

Object Oriented Programming with Python

Gramsci Hermozo

Session 14

- Pytest

Unit test

pytest

pytest is one of the best tools you can use to boost your testing productivity. This packages is not installed by default in python

Unit test

How to install pytest

```
$ pip install pytest
```

```
# or
```

```
$ python -m pip install pytest
```

Unit test

Create your first test

```
def func(x):  
    return x + 1  
  
def test_func():  
    assert func(3) == 5
```

Now execute the test function:

```
$ pytest
```

Unit test

Group multiple tests in a class

You may want to group them into a class.

```
class TestClass:
    def test_one(self):
        assert 1==1
    def test_two(self):
        assert x in "xgames"
```

Execute tests file:

```
$ pytest -q test_class.py
```

Note: Having each test share the same class instance would be very detrimental to test isolation and would promote poor test practices.

Unit test

Expected exception

```
import pytest

def test_zero_division():
    with pytest.raises(ZeroDivisionError):
        1/0
```

Unit test

Parametrize

Parametrization of arguments for a test function.

```
import pytest
```

```
@pytest.mark.parametrize("test_input, expected", [
    ("3+5", 8),
    ("2+4", 6),
    ("6*9", 42)])
```

```
def test_eval(test_input, expected):
    assert eval(test_input) == expected
```


Unit test

Mock

It allows you to replace parts of your system under test with mock objects and make assertions about how they have been used.