8_Clase y Funciones

November 1, 2025

Creado por:

Isabel Maniega

1 Funciones

1.0.1 Motivos de uso:

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

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

[1]: 4

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

[2]: 5

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

[3]: 6

Código repetido...

declaramos una función:

[6]:
```
def suma(x):
    #print(x  + 3)
    return x + 3

resultado = suma(1)
resultado
```

[6]: 4

[7]:
```
resultado = suma(2)
resultado
```

[7]: 5

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

[8]: 6

Realizamos un bucle for para automatizarlo

[9]:
```
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
```

1.1 Función: lambda

[10]:
```
def funcion(x):
    return x + 1
```

[11]:
```
funcion(1)
```

[11]: 2

[12]:
```
(lambda x: x + 1)(3)
```

[12]: 4

[13]:
```
f = lambda x: x + 1
f(3)
```

[13]: 4

[14]:
```
def funcion(x, y):
    return y * x + 1
```

```
funcion(3, 2)
```

[14]: 7

[15]:
```
f = lambda x, y: y * x + 1
f(3, 2)
```

[15]: 7

1.2 Función: Creación y llamada

[16]:
```
def funcion():
    print("Hola mundo")
```

[19]:
```
funcion()
```

```
Hola mundo
```

[20]:
```
def funcion():
    return "Hola Mundo"
```

[21]:
```
funcion()
```

[21]: 'Hola Mundo'

1.3 Función que recibe 2 variables y retorne 2 variables

[22]:
```
def variasOpciones(x, y):
    suma = x + y
    producto = x * y
    return suma, producto
```

[27]:
```
variasOpciones(3, 2)
```

[27]: (5, 6)

[28]:
```
SUMA, PRODUCTO = variasOpciones(3, 2)
SUMA
```

[28]: 5

[25]:
```
PRODUCTO
```

[25]: 6

[29]:
```
SUMA, PRODUCTO
```

[29]: (5, 6)

[ ]:
```
# OJO con el orden que retornamos las variables
# return suma, producto
# al mostrar la variable deben seguir el mismo orden suma, producto =␣
 ↪variasOpciones(x, y)
```

[30]:
```
def variasOpciones(x, y):
    suma = x + y
    producto = x * y
    return producto, suma
```

[31]:
```
SUMA, PRODUCTO = variasOpciones(3, 2)
SUMA, PRODUCTO
```

[31]: (6, 5)

Error al mostrar la información!!!

1.4 Argumento múltiple en funciones

[32]:
```
def suma(x=1, y=3):
    return x + y


suma()
```

[32]: 4

[33]:
```
def suma(x=1, y=3):
    return x + y


suma(2, 6)
```

[33]: 8

[34]:
```
def suma(x=1, y=3):
    return x + y


suma(y=2, x=6)
```

[34]: 8

[35]:
```
def suma(x=1, y=3):
    return x + y


suma(3, y=6)
```

[35]: 9

[36]:
```
def suma(x=1, y=3):
    return x + y


suma(3)
```

[36]: 6

[37]:
```
def suma(x=1, y=3):
    return x + y


suma(y=3)
```

[37]: 4

[38]:
```
def suma(x=1, y=3):
    return x + y


suma(x=2, 6)
```

```
  Cell In[38], line 5
    suma(x=2, 6)
                ^
SyntaxError: positional argument follows keyword argument
```

1.5 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 asi sucesivamente hasta llegar a los primeros dos.

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

[ ]:
```
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.

[1]:
```
def func(x, y):
    if x == y:
        return x
    else:
        return func(x, y-1)


print(func(0, 3))

# func(0, 3)  => return func(x, y-1) ==> return func(0, 2)
# func(0, 2)  => return func(0, 1)
# func(0, 1)  => return func(0, 0)
# func(0, 0)  => return x => 0
```

```
0
```

2 Args vs Kwargs

[2]:
```
# *args: son todos los valores numericos, devueltos como una tupla
# **kwargs: son todos los valores asignados como variables devueltos como␣
 ↪diccionario:

def new_func(x=1, y=2, *args, **kwargs):
    z = x + y
    return z, args, kwargs

new_func(2, 2, 2, 3, 4, singleton=25, decorador='Hola')
```

[2]:
```
(4, (2, 3, 4), {'singleton': 25, 'decorador': 'Hola'})
```

2.0.1 Variables Locales y Globales

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

[39]:
```
x = 6
print(x)
```

```
6
```

[40]:
```
# volvemos a definir el valor de x, el valor de x pasa a valer 5
x = 5
print(x)
```

```
5
```

[41]:
```
def funcion_cambiar_x():
    x = 6
    #print(x)
    return x
```

[42]:
```
funcion_cambiar_x()
```

[42]: 6

[43]:
```
print(x)
```

```
5
```

[44]:
```
y = 6
print(y)
y = 5
print(y)

def cambiar_y():
    y = 3
    return y

print(cambiar_y())

print(y)
```

```
6
5
3
5
```

[45]:
```
y = 6
print(y)
y = 5
print(y)

def cambiar_y():
    y = 3
    return y

print(cambiar_y())

print(y)
```

```
6
5
3
5
```

[46]:
```
def funcion_cambiar_x():
    global x
    x = 6
    # print(x)
    return x
```

[47]:
```
funcion_cambiar_x()
```

[47]: 6

[48]:
```
print(x)
```

```
6
```

[49]:
```
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
```

2.1 Break, continue, pass - For -

- **BREAK**

[50]:
```
L = [5, 10, 15, 20, 25, 30, 35]
L
```

[50]: [5, 10, 15, 20, 25, 30, 35]

[51]:
```
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**

[54]:
```
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**

[59]:
```
def funcion():
    # TODO: funcion de suma de variables
    pass
    # pendiente de describir la actividad de la función
funcion()
```

[55]:
```
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
hemos llegado al valor 20, y CONTINUO (SIN IMPRIMIRLE)
25
30
35
```

2.2 Menús

[60]:
```
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)        ******
*****************************************


Inserte su opción:  1
Inserte el nuevo número:  10


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


Inserte su opción:  1
Inserte el nuevo número:  20


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


Inserte su opción:  3
```

```
Los numeros que tiene en este momento son:
['10', '20']



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


Inserte su opción:  2


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


Inserte su opción:  3


Los numeros que tiene en este momento son:
['10']



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

```
Inserte su opción:  4
por favor, escriba una opción correcta.



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


Inserte su opción:  2


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


Inserte su opción:  3


Los numeros que tiene en este momento son:
[]



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

```
Inserte su opción:  2


no tiene elementos para eliminar



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


Inserte su opción:  99
```

2.3 Main en python

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

2.4 Try - Except

[ ]:
```
# 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
```

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

[61]:
```
x = [10, 20, 30, 40]
x
```

[61]:
```
[10, 20, 30, 40]
```

[62]:
```
try:
    print(s)
except Exception as e:
    print("Error: %s" % str(e))
```

```
    print(type(e))
```

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

[63]:
```
# 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
```

[64]:
```
w = 25
```

[65]:
```
try:
    print(w)
except Exception as e:
    print("Error: %s" % str(e))
```

```
25
```

[66]:
```
try:
    print(x)
except Exception as e:
    print("Error: %s" % str(e))
```

```
[10, 20, 30, 40]
```

Creado por:

Isabel Maniega