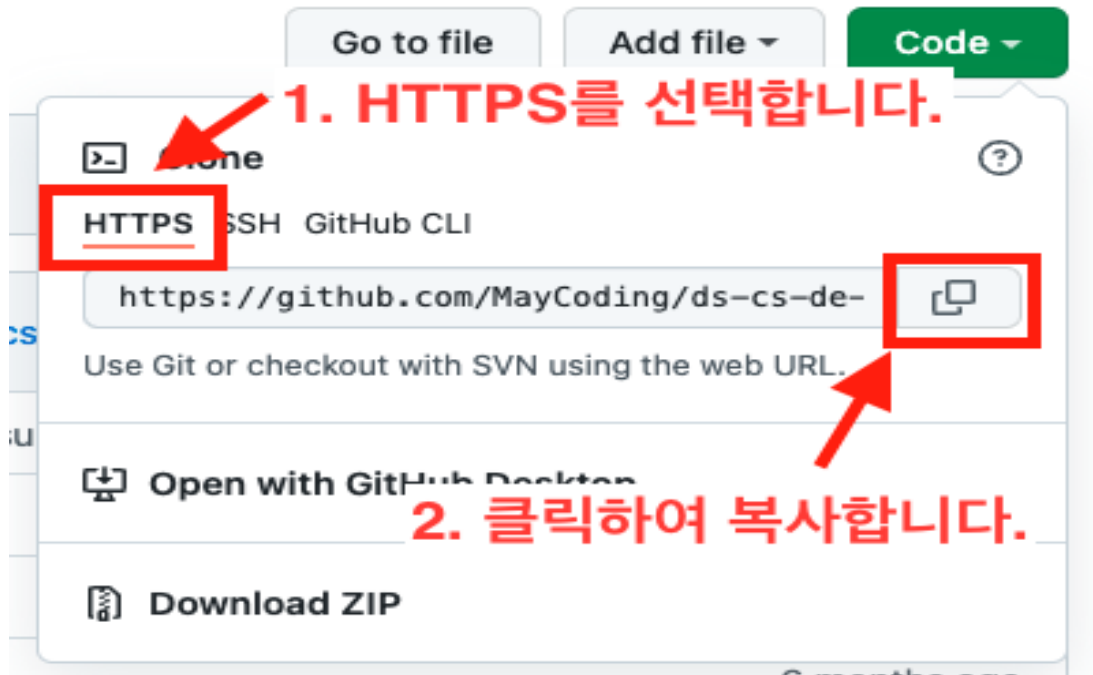
`https://github.com/{유저이름}/ds-pytest-sample`

`$ git clone https://github.com/{유저이름}/{과제명}`



TIP: 레포에서

`https://github.com/{유저이름}/{과제명}` 부분은 복사해 올 수 있습니다.



Step 3 - 과제 폴더로 이동

예시 코드: `$ cd ds-pytest-sample`

`$ cd {과제명}`

Step 4 - 과제에 사용할 파이썬 패키지 설치

💡 TIP: `ls` 명령어를 통해 과제 폴더에 `requirements.txt` 파일이 있는지 확인합니다.

```
$ pip install -r requirements.txt
```

Step 5 - 제출하기

! 윈도우 사용자분들은 **Git-Bash** 의 출력문제로 인해 화면이 정상적으로 동작하지 않을 수 있습니다.

👉 문제가 생기면 `Ctrl + c` 키로 강제종료 후 `$ winpty python -m pytest --submit` 명령어로 실행해 주세요.

```
$ python -m pytest --submit
```

Step 6 - device code 등록

처음 제출 할 때는 여러분의 컴퓨터를 서버에 등록해야합니다.

Step 5 명령어를 통해 과제를 첫 제출하시면 터미널에는 8자리 **device code** 가 출력될 것입니다.

-

```
hello_conda ~/D/w/2/a/git_flow python -m pytest --submit
===== test session starts =====
platform darwin -- Python 3.8.10, pytest-6.2.2, py-1.10.0, pluggy-0.13.1
rootdir: /Users/bsch0111/Desktop/work/2021/ai_02/git_flow, configfile: setup.cfg, testpaths: tests
plugins: metadata-1.11.0, submit-1.3.3, json-report-1.4.0
collected 1 item

tests/Part_1.py::test_example PASSED
=====
GitHub 인증을 통해 사용자를 확인하는 과정을 진행합니다.
다음 여덟자리 코드를 웹브라우저에 입력하세요.
+-----+
| 79DC-62B6 |
+-----+
이후 진행되는 로그인 과정이 완료되면 아래 메시지가 나옵니다.

-----
!!인증이 완료되었습니다!!
-----
=====
```

- device code가 출력된 후 기본 웹브라우저가 실행되면서 아래의 그림과 같이 8자리 코드를 입력할 수 있는 창이 뜹니다.

해당 칸에 8자리 device code를 입력합니다.

