Unittest

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/

¿Qué es Unit Test?

- El módulo **unittest**, a veces referido como **PyUnit**, forma parte de una serie de frameworks conocidos como *xUnit*. Estas librerías se encuentran en la mayoría de lenguajes y son casi un estándard a la hora de programar pruebas unitarias.

- https://docs.python.org/3.5/library/unittest.html

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/

¿Qué es Unit testing?

Se trata de un método para determinar si un módulo o un conjunto de módulos de código funciona correctamente.

El concepto de Unit testing no se limita a ningún lenguaje específico, sino que es una herramienta de la programación en general.

Las pruebas unitarias se implementan a la par con el desarrollo de un módulo o proyecto, y se ejecutan cuando este último sufre modificaciones para garantizar su funcionamiento. Si bien el código mismo de la prueba unitaria puede contener errores, la clave está en la separación del código de un módulo de su respectiva prueba unitaria, de modo que puedan correr independientemente.

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/

¿Qué es un Assertion?

La afirmación existe en casi todos los lenguajes. Ayudan a detectar problemas donde la causa es clara, y no más tarde como un efecto secundario de alguna otra operación.

assert es similar a lanzar una excepción si una condición dada no es verdadera.

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/

Unittest

- unittest ha estado presente de forma nativa en python desde la version 2.1.
- unittest contiene un framework para testing y un test runner.
- unittest requiere:
  - Colocar tus test en clases invocando sus metodos.
  - Utilizar el conjunto de asersiones

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/

Unittest

La documentación provee una tabla con el resto de las funciones de unittest y su respectiva operación.

Método	Comprueba que	Nuevo en
`assertEqual(a, b)`	`a == b`	
`assertNotEqual(a, b)`	`a != b`	
`assertTrue(x)`	`bool(x) is True`	
`assertFalse(x)`	`bool(x) is False`	
`assertIs(a, b)`	`a is b`	3.1
`assertIsNot(a, b)`	`a is not b`	3.1
`assertIsNone(x)`	`x is None`	3.1
`assertIsNotNone(x)`	`x is not None`	3.1
`assertIn(a, b)`	`a in b`	3.1
`assertNotIn(a, b)`	`a not in b`	3.1
`assertIsInstance(a, b)`	`isinstance(a, b)`	3.2
`assertNotIsInstance(a, b)`	`not isinstance(a, b)`	3.2

Mervin Díaz

https://www.linkedin.com/in/mervindiazlugo/