Testing with Python

for fun and profit!

with Alex Todorov, Kiwi TCMS

Kiwi TCMS

$ whoami

Alex Todorov

Career tester with almost 2 decades of experience

Twitter: @atodorov_

https://github.com/atodorov/testing-with-python

https://kiwitcms.org/

Kiwi TCMS

Introduction

- Anatomy of test automation frameworks
- pytest
- unittest.TestCase
- [Doctest]

Technical setup & pre-conditions

- Python 3 - https://www.python.org/downloads/
- A Python 3 venv - https://docs.python.org/3/library/venv.html
  - Linux (use bash):

```
$ python -m venv ~/workshop
```

```
$ source ~/workshop/bin/activate
```

  - Windows (use cmd.exe):

```
C:\> python -m venv C:\workshop
```

```
C:\> C:\workshop\Scripts\activate.bat
```

Kiwi TCMS

Technical setup & pre-conditions

- Preferably git (or zip)
- Clone (or download) the workshop repository

```
$ git clone https://github.com/atodorov/testing-with-python
```

- Install Python dependencies from the Internet
  - Note the active venv !

```
(workshop) $ cd testing-with-python/

(workshop) [testing-with-python]$ pip install -r requirements.txt
```

Kiwi TCMS

Technical setup & pre-conditions

- Verify

```
(workshop) $ python --version

(workshop) $ pytest --version
```

Kiwi TCMS

Anatomy of test automation frameworks

- Discovery & runner
  - https://docs.python.org/3/library/unittest.html#load-tests-protocol
  - Find out what tests are out there & possibly execute them
  - Usually based on naming conventions & class inheritance
- Test writing framework
  - Provides structure
  - Imposes some syntax
- Assertions library
  - Helps us check conditions

Function based testing with pytest

- https://docs.pytest.org/en/6.2.x/assert.html#the-writing-and-reporting-of-assertions-in-tests

```
def test_...():
    result = <some actions>
    assert expected == result
```

- Open ***ex01/test.py*** and add more tests inside
- Execute with `pytest -v ex01/test.py`

Kiwi TCMS

Assert on exceptions with pytest

```
import pytest

def test_...():
    with pytest.raises(IndexError):
        print(['a', 'b'][5])

@pytest.mark.xfail(raises=IndexError)
def test_f():
    print(['a', 'b'][0])
```

pytest fixtures

- https://docs.pytest.org/en/6.2.x/fixture.html
- Open ***ex02/test_without_fixtures.py*** and add more tests inside
- Copy your tests to ***ex02/test_with_fixtures.py*** and update them to use fixtures

Fixture scope in pytest

- https://docs.pytest.org/en/6.2.x/fixture.html#fixture-scopes
  - **function**: the default scope, the fixture is destroyed at the end of the test.
  - class: the fixture is destroyed during teardown of the last test in the class.
  - **module**: the fixture is destroyed during teardown of the last test in the module.
  - package: the fixture is destroyed during teardown of the last test in the package.
  - **session**: the fixture is destroyed at the end of the test session.
- `pytest --setup-show --capture=no ex03/`

Fixture setup/teardown in pytest

- https://docs.pytest.org/en/6.2.x/fixture.html#yield-fixtures-recommended
  - Use **yield** instead of **return**

```
@pytest.fixture
def sending_user(mail_admin):
    user = mail_admin.create_user()   ← setup
    yield user                        ← result
    mail_admin.delete_user(user)      ← teardown
```

Kiwi TCMS

Class based testing with unittest.TestCase

- https://docs.python.org/3/library/unittest.html

```
import unittest

class TestStringMethods(unittest.TestCase):
    def test_upper(self):
        self.assertEqual('foo'.upper(), 'FOO')
        self.assertTrue('FOO'.isupper())

if __name__ == '__main__':
    unittest.main()
```

Kiwi TCMS

Assert methods in unittest.TestCase

- https://docs.python.org/3/library/unittest.html#assert-methods

```
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)
assertRaises()           & assertRaisesRegex()
assertWarns()            & assertWarnsRegex()
assertLogs()              | assertNoLogs()
```

