Write Tests with **UnitTest**

Digital Career Institute

UnitTest

Python unit testing framework, based on Erich Gamma's JUnit and Kent Beck's Smalltalk testing framework.

Concepts

- **Test fixture:** represents the preparation needed to perform one or more tests, and any associated cleanup actions
- **Test case:** is the individual unit of testing. It checks for a specific response to a particular set of inputs.
- **Test suite:** is a collection of test cases, test suites, or both. It is used to aggregate tests that should be executed together.
- **Test runner:** is a component which orchestrates the execution of tests and provides the outcome to the user. The runner may use a graphical interface, a textual interface, or return a special value to indicate the results of executing the tests.

UnitTest

How to write a test:

test.py file

```
import unittest

class TestClass(unittest.TestCase):
     def test1(self):
          self.[assert case](option)

     def test2(self):
          self.[assert case](option)

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

Import unittest library

Create test class extended from unittest.TestCase class

Set the tests using asserts methods

Used if the test will run using command line

How to Run the Test?

- **Run tests from modules, classes or even individual test methods:**

```
python -m unittest test_module1 test_module2

python -m unittest test_module.TestClass

python -m unittest test_module.TestClass.test_method
```

- **Test modules can be specified by file path as well:**
  `python -m unittest tests/test_something.py`
- **You can run tests with more detail (higher verbosity) by passing in the -v flag:**
  `python -m unittest -v test_module`
- **When executed without arguments Test Discovery is started:**
  `python -m unittest`
- **When using pytest:**
  `python -m pytest`
- **When using coverage:**

```
coverage run -m unittest
coverage report -m
```

At the Core of the Lesson

Lessons Learned:

- Write tests with unittest:
  - Create simple test.
  - Different ways to run the tests:
    - from modules, classes or even individual test methods
    - by file path
    - with more detail
    - without arguments (Test Discovery)
    - with pytest
    - with coverage

Digital Career Institute

Overview of Assert Methods

Overview of Assert Methods

Digital Career Institute

DCI

assertEqual	(a, b, optional message)	Pass if a = b
assertNotEqual	(a, b, optional message)	Pass if a != b
assertTrue	(a, optional message)	Pass if a = True
assertFalse	(a, optional message)	Pass if a = False
with assertRaises	(a, optional message)	Pass if sub scope return error with type ‘a’ (TypeError, OverflowError, RecursionError, …..)
assertIs	(a, b, optional message)	Pass if a is b
assertIsNot	(a, b, optional message)	Pass if a is not b
assertIsNone	(a, optional message)	Pass if x is None
assertIsNotNone	(a, optional message)	Pass if x is None
assertIn	(a, b, optional message)	Pass if a in b
assertNotIn	(a, b, optional message)	Pass if a not in b
assertIsInstance	(a, b, optional message)	Pass if a is an instant of b
assertNotIsInstance	(a, b, optional message)	Pass if a is not an instant of b
pass		Force test to pass
fail	(error message)	Force test to fail

At the Core of the Lesson

Digital Career Institute

Lessons Learned:

- Overview of assert methods