



01 PAO25-25 - PYTHON, Data Types



Luis Pilaguano

FUNCIONES

1.- Escribe una función que reciba como entrada una lista con números y devuelva como resultado una lista con los cuadrados de los números contenidos en la lista de entrada.

```
In [15]: def list_Numeros(numeros):  
         cuadrados = []  
         for num in numeros:  
             cuadrados.append(num ** 2)  
         return cuadrados
```

```
In [17]: listanumerica = [1,2,3,4,5,6]  
         listas = list_Numeros(listanumerica)  
         print(listas)
```

```
[1, 4, 9, 16, 25, 36]
```

2.- Escribe una función que reciba números como entrada y devuelva la suma de los mismos. La función debe ser capaz de recibir una cantidad indeterminada de números. La función no debe recibir directamente ningún objeto complejo (lista, conjunto, etc.).

```
In [19]: def suma_num (*valor):  
         suma = 0  
         for num in valor:  
             suma += num
```

```
return suma
```

```
In [21]: print(suma_num(1,2,3,4))
print(suma_num(100,2.5,5.88,200))
```

```
10
308.38
```

3.- Escribe una función que reciba un string como entrada y devuelva el string al revés. Ejemplo: si el string de entrada es 'hola', el resultado será 'aloh'.

```
In [30]: def dato_str (palabra):
dato = palabra[::-1]
return dato
```

```
In [32]: palabra = "hola"
print(dato_str(palabra))
palabra2 = "Programación"
print(dato_str(palabra2))
```

```
aloh
nóicamargorP
```

4.- Escribe una función lambda que, al igual que la función desarrollada en el ejercicio anterior, invierta el string recibido como parámetro. Ejemplo: si el string de entrada es 'hola', el resultado será 'aloh'.

```
In [38]: dato_str_lamb = lambda dato : dato[::-1]
```

```
In [39]: palabra1 = "hola"
print(dato_str_lamb(palabra1))
palabra2 = "Programación"
print(dato_str_lamb(palabra2))
```

```
aloh
nóicamargorP
```

5.- Escribe una función que compruebe si un número se encuentra dentro de un rango específico.

```
In [42]: def num_rango(valor):
if valor >= 0 and valor <= 10:
    print(f"El número {valor}, si se encuentra dentro del rango")
else:
    print(f"El número {valor}, se encuentra fuera del rango")
return valor
```

```
In [43]: valor1 = 10
valor2 = 11
print(num_rango(valor1))
print(num_rango(valor2))
```

```
El número 10, si se encuentra dentro del rango
10
El número 11, se encuentra fuera del rango
11
```

6.- Escribe una función que reciba un número entero positivo como parámetro y devuelva una lista que contenga los 5 primeros múltiplos de dicho número. Por ejemplo, si la función recibe el número 3, devolverá la lista [3, 6, 9, 12, 15]. Si la función recibe un parámetro incorrecto (por ejemplo, un número menor o igual a cero), mostrará un mensaje de error por pantalla y devolverá una lista vacía.

```
In [52]: def num_rango(valor):  
        lista = []  
        if valor > 0:  
            for i in range(1, 6):  
                lista.append(valor * i)  
        else:  
            print(f"El número {valor} no es válido (debe ser mayor que cero).")  
        return lista  
  
        # Ejemplo  
        dato = 5  
        lista = num_rango(dato)  
        print(lista)
```

[5, 10, 15, 20, 25]

7.- Escribe una función que reciba una lista como parámetro y compruebe si la lista tiene duplicados. La función devolverá True si la lista tiene duplicados y False si no los tiene.

```
In [53]: def tiene_duplicados(lista):  
        return len(lista) != len(set(lista))
```

```
In [54]: print(tiene_duplicados([1, 2, 3, 4]))  
        print(tiene_duplicados([1, 2, 2, 4, 5]))
```

False

True

8.- Escribe una función lambda que, al igual que la función desarrollada en el ejercicio anterior, reciba una lista como parámetro y compruebe si la lista tiene duplicados. La función devolverá True si la lista tiene duplicados y False si no los tiene.

```
In [55]: tiene_duplicados = lambda lista: len(lista) != len(set(lista))  
        print(tiene_duplicados([1, 2, 3]))  
        print(tiene_duplicados([1, 2, 2, 4]))
```

False

True

9.- Escribe una función que compruebe si un string dado es un palíndromo. Un palíndromo es una secuencia de caracteres que se lee igual de izquierda a derecha que de derecha a izquierda. Por ejemplo, la función devolverá True si recibe el string "reconocer" y False si recibe el string "python".

```
In [56]: def palindromo(texto):  
        return texto == texto[::-1]
```

```
In [59]: print(palindromo("reconocer"))  
        print(palindromo("python"))
```

True
False

[GitHub Funciones](#)

In []: