

## Argumento:

Un argumento es un valor que se pasa a una función o método cuando este se llama, es decir, para llevar a cabo la acción específica. Los argumentos permiten a las funciones ser flexibles y poder ser utilizadas en varias veces, ya que los valores de los argumentos pueden ser diferentes cada vez que se llama a la función.

Hay diferentes tipos de argumentos:

1. **Argumentos posicionales:** se pasan a una función en un orden específico, basado en la posición de los parámetros definidos en la función. El número de argumentos y su orden deben coincidir con los parámetros definidos en la función.

Ejemplo 1

```
def suma(a, b):  
    return a + b  
  
resultado = suma(3, 5) # 3 se asigna a a y 5 se asigna a b  
print(resultado) # Output: 8
```

2. **Argumentos de palabra clave (*keyword arguments*):** utilizan el nombre del parámetro al que se quiere asignar el valor, lo que permite cambiar el orden de los argumentos sin afectar el resultado.

Ejemplo 2

```
def saludar(nombre, saludo):  
    return f"{saludo}, {nombre}:"  
  
mensaje = saludar(saludo="Hola", nombre="Juan")  
print(mensaje) # Output: "Hola, Juan:"
```

3. **Argumentos por defecto:** tienen un valor predeterminado. Si no se proporciona un valor para ese argumento al llamar a la función, se utilizará el valor predeterminado.

Ejemplo 3

```
def saludar(nombre, saludo="Hola"):  
    return f"{saludo}, {nombre}."  
  
mensaje = saludar("Sara")  
print(mensaje) # Output: "Hola, Sara."
```

4. **Argumentos arbitrarios (*\*args* y *\*\*kwargs*):**
  - **\*args:** Permite pasar un número variable de argumentos posicionales a una función. Los argumentos se recogen como una tupla dentro de la función.

#### Ejemplo 4

```
def promedio(*numeros):  
    return sum(numeros) / len(numeros) # se suman los valores de los argumentos y se dividen entre el  
    número de argumentos  
  
resultado = promedio(5, 10, 15) # se definen los valores de los argumentos. Se especifican tantos como se  
deseen.  
print(resultado) # Output: 10.0
```

- **\*\*kwargs**: Permite pasar un número variable de argumentos de palabra clave a una función. Los argumentos se recogen como un diccionario dentro de la función.

#### Ejemplo 5

```
def detalles(**info):  
    for key, value in info.items(): # formato elementos de un diccionario  
        print(f"{key}: {value}")  
  
detalles(nombre="Ana", edad=30, ciudad="Madrid") # los valores para cada elemento del diccionario.  
  
# Output:  
# nombre: Ana  
# edad: 30  
# ciudad: Madrid.
```