

# Parámetros y argumentos

## Ejercicio: Argumentos posicionales y nominales

Realizar lo siguiente: \* Definir una función que reciba nombre, edad y estatura como parámetros y retorne un diccionario creado con estos datos \* Llamar a la función mediante argumentos posicionales \* Llamar a la función mediante argumentos nominales

### Definir la Función:

- **Función construir\_diccionario:**
- **Parámetros:**
  - nombre: El nombre de la persona.
  - edad: La edad de la persona.
  - estatura: La estatura de la persona.
- **Crea un Diccionario:**
  - Utiliza los parámetros para construir un diccionario con las claves "nombre", "edad" y "estatura".
- **Devuelve el Diccionario:**
  - Retorna el diccionario creado.

```
def construir_diccionario(nombre, edad, estatura):  
    dicc = {"nombre": nombre,  
            "edad": edad,  
            "estatura": estatura}  
    return dicc
```

**Argumentos Posicionales:** En la llamada a construir\_diccionario, los valores se pasan en el orden en que se definen los parámetros en la función. Esto asegura que cada valor se asigne a la clave correcta en el diccionario creado.

```
# Argumentos posicionales
dic = construir_diccionario("José", 29, 1.74)
print(dic)
```

```
{'nombre': 'José', 'edad': 29, 'estatura': 1.74}
```

**Argumentos Nominales:** En la llamada a `construir_diccionario`, los valores se asignan a los parámetros especificando el nombre del parámetro. Esto permite pasar los argumentos en cualquier orden, mejorando la claridad del código y evitando errores de orden en la asignación de valores.

```
# Argumentos nominales
dic = construir_diccionario(edad = 29, estatura = 1.74, nombre = "José")
print(dic)
```

```
{'nombre': 'José', 'edad': 29, 'estatura': 1.74}
```