

Week 5: Unit Test

How To Test a Function:

To test a function import it to a new file used for testing
ie

I want to **test** a function used in **calculator.py**

I will make a new file called **test_calculator.py** and
in this new file at the top i will **import it like so**

from calculator import square

↳ square being the name of
the function I'm testing

then make new functions to test specific parts of the
original function

for instance to test the original function that squares a number,
you can test for positive numbers, negative numbers, and anomalies
such as zero.

```
def test_positive():
```

```
.... assert square(2) == 4
```

```
.... assert square(3) == 9
```

```
def test_negative():
```

```
.... assert square(-2) == 4
```

```
.... assert square(-3) == 9
```

```
def test_anomalies():
```

```
.... assert square(0) == 0
```

Assert - Used to claim something is true. Above I am asserting
(or saying) the square of 2 is 4. If I asserted the square
of 2 is 5 and my original function disagrees or does not output 5
when 2 is entered, it will throw an **Assertion Error**.

Pytest - From this point you can run the program using **pytest**
And it will do the rest for you

pytest test_calculator.py

Pytest for Exception

Additionally in `pytest` you can `test` for catching an `error`. To do so import `pytest` at the top of your test code and add a function for testing strings or other errors.

```
import pytest
```

```
def test_str():
```

```
.... with pytest.raises(TypeError):
```

```
    'is' square("cat")
```