

# TESTING: PYTEST

Cátedra Programación II

Septiembre 2016

## 1. Pytest

Para automatizar los casos de prueba en Python vamos a utilizar la librería `pytest`. Una vez instalada la misma podemos invocar al testeo desde la línea de comandos, escribiendo:

```
1 >>> python -m pytest <nombre_del_programa>.py
```

no haciendo falta poner el `import` en el código del programa.

En el siguiente ejemplo, `prueba.py` mostramos escribir nuestra implementación para utilizar la librería **pytest** para testear la función `fmaximo`.

```
1 def fmaximo(x, y):
2     '''
3     Función que determina el máximo entre dos números.
4     Args:
5         x: Argumento de tipo numérico a comparar con el segundo ←
           argumento.
6         y: Argumento de tipo numérico a comparar con el primer ←
           argumento.
7     Returns:
8         La función retorna el mayor de sus argumentos
9     '''
10    if x > y:
11        return x
12    else:
13        return y
14
15 def test_fmaximo():
16     '''
17     Función de prueba de la función fmaximo
18     '''
19    assert fmaximo(3,4) == 4
20    assert fmaximo(-4,92) == 92
21    assert fmaximo(-10,2) == 2
22    assert fmaximo(-5,72) == 72
```

La función de prueba `test_fmaximo()` sigue el estilo de los "check expect" en Dr. Rackett. Se deben definir distintos `assert` igualando el llamado a la función con ciertos argumentos que se consideren interesantes y relevantes. En la practica, generalmente, les daremos los argumentos con los cuales deben testear las funciones.

EJERCICIO 1. Testear la función anterior, escribiendo en el interprete el siguiente comando:

```
1 >>> python -m pytest prueba.py
```

EJERCICIO 2. Modificar la implementación anterior, agregando los siguientes casos de prueba:

```
1 assert fmaximo(5,6) == 5
2 assert fmaximo(3,0) == 4
```

¿Qué sucede?, ¿Se chequean todos los asserts?

## 2. check-expect vs assert

Dr. Racket	Pytest
<pre>1 (define (fmaximo x y) 2   (if (&gt; x y) x y)) 3 (check-expect(fmaximo 5 6) 6) 4 (check-expect(fmaximo 0 -5) 0)</pre>	<pre>1 def fmaximo(x y): 2   if (x &gt; y): 3     return x 4   else: 5     return y 6 7 def test_fmaximo(): 8   assert fmaximo (5,6) == 6 9   assert fmaximo (0,-5) == 0</pre>

## Referencias

[1]