



Python assertEquals

If this Python Tutorial saves you hours of work, please **whitelist it in your ad blocker** 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn how to use the Python `assertEquals()` method to test if two values are equal.

Introduction to the Python assertEquals() method

The `assertEquals()` is a method of the `TestCase` class of the `unittest` (<https://www.pythontutorial.net/python-unit-testing/python-unittest/>) module. The `assertEquals()` tests if two values are equal:

```
assertEquals(first, second, msg=None)
```

If the `first` value does not equal the `second` value, the test will fail.

The `msg` is optional. If the `msg` is provided, it'll be shown on the test result if the test fails.

Python assertEquals() method example

First, create a new module called `main.py` and define the `add()` function:

```
def add(a, b):  
    return a + b
```

Second, create a test module `test_main.py` to test the `add()` function:

```
import unittest  
  
from main import add  
  
class TestMain(unittest.TestCase):  
    def test_add(self):  
        self.assertEqual(add(1, 2), 3)
```

Third, run the test:

```
python -m unittest test_main.py -v
```

Output:

```
test_add (test_main.TestMain) ... ok
```

```
-----
```

```
Ran 1 test in 0.000s
```

```
OK
```

Python assertNotEqual() method

The `assertNotEqual()` method tests if two values are not equal:

```
assertNotEqual(first, second, msg=None)
```

If the first is equal to the second, the test will fail. Otherwise, it'll pass. For example:



```
import unittest

from main import add

class TestMain(unittest.TestCase):
    def test_add(self):
        self.assertEqual(add(1, 2), 3)

    def test_add_floats(self):
        self.assertNotEqual(add(0.2, 0.1), 0.3)
```

Run the test:

```
python -m unittest test_main.py -v
```

Output:

```
test_add (test_main.TestMain) ... ok
test_add_floats (test_main.TestMain) ... ok

-----
Ran 2 tests in 0.001s

OK
```

Since $0.2 + 0.1$ returns 0.30000000000000004 , it's not equal to 0.3 . Therefore, the following test passes:

```
self.assertNotEqual(add(0.2, 0.1), 0.3)
```

Summary

- Use the `assertEqual()` method to test if two values are equal.

- Use the `assertNotEqual()` method to test if two values are not equal.