

Puesta en Producción Segura

Tarea 1: Fibonacci en Python

INDICE

[Fichero fibo.py.](#)

[Fichero test_ok.py.](#)

[Ejecución del fichero test_ok.py.](#)

[Fichero test_fail.py.](#)

[Ejecución del fichero test_fail.py.](#)

[Bibliografía](#)

Fichero fibo.py

Script escrito en Python que genera la secuencia de Fibonacci como parámetro de entrada la posición.

```
# Función que genera la secuencia de Fibonacci con un parámetro que solicita la
# posición y devuelve el valor de la posición seleccionada
# La primera posición es el 0, la segunda posición es el 1, la tercera es el 2,
# etc...
def fibonacci(n):

    a = 0
    b = 1

    if n < 0:
        print("Incorrecto")

    # Si el valor es 0 devuelve 0
    elif n == 0:
        return 0

    # Si el valor es 1 devuelve 1
    elif n == 1:
        return b

    else:
        # Sino generamos una secuencia Fibonacci hasta el valor dado
        for i in range(1, n):
            c = a + b
            a = b
            b = c

        # Devuelve el valor de la posición dada
        return b

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

Fichero test_ok.py

Script escrito en Python para realizar **pruebas unitarias** con el objetivo de verificar el correcto funcionamiento de la función fibo que está en el fichero **fibo.py** utilizando varios parámetros de entrada con resultado de OK.

```
import unittest
import fibo

class TestFibo(unittest.TestCase):

    def test_fibo(self):

        # Serie Fibonacci:      0, 1, 1, 2, 3, 5, 8
        # Posiciones números:   0, 1, 2, 3, 4, 5, 6

        # Array de dos dimensiones con el número y su posición
        lista = [(0,0),(1,1),(1,2),(2,3),(3,4),(5,5),(8,6)]

        # Recorrer el array 'lista', siendo la variable 'a' el valor y 'b' su
        # posición
        for a, b in lista:

            with self.subTest(a = a, b = b):

                # Ejecutar función fibo con el parámetro de la posición
                c = fibo.fibonacci(b)

                # Ejecutar el test
                self.assertEqual(a,c)

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

Ejecución del fichero test_ok.py con resultado OK

Probando con el fichero **test_ok.py** muestra que el test de los dos valores de cada elemento del array 'lista' son los correctos.

```
C:\test>python -m unittest -v test_ok.py
test_fibo (test.TestFibo.test_fibo) ... ok
```

```
-----
Ran 1 test in 0.000s
```

```
OK
```

```
C:\test>
```

Fichero test_fail.py

Script escrito en Python para realizar **pruebas unitarias** con el objetivo de verificar el correcto funcionamiento de la función fibo que está en el fichero **fibo.py** utilizando varios parámetros de entrada con resultado de FAIL.

```
import unittest
import fibo

class TestFibo(unittest.TestCase):

    def test_fibo(self):

        # Serie Fibonacci:      0, 1, 1, 2, 3, 5, 8
        # Posiciones números:   1, 1, 2, 4, 4, 5, 7

        # Array de dos dimensiones con el número y su posición
        lista = [(0,1),(1,1),(1,2),(2,4),(3,4),(5,5),(8,7)]

        # Recorrer el array 'lista', siendo la variable 'a' el valor y 'b' su
        posición
        for a, b in lista:

            with self.subTest(a = a, b = b):

                # Ejecutar función fibo con el parámetro de la posición
                c = fibo.fibonacci(b)

                # Ejecutar el test
                self.assertEqual(a,c)

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

Ejecución del fichero test_fail.py con resultado FAIL

Probando con el fichero **test_fail.py** muestra que el test de los dos valores de cada elemento del array 'lista' algunos de ellos no son los correctos.

```
C:\test>python -m unittest -v test_fail.py
test_fibo (test_fail.TestFibo.test_fibo) ...
  test_fibo (test_fail.TestFibo.test_fibo) (a=0, b=1) ... FAIL
  test_fibo (test_fail.TestFibo.test_fibo) (a=2, b=4) ... FAIL
  test_fibo (test_fail.TestFibo.test_fibo) (a=8, b=7) ... FAIL

=====
FAIL: test_fibo (test_fail.TestFibo.test_fibo) (a=0, b=1)
-----
Traceback (most recent call last):
  File "C:\test\test_fail.py", line 23, in test_fibo
    self.assertEqual(a,c)
AssertionError: 0 != 1

=====
FAIL: test_fibo (test_fail.TestFibo.test_fibo) (a=2, b=4)
```

```
-----  
Traceback (most recent call last):  
  File "C:\test\test_fail.py", line 23, in test_fibo  
    self.assertEqual(a,c)  
AssertionError: 2 != 3
```

```
=====
```

FAIL: test_fibo (test_fail.TestFibo.test_fibo) (a=8, b=7)

```
-----  
Traceback (most recent call last):  
  File "C:\test\test_fail.py", line 23, in test_fibo  
    self.assertEqual(a,c)  
AssertionError: 8 != 13
```

```
-----  
Ran 1 test in 0.002s
```

FAILED (failures=3)

C:\test>

Bibliografía

[Documentación oficial del Framework Unit test](#)