# **Funciones propias.**

Para iniciar nuestra propia función usaremos “def” seguido del nombre de nuestra función son paréntesis para despues darle parámetros.

def saludar():

    print("Hola que tal")

saludar()

Aquí lo que hacemos es definir nuestra función que simplemente es de un saludo, y para llamarla simplemente usamos el nombre de nuestra función en este caso sin parámetros. Y asi podemos llamar a esta función en cualquier momento.

Pero como dijimos antes podemos dar parámetros a nuestras funciones.

def saludar(nombre,apellido):

    print(f"Hola {nombre} {apellido}, como estas?")

saludar("Alexis", "Moles")

Declaramos las variables como parámetros y despues al momento de llamar nuestra función especificamos el valor de nuestras variables.

def suma(a, b):

    c = a + b

    print(f"El resultado de tu suma es: {c}")

suma(45, 34)

def saludo(nombre, sexo):

    sexo = sexo.lower()

    if (sexo == "mujer"):

        adjetivo = "preciosa"

    elif (sexo == "hombre"):

        adjetivo = "precioso"

    else:

        adjetivo = "Crack"

    print(f"Hola {nombre}, que tal estas {adjetivo}")

saludo("Alexis", "Hombre")

Si yo por ejemplo creo una función que me devuelve un número, yo tengo que hacer cosas con ese número, no siempore mostrárselo al usuario. Si yo saco un numero de una base de datos que es privado, y le hago cálculos, no quiero mostrarlo en consola y lo quiero para uso propio.

Ahora vamos a crear una función que retorne valores, pongamos esto con un ejemplo de un generador de contraseñas:

def generador\_contraseña(num):

    chars = """AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

    @\_."""

    num\_entero = str(num)

    num = int(num\_entero[0])

    c1 = num - 2

    c2 = num

    c3 = num - 5

    c4 = num

    c5 = num - 3

    contraseña = f"{chars[c1]}{chars[c2]}{chars[c3]}{chars[c4]}{chars[c5]}{num \* 3}"

    return contraseña

generador\_contraseña(546)

Ahora expliquemos todo nuestro codigo primero que nada

    chars = """AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

    @\_.1234567890""" *#Creamos lista de caracteres para la contraseña*

La variable chars va a ser la que contendrá todos nuestros caracteres para generar nuestra contraseña.

    num\_entero = str(num)

    num = int(num\_entero[0])

Aquí lo que estamos haciendo es convertir el parámetro num a una cadena y almacenándolo en la variable num, para despues volver a llamar la variable num y guardar la conversión y agarrar solo el primer valor, o sea da igual que ingresemos 123, 34, 43 ,80, que solo va agarrar el primer valor, en este caso 1, 3, 4, 8. Entonces con esto lo que hacemos es que el parámetro que elegimos será el índice para crear nuestra contraseña

    num = int(num\_entero[0])

    c1 = num - 2

    c2 = num

    c3 = num - 5

Esto hara que en base al primer numero de num, se restara en c1 -2 para que sea el carácter, o sea detallemos mas esto:

Si en num elegimos 72, se seleccionara el 7 como índice, entonces el índice será tomado de la lista de caracteres en este caso se tomara el “d”, entonces para el c1 (carácter 1 de la contraseña), se le restara al índice 2 para que se seleccione el primer carácter por lo que quedara seleccionado “c”, despues para el c2 sera el mismo “d”, y para el tercer carácter se le restaran 5 posiciones y por lo tanto será “B”

    contraseña = f"{chars[c1]}{chars[c2]}{chars[c3]}{num \* 2}"

Por lo tanto, se almacenarán en la variable contraseña de esta manera, especificando que de la variable chars se elija la posición que se especificó con los “cn” y por último multiplicamos nuestro índice original o sea 7 por 2

Por lo tanto, la contraseña es: cdB14

Pero yo quiero usar este generador para un uso interno, y no queremos que el usuario lo vea, para hacer que intima nuestra, usamos la función return, esto quiere decir que nuestra función, se va convertir en un valor que podemos asignar a una variable u otras cosas. Y al momento de ejecutarlo no mostrara nada, pero guardara el valor. Para imprimirlo y que se muestre ya que es un valor lo podemos guardar en una variable.

password = generador\_contraseña(9)

frase = f"Tu contraseña nueva es: {password}"

print(frase)

Podemos guardarlo dentro de una variable, de esta manera podemos retornar valores

## **Parámetro args.**

def suma(a,b):

    return a + b

resultado = suma(20, 23)

print(resultado)

Aquí tenemos una función de suma con los parámetros ya definidos y todo, pero aquí queremos agregar mas valores, tenemos varias formas de hacerlo:

Por la forma no optima de sumar valores:

def suma(lista):

    numeros\_sumados = 0

    for numero in lista:

        numeros\_sumados = numeros\_sumados + numero

    return numeros\_sumados

resultado = suma([20, 23, 12, 3, 43, 65, 12])

print(resultado)

Aquí se hace por medio de un bucle.

Ahora utilizaremos el parámetro args:

def suma(\*numeros):

    numeros\_sumados = 0

resultado = suma(23, 10, 11, 12)

print(resultado)

Este ejemplo no esta terminado, pero aquí agregamos un \* antes del parámetro números, para indicar que a todos los parámetros que pongamos, se convertirán en un solo parámetro, o sea el “23, 10, 11, 12” son parámetros diferentes, pero con el \* o args indicamos que se convierten en uno solo.

def suma(\*numeros):

    return sum(numeros)

resultado = suma(23, 10, 11, 12)

print(resultado)

Entonces aquí usamos sum para sumar todos los iterables, y retornamos el valor y por lo tanto el resultado será: 56, que es la suma de todos los números que indicamos como parámetros.

def suma(nombre,\*numeros):

    return f"{nombre} la suma de tus numeros es {sum(numeros)}"

resultado = suma("Sergio", 23, 10, 11, 12)

print(resultado)

El parámetro args, se debe colocar al parámetro final, o sea aquí decimos, el primer valor va a ser el nombre, y lo que viene despues va a ser una lista de números.

Otra manera de hacerlo seria:

def suma(numeros):

    return f"la suma de tus numeros es {sum([\*numeros])}"

resultado = suma([23, 10, 11, 12])

print(resultado)

En las funciones se debe de respetar el orden de los parametros, pero podemos forzar parámetros

def saludo(nombre, apellido, adjetivo):

    return f"Hola {nombre} {apellido}, eres muy {adjetivo}"

frase = saludo(*adjetivo* = "guapo", *apellido* = "Moles", *nombre* = "Sergio")

print(frase)

De esta manera podemos forzar los parámetros, a estos se les conoce como key arguments.

Tambien podemos definir un valor a nuestro parámetro desde la función.

def saludo(nombre, apellido, adjetivo = "tonto"):

    return f"Hola {nombre} {apellido}, eres muy {adjetivo}"

frase = saludo(*apellido* = "Moles", *nombre* = "Sergio")

print(frase)

Pero claro como variables podemos redefinirlas:

def saludo(nombre, apellido, adjetivo = "tonto"):

    return f"Hola {nombre} {apellido}, eres muy {adjetivo}"

frase = saludo(*apellido* = "Moles", *nombre* = "Sergio", *adjetivo* = "inteligente")

print(frase)