Funciones en Python ()

Las funciones en python son bloques de codigo reutilizables que se ejecutan cuando son llamadas. Ayudan a organizar el codigo, lo hacen mas legible y ¡permiten reutilizar código!

Las Funciones tienen las siguientes ventajas:

- Reutilizacion de Código: Las funciones permiten escribir una vez y se pueden usar en cualquier parte del codigo.Esto evita duplicación de código y facilita el mantenimiento.
- Organizacion y Legibilidad: Las funciones ayudan a organizar el código en bloques lógicos._ Al dar nombres descriptivos a las funciones, el código se vuelve más legible y comprensible.
- Abstraccion: Las funciones ocultan los detalles de implementacion.
- Modularidad: Divide un problema grande en problemas más pequeños.

> En resumen, las funciones son una herramienta poderosa en Python. Usalas con moderación y de manera inteligente para aprovechar sus beneficios mientras evitas sus desventajas.

Sintaxis de una Función

```
def nombre_de_la_funcion(parametros):
    #codigo de la función
    return valor_de_retorno
```

- def: es la ***palabra clave*** que indica la definicion de una función.
- nombre*de_la_funcion: es el identificador ***único_** que usamos para llamar a la funcion. `En Python, la convención para nombrar funciones es usar letras minúsculas con palabras separadas por guiones bajos. Esto se conoce como snake_case.`
- parametros: son variables opcionales que pasamos a la función.
- return: es la ***palabra clave*** que devuelve valor desde la funcion al llamador.

Decoradores de Funciones

Dentro de las funciones podemos "decorarla" para ayudarnos en un futuro a recordar que hace, declarando los datos que espera y la salida que pensamos darle, esto es simplemente visual ya que Python no es fuertemente tipado, por lo que dentro de la funcion tendremos que hacer la logica para que por ejemplo de un str obtengamos en el return un int.

Estos decoradores van en los parametros donde despues del nombre de la variable colocamos ***:*** y el tipo de dato esperado y al final de la declaracion con ***->*** podemos colocar la salida esperada, en este caso texto esperamos que sea un str y valor_de_retorno un int.

```
def transformar_str_int(texto:str)->int:
    #codigo de la función
    return valor_de_retorno
```

Pero no solo eso, tambien podemos agregarle un atributo extra en los parametros para poner por defecto en caso de que el usuario no coloque nada y sean si o si necesarios para la logica de la funcion. usando la siguiente expresion ***| None = (valor por defecto)***

```
def numero (texto_input:str|None= "Ingrese un numero:")-> int:
    while True:
        numero_1 = input (texto_input)
        if numero_1.isdigit():
            numero_1 = int(numero_1)
            if numero_1 > 0:
                break
            else:
                print("El valor ingresado debe ser mayor a 0")
        else:
            print("El valor ingresado no es un Nro")
    return numero_1
    cantidad = numero("Ingrese la cantidad de veces a calcular:")
```

En este caso cuando le pasamos un parametro a nuestra funcion numero() tomará lo que tiene dentro de los parentesis como parametro para imprimir el primer mensaje.

```
def calcular(cantidad:int|None=2):
    mayor = 0

    for i in range(cantidad):

        numero_1 = numero()
        print(f"El numero ingresado es:{numero_1}")
        if numero_1 > mayor:
            mayor = numero_1
    print(f"El numero mayor es : {mayor}")
    return
```

En este otro ejemplo realizamos una funcion que se encargue de hacer bucles y buscar el mayor entre X cantidad de numeros, en caso de no especificarselo dentro de los parentesis nos dará dos como predeterminado.

calcular() #Da dos Intentos.

calcular(5) #Da cinco Intentos.

Ejemplos de Uso

Una simple funcion que suma dos números:

```
def sumar(a, b):
    return a + b
 resultado = sumar(5,3)
 print(resultado) #Devuelve: 8
```

Buenas Practicas con Funciones

- Nombres Descriptivos: Recuerda elegir nombres que describan claramente lo que hace la funcion.
- Documentacion: Usa documentos o comentarios para explicar que hace la funcion y que parametros espera(te será de ayuda dias despues de programar algo)
- Codigo limpio: Manten las funciones enfocadas en una sola accion o proposito.
- Las funciones son herramientas esenciales que te permiten escribir código más modular y mantenible.
- Recorda que los nombres van en snake_case (ej: mi_funcion)

Funcion Lambda

(No las vamos a usar ni a preguntar, pero esta bueno que sepan que existen)

Las funciones lambda son una forma de crear funciones anónimas en una sola línea. Son utiles cuando necesitas una funcion por corto periodo de tiempo y no queres definirla con *def*

Su Sintaxis es:

```
lambda parametros: expresion
```

Para ver la diferencia con una funcion normal, volveremos a usar la que ya vimos "sumar" con lambda se veria asi:

```
 sumar = lambda a,b : a + b
print(sumar(5,3)) #Devuelve: 8
```

✅Ventajas de las funciones Lambda✅

- Son útiles para funciones simples y temporales.
- Se pueden usar en expresiones como argumentos de otras funciones.
- Reducen la necesidad de declarar funciones completas.

❌Desventajas de las funciones Lambda❌

- Son limitadas en funcionalidad(solo una expresión)
- No son tan legibles como las funciones regulares.
- No tienen nombres descriptivos.

Actividades ✍️

Crea una funcion que nos salude por consola.

Crea una funcion para calcular el área de un cuadrado.

Crea una funcion para ordenar una lista de palabras.

Función Verificar si un número es par o impar.

Función de conversión de Temperatura. (Facil) ✍️

Escribe una función que convierta la temperatura de Celsius a Fahrenheit y viceversa. La función debe aceptar dos argumentos: *el valor de la temperatura y una cadena* que indique si se está convirtiendo a Celsius('C') o a Fahrenheit('F').

> Una ayuda practica ¡se necesita usar if, elif y else! formula celsius (temperatura - 32) * 5/9 formula Fahrenheit = temperatura * 9/5 + 32

Funcion de Calculo Factorial. (Facil) ✍️

Escribe una funcion que calcule el factorial de un numero dado.

> El factorial de un número entero positivo se define como el producto de todos los enteros positivos menores o iguales al número, Ejemplo: Factorial de 4 = 24(4 x 3 x 2 x 1)

Funcion de Secuencia Fibonacci (Medio)😐

Desarrolla una funcion que genere una lista con los primeros 'n' numeros de la secuencia de Fibonacci, donde cada numero es la suma de los dos anteriores comenzando con 0 y 1.

> Ejemplo el decimo número de la secuencia es: 34

Funcion de encontrar el maximo entre 3 numeros

cree una funcion que permita ingresar 3 números y nos muestre cual es el mayor

Generar una lista de números aleatorios:

Generar un programa que permita el ingreso de la cantidad de numeros aleatorios a generar y los muestre.

VOLVER