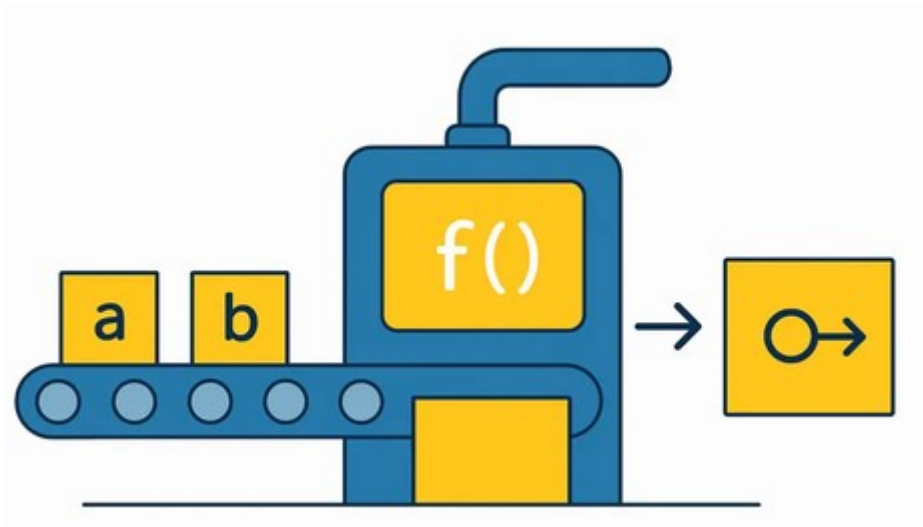


## Funciones con parámetros

Los parámetros permiten que las funciones sean más flexibles y reutilizables. Puedes pasar información a la función para que trabaje con diferentes datos.



## Parámetros posicionales

Los parámetros se pasan en el orden en que están definidos:

```
def calcular_area_rectangulo(ancho, alto):  
    """Calcula el área de un rectángulo"""  
    area = ancho * alto  
    return area
```

```
# Llamar a la función  
resultado = calcular_area_rectangulo(5, 10)  
print(resultado) # Salida: 50
```

```
# El orden importa  
resultado2 = calcular_area_rectangulo(10, 5)  
print(resultado2) # Salida: 50 (mismo resultado en este caso)
```

## Parámetros con nombre (keyword arguments)

Puedes especificar los parámetros por nombre, lo que hace el código más claro:

```
def crear_usuario(nombre, edad, ciudad):  
    return f"{nombre}, {edad} años, de {ciudad}"
```

```
# Usando parámetros posicionales
```

```

usuario1 = crear_usuario("Ana", 25, "Madrid")

# Usando parámetros con nombre (más claro)
usuario2 = crear_usuario(nombre="Carlos", edad=30, ciudad="Barcelona")

# Puedes cambiar el orden si usas nombres
usuario3 = crear_usuario(ciudad="Valencia", nombre="Laura", edad=28)

print(usuario1) # Ana, 25 años, de Madrid
print(usuario2) # Carlos, 30 años, de Barcelona
print(usuario3) # Laura, 28 años, de Valencia

```

## Parámetros con valores por defecto

Puedes definir valores predeterminados para parámetros opcionales:

```

def saludar(nombre, saludo="Hola"):
    """Saluda a una persona con un saludo personalizable"""
    return f'{saludo}, {nombre}!'

print(saludar("María")) # Hola, María!
print(saludar("Pedro", "Buenos días")) # Buenos días, Pedro!
print(saludar("Ana", saludo="Qué tal")) # Qué tal, Ana!

```

**Importante:** Los parámetros con valores por defecto deben ir después de los parámetros obligatorios:

```

# Correcto
def funcion(param_obligatorio, param_opcional="valor"):
    pass

# Incorrecto - SyntaxError
def funcion(param_opcional="valor", param_obligatorio):
    pass

```

## La sentencia return

La palabra clave return se usa para devolver un valor desde una función. Cuando Python encuentra un return, la función termina inmediatamente y devuelve el valor especificado.



## Return simple

```
def sumar(a, b):  
    return a + b
```

```
resultado = sumar(5, 3)  
print(resultado) # 8
```

## Return múltiple (tuplas)

Python permite devolver múltiples valores usando tuplas:

```
def calcular_estadisticas( numeros ):  
    """Calcula suma, promedio y máximo de una lista"""  
    total = sum( numeros )  
    promedio = total / len( numeros )  
    maximo = max( numeros )  
    return total, promedio, maximo # Devuelve una tupla
```

```
# Desempaquetado de valores  
suma, prom, max_val = calcular_estadisticas([10, 20, 30, 40, 50])  
print(f"Suma: {suma}, Promedio: {prom}, Máximo: {max_val}")  
# Salida: Suma: 150, Promedio: 30.0, Máximo: 50
```

## Ámbito de variables (scope)

El ámbito o "scope" de una variable determina desde dónde se puede acceder a ella. Python tiene dos ámbitos principales:

### Variables locales

Las variables definidas dentro de una función solo existen dentro de esa función:

```
def calcular_impuesto( precio ):  
    impuesto = precio * 0.21 # Variable local  
    return impuesto
```

```
resultado = calcular_impuesto(100)  
print(resultado) # 21.0
```

```
# Esto da error - NameError  
print(impuesto) # impuesto no existe fuera de la función
```

## Variables globales

Las variables definidas fuera de funciones son accesibles desde cualquier parte:

```
tasa_iva = 0.21 # Variable global

def calcular_precio_final(precio):
    # Podemos leer variables globales
    return precio * (1 + tasa_iva)

print(calcular_precio_final(100)) # 121.0
```

## Sistema de menú

```
def mostrar_menu():
    """Muestra las opciones del menú"""
    print("\n=== MENÚ PRINCIPAL ===")
    print("1. Opción 1")
    print("2. Opción 2")
    print("3. Salir")

def procesar_opcion(opcion):
    """Procesa la opción seleccionada"""
    if opcion == "1":
        return "Ejecutando opción 1..."
    elif opcion == "2":
        return "Ejecutando opción 2..."
    elif opcion == "3":
        return "¡Hasta luego!"
    else:
        return "Opción no válida"

# Uso
mostrar_menu()
# opcion = input("Selecciona una opción: ")
# print(procesar_opcion(opcion))
```