Advanced Python.
Introduction to testing.

Теория

Виды тестов

- Unit test
- Regression
- Stress
- Integration

Unit testing

- тестируйте по возрастающей, параллельно с написанием кода
- тестируйте bottom up, сначала минимальные по объему блоки
- поймите, что вы ожидаете от теста, какие кейсы он покроет
- отслеживайте охват тестов

Unit testing

- Тесты должны писаться быстро.
- Тесты должны выполняться быстро.

Unit testing

- Assertions
- Fixtures
- Test cases, test suites
- Discovery, runners

Assertions

```
import unittest

class TestStringMethods(unittest.TestCase):

    def test_upper(self):
        self.assertEqual('foo'.upper(), 'FOO')

    def test_isupper(self):
        self.assertTrue('FOO'.isupper())
        self.assertFalse('Foo'.isupper())
```

Fixtures

```
import unittest

class ServerTestCase(unittest.TestCase):
    def setUp(self):
        self.redis = redis.Redis()
        self.es = elasticsearch.Elasticsearch()

    def tearDown(self):
        self.redis.close()
        self.es.close()
```

Pytest

Assertions

```
def f():
    return 3

def test_function():
    assert f() == 4
```

Fixtures

```
import pytest

@pytest.fixture
def smtp():
    import smtplib
    return smtplib.SMTP("smtp.gmail.com")
```

Fixtures

```
import smtplib
import pytest

@pytest.fixture(scope="module")
def smtp(request):
    smtp = smtplib.SMTP("smtp.gmail.com")
    yield smtp  # provide the fixture value
    print("teardown smtp")
    smtp.close()
```

Fixture scope

- function (default)
- class
- module
- session

Exceptions

```
import pytest

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

Marks

- parametrize(argnames, argvalues)
- skip
- skipif
- xfail