

FuncionesPython

December 10, 2020

1 Funciones en Pyhton

```
[1]: # La definición de la función  
def positivos(numeros):  
    total = 0  
    for numero in numeros:  
        if numero > 0:  
            total+=1  
    return total
```

```
[5]: # Llamada a la función  
print(positivos([23,45,21,-2,43,-45,-7, 9, 29]))
```

6

1.1 Variables globales

En el ejemplo anterior de usan variables locales. También se puede trabajar con variables globales.

```
[11]: # La definición de la función  
def imprimir(numeros):  
    for numero in numeros:  
        print(f"{numero} + {factor} = {numero+factor}")  
    return  
  
# Llamamos a la función usando la variable total ahora como una variable global  
# La definición de la función  
factor = 3  
print(imprimir([23,45,21,-2,43,-45,-7, 9, 29]))
```

```
23 + 3 = 26  
45 + 3 = 48  
21 + 3 = 24  
-2 + 3 = 1  
43 + 3 = 46  
-45 + 3 = -42  
-7 + 3 = -4
```

```
9 + 3 = 12
29 + 3 = 32
None
```

1.1.1 Devolviendo más de un valor

```
[10]: def funcion(lista, x, y):
        total_x, total_y = 0, 0
        for elemento in lista:
            if elemento == x:
                total_x += 1
            if elemento == y:
                total_y += 1
        return total_x, total_y

# Probamos la función
v_x, v_y = 2, 6
t_x, t_y = funcion([2,4,2,5,3,5,2,6], v_x, v_y)
print(f"{v_x} aparece {t_x} veces y {v_y} aparece {t_y} veces")
```

2 aparece 3 veces y 6 aparece 1 veces

1.1.2 Argumentos con valores por defecto

```
[20]: def funcion_x(lista, tipo = 2):
        r = 0
        for elemento in lista:
            if tipo == 1:
                r -= elemento
            elif tipo == 2:
                r += elemento
        return r

print(funcion_x([1,2,3,4]))
print(funcion_x([1,2,3,4], tipo = 1))
```

10
-10

1.2 Usando funciones del lenguaje

```
[12]: import numpy as np
        lista = [2,4,2,5,3,5,2,6]
        np.mean(lista)
```

[12]: 3.625

```
[13]: print(f"MAX = {np.max(lista)} - MIN = {np.min(lista)}")
```

MAX = 6 - MIN = 2

```
[15]: datos = [[1,2,3], [5,10,15], [4,6,8]]  
      # Calcula el promedio por filas  
      np.mean(datos, axis = 1)
```

[15]: array([2., 10., 6.])

```
[16]: # Calcula el promedio por columnas  
      np.mean(datos, axis = 0)
```

[16]: array([3.33333333, 6. , 8.66666667])

```
[17]: # Calcula el promedio de todos los datos  
      np.mean(datos)
```

[17]: 6.0