

*Creado por:*

*Isabel Maniega*

# Funciones

## Motivos de uso:

- Algo que no podemos repetir (mucha repetición)
- Requerimos automatizar para no repetir el código muchas veces

```
In [1]: x = 1  
y = x + 3  
y
```

Out[1]: 4

```
In [2]: x = 2  
y = x + 3  
y
```

Out[2]: 5

```
In [3]: x = 3  
y = x + 3  
y
```

Out[3]: 6

## Código repetido...

### declaramos una función:

```
In [4]: def suma(x):  
        # print(x + 3)  
        return x + 3  
  
resultado = suma(1)  
resultado
```

Out[4]: 4

```
In [5]: resultado = suma(2)  
resultado
```

Out[5]: 5

```
In [8]: resultado = suma(3)  
resultado
```

Out[8]: 6

## Realizamos un bucle for para automatizarlo

```
In [16]: for x in range(1, 6):  
         print("valor de x:", x)  
         print(suma(x))
```

```
valor de x: 1  
4  
valor de x: 2  
5  
valor de x: 3  
6  
valor de x: 4  
7  
valor de x: 5  
8
```

## Función: lambda

```
In [17]: def funcion(x):  
         return x + 1
```

```
In [18]: funcion(1)
```

```
Out[18]: 2
```

```
In [19]: (lambda x: x + 1)(3)
```

```
Out[19]: 4
```

```
In [20]: f = lambda x: x + 1  
         f(3)
```

```
Out[20]: 4
```

## Función: Creación y llamada

```
In [21]: def funcion():  
         print("Hola mundo")
```

```
In [22]: funcion()
```

```
Hola mundo
```

```
In [23]: def funcion():  
         return "Hola Mundo"
```

```
In [24]: funcion()
```

```
Out[24]: 'Hola Mundo'
```

## Función que recibe 2 variables y retorne 2 variables

```
In [32]: def variasOpciones(x, y):  
        suma = x + y  
        producto = x * y  
        return suma, producto
```

```
In [33]: variasOpciones(3, 2)
```

```
Out[33]: (5, 6)
```

```
In [35]: SUMA, PRODUCTO = variasOpciones(3, 2)  
SUMA
```

```
Out[35]: 5
```

```
In [36]: PRODUCTO
```

```
Out[36]: 6
```

```
In [37]: SUMA, PRODUCTO
```

```
Out[37]: (5, 6)
```

```
In [ ]: # OJO con el orden que retornamos las variables  
# return suma, producto  
# al mostrar la variable deben seguir el mismo orden suma, producto = var
```

```
In [42]: def variasOpciones(x, y):  
        suma = x + y  
        producto = x * y  
        return producto, suma
```

```
In [43]: SUMA, PRODUCTO = variasOpciones(3, 2)  
SUMA, PRODUCTO
```

```
Out[43]: (6, 5)
```

**Error al mostrar la información!!!**

## Funciones Recursivas

Es una técnica donde una función se invoca a sí misma.

La serie fibonnacci es un claro ejemplo de recursividad:

$Fib\ i = Fib\ i-1 + Fib\ i-2$

El número  $i$  se refiere al número de  $i-1$ , y así sucesivamente hasta llegar a los primeros dos.

Se puede crear una funcion Fib() para usar la recursividad:

```
In [44]: def fib(n):  
        if n < 1:  
            return None  
        if n < 3:  
            return 1  
        return fib(n-1) + fib (n-2)
```

**Este programa necesita de una condición que detenga el bucle infinito, esto ocasiona un consumo alto en memoria y por lo tanto pueden ser en ocasiones ineficientes.**

## Variables Locales y Globales

```
In [48]: # Podemos cambiar el valor de una variable
```

```
In [58]: x = 6  
        print(x)
```

6

```
In [59]: # volvemos a definir el valor de x, el valor de x pasa a valer 5  
        x = 5  
        print(x)
```

5

```
In [60]: def funcion_cambiar_x():  
        x = 6  
        #print(x)  
        return x
```

```
In [61]: funcion_cambiar_x()
```

Out[61]: 6

```
In [62]: print(x)
```

5

```
In [63]: y = 6  
        print(y)  
        y = 5  
        print(y)  
  
        def cambiar_y():  
            y = 3  
            return y  
  
        print(cambiar_y())  
  
        print(y)
```

6  
5  
3  
5

```
In [64]: def funcion_cambiar_x():  
         global x  
         x = 6  
         # print(x)  
         return x
```

```
In [65]: funcion_cambiar_x()
```

Out[65]: 6

```
In [66]: print(x)
```

6

```
In [67]: y = 6  
         print(y)  
         y = 5  
         print(y)  
  
         def cambiar_y():  
             global y  
             y = 3  
             return y  
  
         print(cambiar_y())  
  
         print(y)
```

6  
5  
3  
3

## Break, continue, pass - For -

- **BREAK**

```
In [1]: L = [5, 10, 15, 20, 25, 30, 35]  
        L
```

Out[1]: [5, 10, 15, 20, 25, 30, 35]

```
In [2]: for numero in L:  
         if numero == 20:  
             print("\n")  
             break  
         else:  
             print(numero) # mostrar: 5, 10, 15  
print("hemos llegado al 20, y salida del bucle FOR")
```

```
5
10
15
```

hemos llegado al 20, y salida del bucle FOR

- **CONTINUE**

```
In [3]: L = [5, 10, 15, 20, 25, 30, 35]
L
for numero in L:
    if numero == 20:
        print("hemos llegado al valor 20, y CONTINUO (SIN IMPRIMIRLE)")
        continue
    else:
        print(numero) # mostrar: 5, 10, 15, 25, 30, 35
```

```
5
10
15
hemos llegado al valor 20, y CONTINUO (SIN IMPRIMIRLE)
25
30
35
```

- **PASS**

```
In [7]: def funcion():
        # TODO: funcion de suma de variables
        pass
        # pendiente de describir la actividad de la función
funcion()
```

```
In [9]: L = [5, 10, 15, 20, 25, 30, 35]
L
for numero in L:
    if numero == 20:
        # print("hemos llegado al valor 20, y CONTINUO (SIN IMPRIMIRLE)")
        pass
    else:
        print(numero) # mostrar: 5, 10, 15, 25, 30, 35
```

```
5
10
15
25
30
35
```

## Menús

```
In [11]: L = []

def insertar(elemento):
    L.append(elemento)
```

```

def eliminar():
    L.remove(L[-1])

def consultar():
    print("\n")
    print("Los numeros que tiene en este momento son: ")
    print(L)
    print("\n")

while True:
    print("\n")
    print("***** MENU *****")
    print("*****")
    print("***** 1. Insertar (nuevo elemento) *****")
    print("***** 2. Eliminar (último elemento) *****")
    print("***** 3. Consultar (toda la lista) *****")
    print("***** 99. Salir (del menú) *****")
    print("*****")

    print("\n")

    opcion = int(input("Inserte su opción: "))

    if opcion == 1:
        # recoger el valor a insertar con elemento
        elemento = input("Inserte el nuevo número: ")
        # ir a la funcion insertar
        insertar(elemento)
    elif opcion == 2:
        if len(L) != 0:
            # ir a la funcion eliminar
            eliminar()
        else:
            print("\n")
            print("no tiene elementos para eliminar")
            print("\n")
    elif opcion == 3:
        # ir a la funcion consultar
        consultar()
    elif opcion == 99:
        break
    else:
        print("por favor, escriba una opción correcta. ")
        print("\n")

```

```

***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****

```

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

Los numeros que tiene en este momento son:  
['102']

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

Los numeros que tiene en este momento son:  
['102', '204']

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```



por favor, escriba una opción correcta.

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

Los numeros que tiene en este momento son:  
['102']

```
***** MENU *****
*****
***** 1. Insertar (nuevo elemento) *****
***** 2. Eliminar (último elemento) *****
***** 3. Consultar (toda la lista) *****
***** 99. Salir (del menú) *****
*****
```

## Main en python

```
In [ ]: # sirve para simular el int main() de otros lenguajes de programación
        # un ejemplo de Main en otros lenguajes como C, sería el siguiente:
```

```
In [12]: from IPython.display import Image
        Image("ejemploMainC.png")
```

Out[12]:

```
1 //FACTORIAL CON FUNCIONES
2
3 #include <stdio.h>
4 //Variables globales
5 //deben ser reconocidas en ambas funciones
6 int num;
7 int y=0; //simplemente recoge el valor de la variable
8 int factorial=1;
9
10 int main()
11 {
12     printf("\nIntroduzca un numero y le diré su factorial:\n");
13     scanf("%d", &num);
14     y=num;
15
16     //AQUI LLAMA A LA FUNCION CREADA
17     funcion_factorial();
18     //UNA VEZ AQUI EJECUTA SIN MAS
19     printf("El factorial de %d es %d\n", y, factorial);
20 }
21
22 int funcion_factorial()
23 {
24     for(num=num; num>=1; num--)
25     {
26         factorial=factorial*num;
27     }
28 }
```

## Ejemplo 1

```
In [13]: def main():
          print("Estamos en la funcion main()")

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

Estamos en la funcion main()

## Ejemplo 2

```
In [17]: def cuadrado(x):
          return x * x

          def main():
              print("Estamos en la funcion main()")

          def funcion():
              print("test")

          if __name__ == "__main__":
              print(cuadrado(8))
              main()
```

64

Estamos en la funcion main()

# Try - Except

```
In [ ]: # Ejecutamos todo el código de una sola pasada  
# para ver como funciona el except  
  
# Sirve para testear errores en el código  
  
# de tal manera que no para todo el programa al detectar un error
```

```
In [ ]: # Asumimos que teniamos:  
# una variable "x" que apareció anteriormente  
# una variable "w" que no apareció previamente (NO DECLARADA)
```

```
In [18]: x = [10, 20, 30, 40]  
x
```

```
Out[18]: [10, 20, 30, 40]
```

```
In [19]: try:  
        print(s)  
except Exception as e:  
    print("Error: %s" % str(e))  
    print(type(e))
```

Error: name 's' is not defined  
<class 'NameError'>

```
In [20]: # Excepciones según error  
try:  
    print(s)  
except NameError:  
    print("error en el nombre no definido")  
except Exception as e:  
    print("Error: %s" % str(e))  
    print(type(e))
```

error en el nombre no definido

```
In [21]: w = 25
```

```
In [22]: try:  
        print(w)  
except Exception as e:  
    print("Error: %s" % str(e))
```

25

```
In [23]: try:  
        print(x)  
except Exception as e:  
    print("Error: %s" % str(e))
```

[10, 20, 30, 40]

Creado por:

Isabel Maniega