

Graded Assignment 6.2

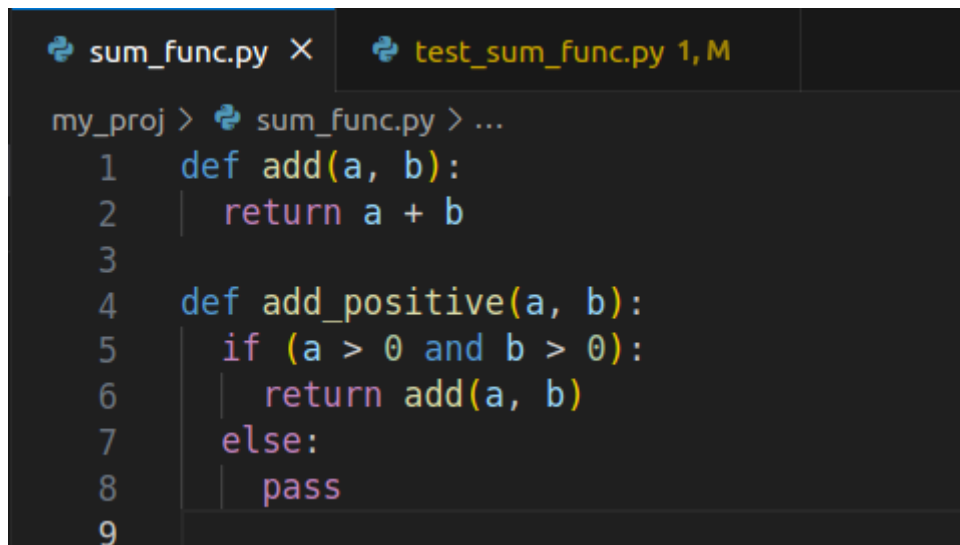
Name: Saad Sameer Khan

Employee#: 2303.KHI.DEG.034

Collaborated with: Mohammad Hamza Asim (2303.KHI.DEG.014)

sum_func.py

The function(s) that we are going to be testing.



```
sum_func.py X test_sum_func.py 1, M
my_proj > sum_func.py > ...
1  def add(a, b):
2      return a + b
3
4  def add_positive(a, b):
5      if (a > 0 and b > 0):
6          return add(a, b)
7      else:
8          pass
9
```

test_sum_func.py

The test functions we have created.

Since the 'add' function is used by the 'add_positive' function, we have directly done most of the testing on the 'add_positive' function.

```
sum_func.py  test_sum_func.py 1, M X
my_proj > tests > test_sum_func.py > ...
1  from my_proj.sum_func import add_positive, add
2  import pytest
3
4
5  # first testing add_positive function for correct outputs
6  def test_add_all_positive():
7      |   assert add_positive(5, 9) == 14
8
9  def test_add_all_negative():
10     |   assert add_positive(-4, -3) == None
11
12  def test_add_positive_and_negative():
13     |   assert add_positive(9, -3) == None
14
15  def test_add_negative_and_positive():
16     |   assert add_positive(-4, 3) == None
17
18  def test_add_both_zero():
19     |   assert add_positive(0, 0) == None
20
21  def test_add_zero():
22     |   assert add_positive(0, 7) == None
23
```

Now, we test how the 'add_positive' function handles string inputs.

```
23
24 # now testing add_positive function whether it throws errors on string inputs
25 def test_add_with_one_string_one_int():
26     with pytest.raises(TypeError):
27         add_positive("2", 3)
28
29 def test_add_with_one_int_one_string():
30     with pytest.raises(TypeError):
31         add_positive(2, "3")
32
33 def test_add_with_all_strings():
34     with pytest.raises(TypeError):
35         add_positive("2", "3")
36
```

Finally, we test 'add' function's handling of string inputs.

```
37 # now testing add function whether it throws errors on string inputs
38 def test_first_add_with_one_string_one_int():
39     with pytest.raises(TypeError):
40         add("2", 3)
41
42 def test_first_add_with_one_int_one_string():
43     with pytest.raises(TypeError):
44         add(2, "3")
45
46 '''
47     this test should fail as the add function will NOT throw an error
48     it will instead give a result "23"
49 '''
50 def test_first_add_with_all_strings():
51     with pytest.raises(TypeError):
52         add("2", "3")

```

Results

Finally, we observe the results of our tests using pytest's coverage report.

```
(base) saadsameerkhan@all-MS-7D35:~/data_engineering_bootcamp_2303/tasks/6_writing_robust_readable_and_maintainable_code/day_2_testing
○ /assignment/my_proj/tests$ pytest --cov=. test_sum_func.py
===== test session starts =====
platform linux -- Python 3.10.9, pytest-7.1.2, pluggy-1.0.0
rootdir: /home/saadsameerkhan/data_engineering_bootcamp_2303/tasks/6_writing_robust_readable_and_maintainable_code/day_2_testing/assignment/my_proj/tests
plugins: dash-2.9.3, anyio-3.6.2, cov-4.0.0
collected 12 items

test_sum_func.py .....F [100%]

===== FAILURES =====
_____ test_first_add_with_all_strings _____

    def test_first_add_with_all_strings():
>     with pytest.raises(TypeError):
E         Failed: DID NOT RAISE <class 'TypeError'>

test_sum_func.py:49: Failed

----- coverage: platform linux, python 3.10.9-final-0 -----
Name                Stmts   Miss  Cover
-----
__init__.py           0      0   100%
test_sum_func.py     32      0   100%
-----
TOTAL                 32      0   100%

===== short test summary info =====
FAILED test_sum_func.py::test_first_add_with_all_strings - Failed: DID NOT RAISE <class 'TypeError'>
===== 1 failed, 11 passed in 0.13s =====
(base) saadsameerkhan@all-MS-7D35:~/data_engineering_bootcamp_2303/tasks/6_writing_robust_readable_and_maintainable_code/day_2_testing
○ /assignment/my_proj/tests$
```

We observe that one of our tests fails, just as we predicted.