



YungchAI debugging & testing manual

1. 현 프로젝트 문제점 🤖
 - 가. 디버거 미지원 🐛
 - 나. QA시 정밀한 테스트케이스 필요 🎯
2. pdb 모듈을 이용한 대화식 디버깅 💬
 - 가. pdb 튜토리얼 - 짝수 구하기 🙌
3. unittest 모듈을 이용한 unit test
 - 가. unittest 튜토리얼 - to_str 테스트 👩
4. 참고 📖

1. 현 프로젝트 문제점 🤖

가. 디버거 미지원 🐛

jupyter notebook은 디버거를 지원하지 않는다. 따라서 이제까지는 프로그램의 상태를 조사하고, 지역 변수를 출력하는 등의 행위를 위해서 print를 사용하여 일일이 출력했다.

하지만 ipynb 파일은 출력 내용까지 저장하기 때문에 과도하게 많이 출력할 경우 파일 사이즈가 폭증하게 된다.

이 경우 거대한 사이즈의 파일은 여는데 시간이 오래 걸릴 뿐만 아니라 간혹 열리지 않는 문제도 발생한다.

나. QA시 정밀한 테스트케이스 필요 🎯

관리자는 코드를 작성한 사람이 아니기 때문에 테스트케이스를 정밀하게 작성하는데 한계가 있다.

따라서 개발자가 유닛 테스트를 진행한 뒤 push하고 관리자는 적절하게 테스트가 진행되었는지 확인하는 방식으로 QA를 진행한다.

2. pdb 모듈을 이용한 대화식 디버깅

YungchAI ML project에서는 파이썬에서 지원하는 대화식 디버거 모듈인 pdb를 사용하여 디버깅한다.

디버깅시 다음과 같은 규칙을 따른다.

1. 반드시 `import pdb;pdb.set_trace()` 를 한 줄에 작성한다.
2. 코드를 push할때는 `import pdb;pdb.set_trace()` 를 주석처리하거나 삭제한다.

가. pdb 튜토리얼 - 짝수 구하기

0부터 9까지 양수 중 짝수를 모은 리스트를 반환하는 간단한 예제를 통해 pdb 모듈 사용법을 익혀보자.

```
In [1]: def get_even_nums():
        even_nums = []
        for i in range(10):
            if i % 2 == 0:
                even_nums.append(i)
        return even_nums

In [2]: if __name__ == '__main__':
        even_nums = get_even_nums()
```

우리는 5행에 정말 짝수만 들어가는지 궁금하기 때문에 `import pdb;pdb.set_trace()` 를 입력하여 아래와 같이 breakpoint를 설정한다. 그 후 실행하면 대화식 디버거가 실행된다.

```
In [3]: def get_even_nums():
        even_nums = []
        for i in range(10):
            if i % 2 == 0:
                even_nums.append(i)
                import pdb; pdb.set_trace()
        return even_nums

In [*]: if __name__ == '__main__':
        even_nums = get_even_nums()

> /var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_27991/2594381084.py(3)get_even_nums()
1 def get_even_nums():
2     even_nums = []
----> 3     for i in range(10):
4         if i % 2 == 0:
5             even_nums.append(i)

ipdb> 
```

지역 변수명을 입력하거나 내장 함수 locals를 호출하여 지역 변수 값을 확인할 수 있다.

다음 breakpoint로 이동하려면 `continue` 를 입력한다.

```
In [*]: if __name__ == '__main__':
        even_nums = get_even_nums()

> /var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_28064/2594381084.py(3)get_even_nums()
1 def get_even_nums():
2     even_nums = []
----> 3     for i in range(10):
4         if i % 2 == 0:
5             even_nums.append(i)

ipdb> i
0
ipdb> locals()['i']
0
ipdb> continue
> /var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_28064/2594381084.py(3)get_even_nums()
1 def get_even_nums():
2     even_nums = []
----> 3     for i in range(10):
4         if i % 2 == 0:
5             even_nums.append(i)

ipdb> i
2
ipdb> 
```

디버거를 중단할 때는 **q** 를 입력한다.

```
In [2]: if __name__ == '__main__':
        even_nums = get_even_nums()

> /var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_28064/2594381084.py(3)get_even_nums()
1 def get_even_nums():
2     even_nums = []
----> 3     for i in range(10):
4         if i % 2 == 0:
5             even_nums.append(i)

ipdb> i
0
ipdb> locals()['i']
0
ipdb> continue
> /var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_28064/2594381084.py(3)get_even_nums()
1 def get_even_nums():
2     even_nums = []
----> 3     for i in range(10):
4         if i % 2 == 0:
5             even_nums.append(i)

ipdb> i
2
ipdb> q
```

이 외에도 함수 호출 스택을 한 단계 낮추거나 프로그램의 다음 줄까지 실행하는 명령어도 있다.

