



Prof. Jessen Vidal

UNIT TEST ON PYTHON

SHAM VINICIUS FIORIN
ANALISE E DESENVOLVIMENTO DE SISTEMAS

ROTEIRO

- Introdução
- Unittest no Python
- Setup
- Linha de comando
- Testando Insertion Sort
- Mock
- pymock
- Suite

UNITTEST

- Unittest é o primeiro *framework* para testes unitários a ser incluída na biblioteca padrão da linguagem Python;
- Fácil de usar por pessoas familiarizadas com os frameworks xUnit;
- Suporte para organização dos testes e reuso via suites de testes;

UM TESTE COM UNITTEST

```
1         import unittest
2
3         class SizeTestCase(unittest.TestCase):
4             def runTest(self):
5                 widget = Widget("The widget")
6                 assert widget.size() == (50,50), 'incorrect default size'
```

SET UP

```
1      import unittest
2
3      class SimpleWidgetTestCase(unittest.TestCase):
4          def setUp(self):
5              self.widget = widget("The widget")
6
7      class DefaultWidgetSizeTestCase(SimpleWidgetTestCase):
8          def runTest(self):
9              assert self.widget.size() == (50,50), 'incorrect default size'
10
11     class WidgetResizeTestCase(SimpleWidgetTestCase):
12         def runTest(self):|
13             self.widget.resize(100,150)
14             assert self.widget.size() == (100,150),
15                 'wrong size after resize'
```

RODANDO TESTE DA LINHA DE COMANDO

```
1         if __name__ == "__main__":  
2             unittest.main()
```

INSERTION SORT

```
import unittest

def insertion_sort(seq):
    if len(seq) <= 1:
        return seq
    local = 1
    for i in range(1, len(seq)+1):
        local = i - 1;
        while local != 0:
            if seq[local] < seq[local-1]:
                seq[local], seq[local-1] = seq[local-1], seq[local]
            local -= 1
    return seq

class OrdenacaoTestes(unittest.TestCase):
    def teste_lista_vazia(self):
        self.assertEqual([], insertion_sort([]))

    def teste_lista_unitaria(self):
        self.assertEqual([1], insertion_sort([1]))

    def teste_lista_binaria(self):
        self.assertEqual([1, 2], insertion_sort([2, 1]))

    def teste_lista_desordenada(self):
        self.assertEqual([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], insertion_sort([9, 7, 1, 8, 5, 3, 6, 4, 2, 0]))

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

```
>>> from mock import Mock
>>> myMock = Mock()
>>> myMock.greeting = "hello world"
>>> print myMock.greeting
hello world
>>>|
```

MOCK

"hello world" com mock

MOCK

```
1  >>> mock = MagicMock(side_effect=[4, 5, 6])
2  >>> mock()
3  4
4  >>> mock()
5  5
6  >>> mock()
7  6
8  |
```

MOCK

```
1  >>> vals = {(1, 2): 1, (2, 3): 2}
2  >>> def side_effect(*args):
3  ...     return vals[args]
4  ...
5  >>> mock = MagicMock(side_effect=side_effect)
6  >>> mock(1, 2)
7  1
8  >>> mock(2, 3)
9  2|
```

MOCK

```
1  >>> mock = Mock(side_effect=Exception('Boom!'))
2  >>> mock()
3  Traceback (most recent call last):
4      ...
5  Exception: Boom!
```

MOCK COM PYMOCK

```
1  from pymock import *
2  class MyTestCase(PyMockTestCase):
3
4      def testMethod(self):
5          m = self.mock()
6
7      def testMethod(self):
8          m = self.mock()
9          m.function()
10         self.replay()
11         m.function()
12         self.verify()
13
14     def testMethod(self):
15         m = self.mock()
16         m.function(5)
17         self.raiseException(Exception())
18         self.replay()
19         try:
20             m.function(5)|
21             self.fail()
22         except Exception, e:
23             pass
24         self.verify()
```

SUITE

```
1     def suite():|
2         suite = unittest.TestSuite()
3         suite.addTest(WidgetTestCase("testDefaultSize"))
4         suite.addTest(WidgetTestCase("testResize"))
5         return suite
```

REFERENCIAS

- Python Brasil. Acesso em: 07/06/2017; Disponível em: <https://wiki.python.org.br/FerramentasDeTeste>
- PURCELL; STEVE. Acesso em: 07/06/2017. Disponível em: <http://pyunit.sourceforge.net/pyunit.html>
- The Python Mock Module; Acesso em: 07/06/2017. Disponível em: <http://python-mock.sourceforge.net/>
- getting started, unittest.mock; Acesso em: 07/06/2017. Disponível em: <https://docs.python.org/3/library/unittest.mock-examples.html>



OBRIGADO