-



Device Activation

Enter the code displayed on your device

-

Continue

GitHub staff will never ask you to enter your code on this page.

- **Continue** 버튼을 클릭하면 웹 페이지와 터미널에서 등록이 되었다는 메시지가 나옵니다.

```
{'assignment': 'ds_sa_simple_git_flow',
  'assignment_name': 'ds_sa_simple_git_flow/2371242',
  'github_id': '2371242',
  'report': {'assignment_version': '0.1.0',
    'created': '2021-07-02 15:25:08.536733',
    'duration': '2021-07-02 15:25:08.548708',
    'environment': {'Python': '3.8.10',
      'platform': 'macOS-10.15.7-x86_64-i386-64bit'},
    'exitcode': 0,
    's3_object_url': 'https://aib-assignment-submissions.s3.ap-northeast-2.amazonaws.com/7495a6',
    'summary': {'collected': 1, 'passed': 1, 'total': 1}},
  'username': 'bsch0111'}
```

과제 제출을 완료했습니다.

----- JSON report -----
report saved to: .report.json
===== 1 passed in 89.16s (0:01:29) =====
hello_conda ~ /D/w/2/a/git_flow



Congratulations, you're all set!

Your device is now connected.

🎉 첫 제출을 완료하셨습니다!

이후에 장치 인증은 따로 진행하지 않고 `python -m pytest --submit` 을 이용하시면 여러분의 테스트 결과를 코드스테이츠 서버에 전달하실 수 있습니다.



Pytest-submit: #3. 점수

확인하기

`--score`, `--all` 명령을 통해 여러분의 과제 점수를 알 수 있습니다.

Step 1 `---score` 로 점수 확인해보기

- `--score`는 현재 디렉토리를 기반으로 점수를 확인합니다.
- **!** 여러분이 점수를 확인할 위치를 `ls -al` 명령으로 확인해서 `setup.cfg`가 있는지 확인하세요!

```
$ ls -al
```

```

hello_conda ~/D/w/2/a/git_flow ls -al
total 32
drwxr-xr-x 10 bsch0111 staff 320 7 2 15:07 ./
drwxr-xr-x 40 bsch0111 staff 1280 7 2 15:02 ../
drwxr-xr-x 12 bsch0111 staff 384 7 2 11:30 .git/
-rw-r--r-- 1 bsch0111 staff 0 7 2 11:30 .gitignore
drwxr-xr-x 6 bsch0111 staff 192 7 2 11:34 .pytest_cache/
-rw-r--r-- 1 bsch0111 staff 1036 7 2 15:28 .report.json
-rw-r--r-- 1 bsch0111 staff 252 7 2 11:30 README.md
-rw-r--r-- 1 bsch0111 staff 1805 7 2 11:30 requirements.txt
-rw-r--r-- 1 bsch0111 staff 114 7 2 11:30 setup.cfg
drwxr-xr-x 4 bsch0111 staff 128 7 2 11:34 tests/

```

\$ python -m pytest --score

- 여러분의 점수를 확인할 수 있습니다!

```

hello_conda ~/D/w/2/a/git_flow python -m pytest --score
***** bsch0111 님 *****
Assignment          Score
-----
ds_sa_simple_git_flow      3
Exit: 3
! hello_conda ~/D/w/2/a/git_flow

```

Step 2 - 모든 과제 점수 확인하기!

\$ python -m pytest --all

여러분이 section3, section5 에 제출한 모든 과제 점수를 확인할 수 있습니다.

```
hello_conda ~/D/w/2/a/git_flow python -m pytest --all

***** bsch0111 님 *****

Assignment          Score
-----
ds_sa_database      2
ds_sa_docker        1
ds_sc_31            3
ds_sa_pythonprogramming 0
pytest-submit       3
ds_sa_simple_git_flow 3
ds_sa_api           0
ds_sa_mini_flask    3
ds_sa_star_scraper  1
ds_sc_33            3
ds_sa_dp            0

Exit: 3
! hello_conda ~/D/w/2/a/git_flow
```

Step 3 - 섹션 별 점수확인하기

```
$ python -m pytest --all 3
```

여러분이 section3 에 제출한 모든 과제 점수를 확인할 수 있습니다.

```
hello_conda ~/D/w/2/a/git_flow python -m pytest --all 3

***** bsch0111 님 *****

Assignment          Score
-----
ds_sa_database      2
ds_sa_docker        1
ds_sc_31            3
pytest-submit       3
ds_sa_simple_git_flow 3
ds_sa_api           0
ds_sa_mini_flask    3
ds_sa_star_scraper  1
ds_sc_33            3

Exit: 3
```

```
$ python -m pytest --all 5
```

여러분이 section5 에 제출한 모든 과제 점수를 확인할 수 있습니다.

```
hello_conda ~/D/w/2/a/git_flow python -m pytest --all 5

***** bsch0111 님 *****

Assignment          Score
-----
ds_sa_pythonprogramming 0
ds_sa_dp              0

Exit: 3
```