

# Python

базовый тренинг

Ружин Алексей  
[ruzin@me.com](mailto:ruzin@me.com)

# Тестирование

- doctests
- unittests
- py.test
- mock, patch

# doctests

```
def square(x):
```

```
    //////
```

```
    >>> square(2)
```

```
    4
```

```
    //////
```

```
    return x*x
```

# unittest

- `import unittest`
- ```
class XYZTestCase(unittest.TestCase):  
    def test_xyz(self):  
        self.assertEqual(3, 3)
```
- тесты класть в файлы вида: **test\_\*.py**

# unittest (2)

- `assertEqual(a, b)`
- `assertNotEqual(a, b)`
- `assertTrue(x)`
- `assertFalse(x)`
- `assertIs(a, b)`
- `assertIsNot(a, b)`
- `assertIsNone(x)`
- `assertIsNotNone(x)`
- `assertIn(a, b)`
- `assertNotIn(a, b)`
- `assertIsInstance(a, b)`
- `assertNotIsInstance(a, b)`

# unittest (3)

- `assertRaises(exception, callable, *args, **kwargs)`

# unittest (4)

- `def setUp(self):`  
    ...
- `def tearDown(self):`  
    ...
- `@classmethod`  
  `def setUpClass(cls):`  
    ...
- `@classmethod`  
  `def tearDownClass(cls):`  
    ...

# pytest

- ```
def func(x):  
    return x + 1
```
- ```
def test_func():  
    assert func(3) == 4  
    assert func(4) == 6
```



# pytest (2)

- файлы:  
test\_\*.py или \*\_test.py
- функции:  
def test\_\*()
- методы в классах:  
class TestXYZ():  
 def test\_\*()

# pytest (3)

- Fixtures
- ```
def test_func(tmpdir):  
    print(tmpdir)
```
- `pytest —fixtures`
- ```
@pytest.fixture  
def myfixture():  
    return ...  
  
def test_func(myfixture):  
    ...
```

# pytest (4)

- `@pytest.fixture(scope=«module»)`  
def fixturename():  
 print('setup code')  
 yield 'somevalue'  
 print('teardown code')

# ССЫЛКИ

- <http://docs.python-guide.org/en/latest/writing/tests/>

# Задача

- Придумать и написать тесты к задаче 2 из занятия 5 (спроектировать систему классов Сотрудник-Менеджер-Отдел).