

Funciones en Python: Bloques de Código Reutilizables

Esta presentación refuerza conceptos clave sobre funciones en Python. Veremos definición, parámetros, valores de retorno y técnicas avanzadas.



by Martina Zabala

¿Qué es una Función?

Definición

Un bloque de código reutilizable que realiza una tarea específica.

Ayuda a organizar y modularizar tu código.

Sintaxis Básica

```
def saludar():  
    print("¡Hola, mundo!")  
  
# Llamada a la función  
saludar()
```

Parámetros y Retorno



Entrada (Parámetros)

Valores que recibe la función para trabajar.

```
def saludar(nombre):  
    print(f"Hola, {nombre}")
```



Procesamiento

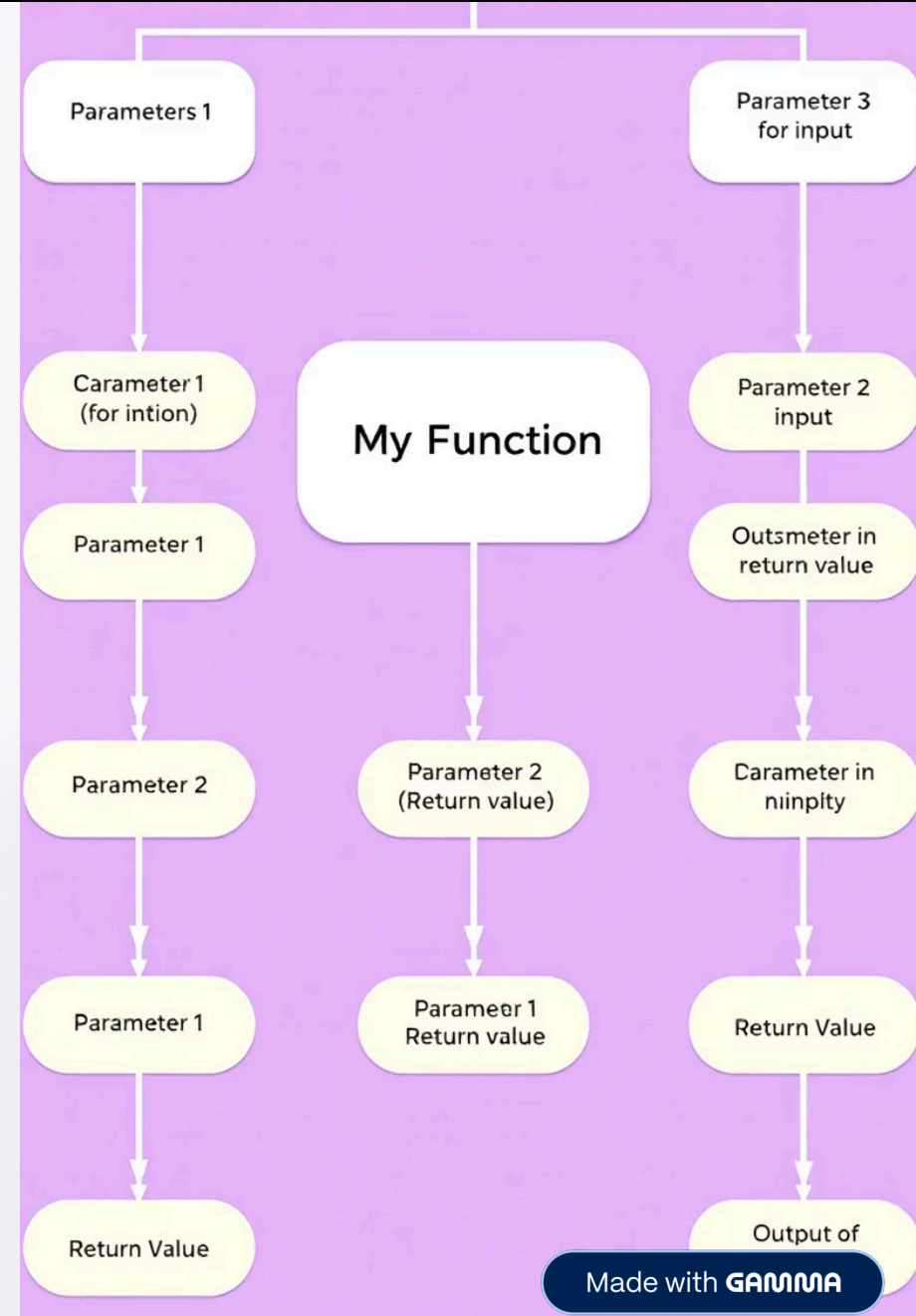
Operaciones que realiza la función con los parámetros.



