

E8 – Funciones en Python

Objetivo del ejercicio

Comprender y aplicar diferentes formas de definir y llamar funciones en Python, incluyendo:

- Paso de parámetros por posición
- Parámetros con valor por omisión
- Paso de parámetros por nombre
- Número indeterminado de parámetros (*args)
- Parámetros con nombres arbitrarios (**kwargs)
- Funciones lambda
- Múltiples valores de retorno

El objetivo es que el estudiante vea **cómo se comporta cada tipo de función** y cómo se usan en un programa real.

Conceptos teóricos esenciales

◆ 1. Parámetros por posición

Los argumentos se pasan en el orden en que la función los define.

```
def sumar(a, b):  
    return a + b  
sumar(3, 5)
```

◆ 2. Parámetros con valor por omisión

Si no se envía un valor, se usa el valor por defecto.

```
def saludar(nombre="invitado"):  
    print("Hola", nombre)
```

◆ 3. Parámetros por nombre

Se envían indicando explícitamente el nombre del parámetro.

```
sumar(b=10, a=3)
```

◆ 4. Número indeterminado de parámetros (*args)

Recibe cualquier cantidad de argumentos.

```
def sumar_todos(*numeros):  
    return sum(numeros)
```

◆ 5. Parámetros con nombres arbitrarios (**kwargs)

Recibe parámetros con nombre en forma de diccionario.

```
def mostrar_info(**datos):  
    print(datos)
```

◆ 6. Funciones lambda

Funciones pequeñas y anónimas.

```
cuadrado = lambda x: x * x
```

◆ 7. Múltiples valores de retorno

Una función puede regresar varios valores como tupla.

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

CÓDIGO COMPLETO

```
# 1. Parámetros por posición
def multiplicar(a, b):
    return a * b

# 2. Parámetros con valor por omisión
def saludar(nombre="invitado"):
    print(f"Hola, {nombre}!")

# 3. Parámetros por nombre
def resta(a, b):
    return a - b

# 4. Número indeterminado de parámetros (*args)
def sumar_todos(*numeros):
    print("Recibí estos números:", numeros)
    return sum(numeros)

# 5. Parámetros con nombres arbitrarios (**kwargs)
def mostrar_info(**datos):
    print("Información recibida:")
    for clave, valor in datos.items():
        print(f" {clave}: {valor}")

# 6. Función lambda
cuadrado = lambda x: x * x

# 7. Función con múltiples valores de retorno
def estadisticas(a, b, c):
    suma = a + b + c
    promedio = suma / 3
    mayor = max(a, b, c)
    return suma, promedio, mayor

# =====

print("\n1) Parámetros por posición:")
print("multiplicar(4, 5) =", multiplicar(4, 5))
print("multiplicar(7, 3) =", multiplicar(7, 3))

print("\n2) Parámetros con valor por omisión:")
saludar()
saludar("Alumno")

print("\n3) Parámetros por nombre:")
print("resta(a=10, b=3) =", resta(a=10, b=3))
print("resta(b=20, a=5) =", resta(b=20, a=5))

print("\n4) Número indeterminado de parámetros (*args):")
print("sumar_todos(1, 2, 3) =", sumar_todos(1, 2, 3))
print("sumar_todos(10, 20, 30, 40) =", sumar_todos(10, 20, 30, 40))

print("\n5) Parámetros con nombres arbitrarios (**kwargs):")
mostrar_info(nombre="Ana", edad=20, carrera="Física")
mostrar_info(ciudad="Monterrey", clima="Soleado")

print("\n6) Función lambda:")
print("cuadrado(6) =", cuadrado(6))
print("cuadrado(10) =", cuadrado(10))

print("\n7) Múltiples valores de retorno:")
suma, promedio, mayor = estadisticas(10, 20, 30)
```

```
print("Resultados:", suma, promedio, mayor)
s2, p2, m2 = estadisticas(5, 15, 25)
print("Resultados:", s2, p2, m2)
```

¿Qué hace este programa?

Este programa demuestra **todas las formas importantes de trabajar con funciones en Python**:

- Llama funciones enviando parámetros por posición y por nombre.
- Usa funciones con valores por omisión para mostrar cómo funcionan los parámetros opcionales.
- Utiliza *args para recibir una cantidad variable de números.
- Utiliza **kwargs para recibir datos con nombre y procesarlos dinámicamente.
- Declara una función lambda para cálculos rápidos.
- Muestra cómo una función puede regresar varios valores al mismo tiempo.
- Llama cada función **al menos dos veces** con parámetros distintos para observar su comportamiento.

El objetivo es que el estudiante entienda **cómo se construyen y cómo se usan** funciones de diferentes tipos.

Entregable

Sube **dos fotografías**:

1. **Foto del código completo ejecutándose**, mostrando todas las salidas.
2. **Foto del código fuente**, donde se vean claramente todas las funciones definidas.