

# Python: testing con unittest

---

El módulo `unittest` viene con la biblioteca estándar de Python. Proporciona una clase llamada `TestCase`, de la que se puede derivar su clase.

## Definición de un test

Primero veamos la función que queremos probar

```
# funciones.py
def calcula_media(*args):
    return(sum(*args)/len(*args))
```

y usando `unittest` definimos el fichero de pruebas, que por convenio debe seguir el patrón `test*.py`

```
# tests.py
from funciones import calcula_media
import unittest

class TestCalculaMedia(unittest.TestCase):
    def test_1(self):
        resultado = calcula_media([10, 10, 10])
        self.assertEqual(resultado, 10)

    def test_2(self):
        resultado = calcula_media([5, 3, 4])
        self.assertEqual(resultado, 4)
```

## Ejecución del test

La opción `-v` activa el modo "verbose" y `tests` es el fichero con nuestros tests

```
(venv) @tos:~/testing$ python -m unittest -v tests.py
test_1 (tests.TestCalculaMedia) ... ok
test_2 (tests.TestCalculaMedia) ... ok

-----
Ran 2 tests in 0.000s

OK
(venv) @tos:~/testing$
```

Si los ficheros de tests están en un directorio y siguen el patrón `test*.py` se puede usar el modo "discover"

```
$ python -m unittest discover <directorio_tests>
```

## Otros assert

```
self.assertTrue('FOO'.isupper())  
self.assertFalse('Foo'.isupper())
```

## Enlaces relacionados

- <https://ellibrodepython.com/python-testing>
- <https://rico-schmidt.name/pymotw-3/unittest/index.html#estructura-basica-de-las-pruebas>
- <https://realpython.com/python-testing/#testing-your-code>
- <https://docs.python.org/3/library/unittest.html>
- <https://docs.python.org/es/3.9/library/unittest.html>