Salida (Return)

Valor que devuelve la función al finalizar.

```
def cuadrado(x):  
    return x * x
```



Parámetros Opcionales

Valores por Defecto

Permiten omitir argumentos al llamar a la función.

```
def  
saludar(nombre="amigo"):  
    print(f"Hola, {nombre}")
```

Ejemplos de Uso

- saludar() → "Hola, amigo"
- saludar("Laura") → "Hola, Laura"

Ventajas

- Mayor flexibilidad
- Menos código repetitivo
- Interfaces más amigables

```
{  
defuncion paramtil: are feast to car (argget fow))  
puython  
    function {  
        optional, = tupit ); f  
        sactiual = b eet);  
  
putull;  
defaults,= (eafllit, fcs: funcitil na, Lasien de f uadit fawt fawt)  
    fraol  
    defins: (pttion) exampl);  
  
defuil:  
    instuntion ();  
    defaults fon = is default (b b) f  
    } )  
    debil)  
  
};  
  
}}
```

Retorno Múltiple con Tuplas

Definir Función

```
def operaciones(a, b):  
    suma = a + b  
    resta = a - b  
    return suma, resta
```

Desempaquetado

```
s, r = operaciones(10, 4)  
print(f'Suma: {s}, Resta: {r}')
```



Empaquetado

Python empaqueta automáticamente en tupla.

Recibir Tupla

```
resultado = operaciones(10, 4)  
print(resultado) # (14, 6)
```

Buenas Prácticas y Tips



Nombres Descriptivos

Usa nombres claros para funciones y parámetros.



Documentación (Docstrings)

```
def area_circulo(radio):  
    """Calcula el área de un  
    círculo dado su radio."""  
    import math  
    return math.pi * radio ** 2
```



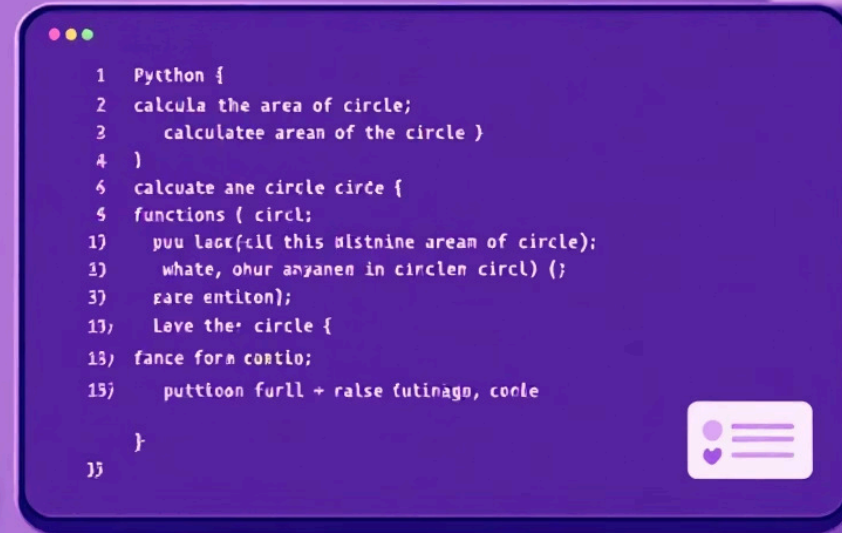
Funciones Anidadas

Definir funciones dentro de otras para encapsular mejor.



Evitar Efectos Secundarios

Una función debe hacer solo lo que promete.



Funciones Lambda



Definición

Funciones anónimas de una sola línea.



Sintaxis

lambda argumentos: expresión



Uso con filter()

```
edades = [12, 18, 25, 7, 30, 16]
mayores = list(filter(lambda e: e >= 18, edades))
print(mayores) # [18, 25, 30]
```



Uso con map()

```
nombres = ["Marti", "Laure", "Juan"]
mayusculas = list(map(lambda n: n.upper(), nombres))
```


Errores Comunes y Soluciones

Error	Ejemplo	Solución
No llamar a la función	saludar	saludar()
Olvidar el return	def suma(a,b): a+b	def suma(a,b): return a+b
Argumentos incorrectos	suma(2) cuando necesita dos	suma(2,3)
Confundir print con return	def f(x): print(x*2)	def f(x): return x*2
Ámbito incorrecto	Variables fuera de alcance	Usar parámetros adecuadamente