자세한 내용은 참고 문서를 확인하길 바란다.

3. unittest 모듈을 이용한 unit test

GEA60 ML project에서는 파이썬에서 지원하는 unittest 모듈을 이용하여 unit test를 진행한다.

unit test시 다음과 같은 규칙을 따른다.

1. 테스트 코드는 아래 경로에 있는 tests 폴더 아래에 생성한다.



YungchAI_ML

L tests

L func_test.ipynb # 테스트 코드

L func.ipynb # 타겟 코드

2. 테스트 코드의 이름은 `타겟 코드_test` 이다.

3. 테스트 코드의 기본 형식은 다음과 같다.

*** 반드시 *TestCase*의 *method*는 *test* 단어로 시작해야 함**

```
import import_ipynb
from unittest import TestCase, main
from 타겟 코드 import 함수명, 클래스 등

class 타겟코드TestCase(TestCase):
    def test_함수명(self):
        self.assertEqual()
    ...

if __name__ == '__main__':
    main(argv=['first-arg-is-ignored'], exit=False)
```

가. unittest 튜토리얼 - to_str 테스트

아래와 같은 구조의 프로젝트가 있다고 하자.



YungchAI_ML

L tests

L utils_test.ipynb # 테스트 코드

L utils.ipynb # 타겟 코드

utils_ipynb는 다음과 같이 to_str이라는 함수를 가지고 있다

```
In [1]: def to_str(data):
        if isinstance(data, str):
            return data
        elif isinstance(data, bytes):
            return data.decode('utf-8')
        else:
            raise TypeError('Must supply str or bytes, found : %r' % data)
```

이를 테스트 하기 위해 tests라는 폴더에 utils_test.ipynb를 생성하고 아래와 같이 작성한다.

🔥 이때 **반드시 각 테스트는 test 단어로 시작**해야 한다.

+ 추가) TestCase 클래스는 assertion을 위한 헬퍼 메서드를 제공한다. 자세한 내용은 참고 문서를 확인하거나 help(TestCase)를 실행하여 확인하길 바란다.

```
In [48]: import import_ipynb
        from unittest import TestCase, main
        from utils import to_str

In [49]: class UtilsTestCase(TestCase):
        def test_to_str_bytes(self):
            self.assertEqual('hello', to_str(b'hello'))

        def test_to_str(self):
            self.assertEqual('hello', to_str('hello'))

        def test_to_str_bad(self):
            self.assertRaises(TypeError, to_str, object())

In [50]: if __name__ == '__main__':
        main(argv=['first-arg-is-ignored'], exit=False)
```

테스트가 하나라도 실패한다면 아래와 같은 메시지가 보일 것이다.

```
..F
=====
FAIL: test_to_str_bytes (__main__.UtilsTestCase)
=====
Traceback (most recent call last):
  File "/var/folders/c9/y3tyld_57fx2xmp_h_mhwwfr0000gn/T/ipykernel_28585/2982569299.py", line
  3, in test_to_str_bytes
    self.assertEqual('hello', to_str(b'hallo'))
AssertionError: 'hello' != 'hallo'
- hello
? ^
+ hallo
? ^

=====

Ran 3 tests in 0.006s

FAILED (failures=1)
```

테스트들이 모두 통과한다면 아래와 같은 메시지가 보일 것이다.

```
...
-----
Ran 3 tests in 0.004s

OK
```

테스트를 모두 통과하는 것을 확인하면 테스트 코드와 타겟 코드 모두를 push한다.

4. 참고

1. 똑똑하게 코딩하는 법 파이썬 코딩의 기술(Effective python), 길벗, 브렛 슬라킨 지음

<http://www.kyobobook.co.kr/product/detailViewKor.laf?ejkGb=KOR&mallGb=KOR&barcode=9791165213190&orderClick=LAG&Kc=>

2. pdb - 파이썬 디버거

<https://docs.python.org/ko/3.7/library/pdb.html>

3. unittest - 단위 테스트 프레임워크

<https://docs.python.org/ko/3/library/unittest.html>