

# Python unittest

# Python

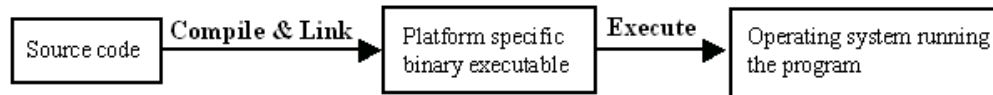
## ¿Qué es Python?



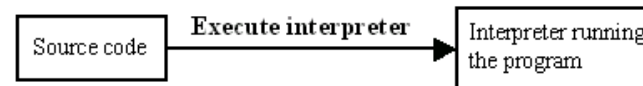
# Python

## Interpretado

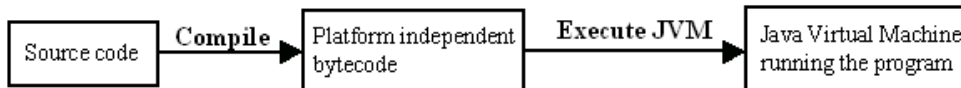
### Compiled programs



### Interpreted programs



### Java programs



# Python

## Multiparadigma

- OOP
- functional
- scripting...

# Python

## Tipado dinámico

```
1  a = 45
2  a = 'HMIS mola'
3  a = [1,2,3,4,5]
4  a = open('file.in', 'r')
5  |
```

# Python

## Multiplataforma



# Python

Rápido de escribir, fácil de leer.

```
1  fi = open('file.in', 'r')
2  fo = open('file.out', 'w')
3  |
4  fo.write(fi.read())
5
```

No echaréis de menos BufferedReader... ;)

# unittest

- unit testing framework for python
- inspired in JUnit



# Definir un testCase

- Clase que hereda de `unittest.TestCase`
- Contiene tests methods

# Definir un testSuite

- Clase que hereda de `unittest.testSuite`
- Es un conjunto de `testCases` y otros `testSuits`
- Es iterable

# TearDown y SetUp

setUpModule / setUpClass

- Queremos que se ejecute al empezar a usar un módulo/clase o cambiar a otro módulo/clase (crear objetos, conexión BBDD, crear un elemento gráfico, etc)

# TearDown y SetUp

## TearDownModule / TearDown Class

- Lo ejecutaremos al terminar de usar una clase o módulo bien porque haya finalizado o porque se haya cambiado a otro módulo/clase diferente. (Desconectar de las BBDD, hacer un dispose de un elemento gráfico, etc)

# Asserts!

Method	Checks that
<code>assertEqual(a, b)</code>	<code>a == b</code>
<code>assertNotEqual(a, b)</code>	<code>a != b</code>
<code>assertTrue(x)</code>	<code>bool(x)</code> is True
<code>assertFalse(x)</code>	<code>bool(x)</code> is False
<code>assertIs(a, b)</code>	<code>a</code> is <code>b</code>
<code>assertIsNot(a, b)</code>	<code>a</code> is not <code>b</code>
<code>assertIsNone(x)</code>	<code>x</code> is None
<code>assertIsNotNone(x)</code>	<code>x</code> is not None
<code>assertIn(a, b)</code>	<code>a</code> in <code>b</code>
<code>assertNotIn(a, b)</code>	<code>a</code> not in <code>b</code>
<code>assertIsInstance(a, b)</code>	<code>isinstance(a, b)</code>
<code>assertNotIsInstance(a, b)</code>	not <code>isinstance(a, b)</code>

# More Asserts!!

Method	Checks that
<code>assertAlmostEqual(a, b)</code>	<code>round(a-b, 7) == 0</code>
<code>assertNotAlmostEqual(a, b)</code>	<code>round(a-b, 7) != 0</code>
<code>assertGreater(a, b)</code>	<code>a &gt; b</code>
<code>assertGreaterEqual(a, b)</code>	<code>a &gt;= b</code>
<code>assertLess(a, b)</code>	<code>a &lt; b</code>
<code>assertLessEqual(a, b)</code>	<code>a &lt;= b</code>
<code>assertRegex(s, r)</code>	<code>r.search(s)</code>
<code>assertNotRegex(s, r)</code>	<code>not r.search(s)</code>
<code>assertCountEqual(a, b)</code>	<code>a</code> and <code>b</code> have the same elements in the same number, regardless of their order

# Even more Asserts!!!

Method	Used to compare
<code>assertMultiLineEqual(a, b)</code>	strings
<code>assertSequenceEqual(a, b)</code>	sequences
<code>assertListEqual(a, b)</code>	lists
<code>assertTupleEqual(a, b)</code>	tuples
<code>assertSetEqual(a, b)</code>	sets or frozensets
<code>assertDictEqual(a, b)</code>	dicts

# Alternativas

py.test - <http://pytest.org/latest/>

nose - <https://nose.readthedocs.org/en/latest/>

doctest - <https://docs.python.org/3.4/library/doctest.html>



# DEMO

A continuación veremos una demo de un ejercicio visto en clase implementado y testado con python unittest.

# Gracias por vuestra atención

Herramientas y métodos de Ingeniería del software. 2015.

Víctor Manuel García Valdeavero

Samuel Maillo Cabrerizo