Kiwi TCMS

Assert methods in unittest.TestCase

- https://docs.python.org/3/library/unittest.html#assert-methods

```
assertAlmostEqual(a, b) | assertNotAlmostEqual(a, b)
assertGreater(a, b)     | assertGreaterEqual(a, b)
assertLess(a, b)        | assertLessEqual(a, b)
assertRegex(s, r)       | assertNotRegex(s, r)
assertCountEqual(a, b)
```

- Used by assertEqual() internally

```
- assertMultiLineEqual(a, b)
- assertSequenceEqual(a, b)
- assertListEqual(a, b)
- assertTupleEqual(a, b)
- assertSetEqual(a, b)
- assertDictEqual(a, b)
```

Kiwi TCMS

Exercise with unittest.TestCase

- Open ***ex04/test.py*** and add more tests inside
- Execute with `python -m unittest discover -v ex04/`

Setup/teardown with unittest.TestCase

- https://docs.python.org/3/library/unittest.html#organizing-tests
- https://docs.python.org/3/library/unittest.html#class-and-module-fixtures
- setUp / tearDown
- setUpClass / tearDownClass
- setUpModule / tearDownModule
- Execute `python -m unittest discover -v ex05/`

Kiwi TCMS

unittest.TestCase inheritance example

- https://github.com/kiwitcms/Kiwi/blob/master/tcms/bugs/tests/test_permissions.py

```
tcms.bugs.tests.test_permissions.TestNew – testdata & asserts *
tcms.tests.PermissionsTestCase - verify_get|post_with|without
tcms.tests.PermissionsTestMixin - setup/teardown/execution
tcms.tests.LoggedInTestCase – simulate logged in session
django.test.testcases.TestCase – DB wrappers & setUpTestData
django.test.testcases.TransactionTestCase – DB connection
django.test.testcases.SimpleTestCase – browser client & helpers
unittest.case.TestCase – basic structure and runtime
```

Kiwi TCMS

Test interactive examples with doctest

- https://docs.python.org/3/library/doctest.html
- Try ***python -m doctest -v ex06/example.py***
- Or see the ***doctest.testmod()*** method !

How does doctest parse docstrings ?

```
>>> # comments are ignored

>>> x = 12                  ← expression
>>> x                       ← another expression or function call
12                          ← expected result

>>> if x == 13:
...     print("yes")
... else:
...     print("NO")
...                         ← last line of multi-line expression
NO                          ← expected output (stdout)
>>>
```

Kiwi TCMS

How does doctest parse exceptions ?

- https://docs.python.org/3/library/doctest.html#what-about-exceptions

```
>>> <some expression>
Traceback (most recent call last):          ← this is required
  File "<stdin>", line 1, in <module>       ← this is ignored
ValueError: list.remove(x): x not in list ← this is matched
>>>
```

Kiwi TCMS

Write your own doctest examples

- Try ***python -m doctest -o ELLIPSIS -v ex07/bank.py***

```
Trying:
    b = BankAccount('Alex', 'Savings Account', 100, 'EUR')
Expecting nothing
ok
Trying:
    b
Expecting:
    <bank.BankAccount object at 0x...>
ok
5 items had no tests:                          ← write some examples for these
    bank.BankAccount
    bank.BankAccount.__init__
    bank.BankAccount.deposit
    bank.BankAccount.transfer
    bank.BankAccount.withdraw
```

Kiwi TCMS

setUp, tearDown & more

- https://docs.python.org/3/library/doctest.html#unittest-api

```
import unittest
import doctest
import my_module_with_doctests

def load_tests(loader, tests, ignore):
    tests.addTests(doctest.DocTestSuite(my_module_with_doctests))
    return tests
```

- Other ideas?

Kiwi TCMS

Summary

- Now you know the basics
- Python is rich with [more] test frameworks
- Go write some tests
- Make mistakes and learn from them!

Please support my open source project

- GitHub Star
- Subscribe to Newsletter
- Use Kiwi TCMS
- Contribute PRs

https://kiwitcms.org/

Kiwi TCMS

Happy Testing

Kiwi TCMS