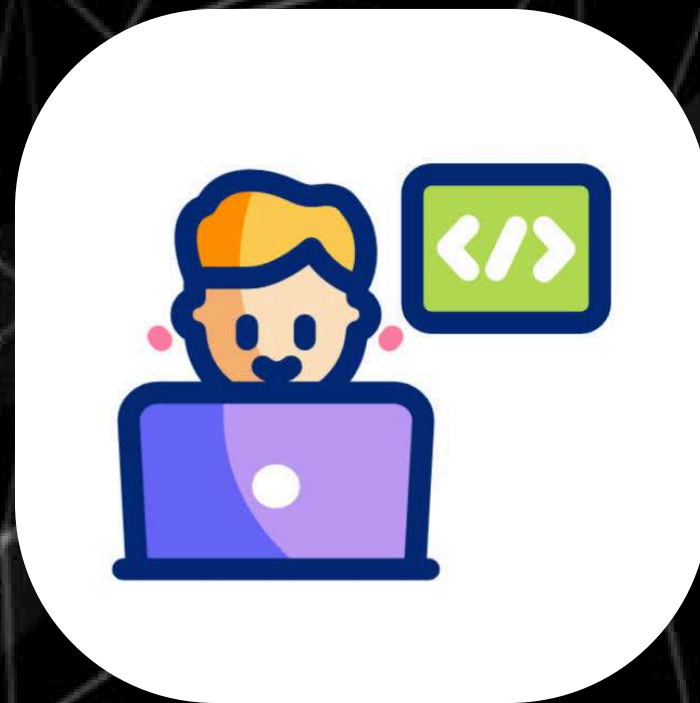


# CURSO DE PYTHON DESDE CERO



# ¿CUÁLES SON LOS PARÁMETROS Y VALORES DE RETORNO?

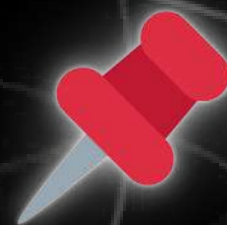
En Python, los parámetros y los valores de retorno son la base para que las funciones reciban información y devuelvan resultados.



# ✓ PARÁMETROS EN PYTHON

Los parámetros son las variables que una función recibe cuando se define.

Los argumentos son los valores reales que se envían al llamar la función.



# EJEMPLO BÁSICO

```
1  def saludar(nombre):  
2      print(f"Hola {nombre}")  
3  
4  saludar("Polo")  
5  
6  # nombre → parámetro  
7  # "Polo" → argumento
```



# ■ TIPOS DE PARÁMETROS



# PARÁMETROS OBLIGATORIOS

Debes enviarlos sí o sí al llamar la función.

```
1 def sumar(a, b):  
2     return a + b  
3  
4 sumar(3, 4)
```

# PARÁMETROS CON VALORES POR DEFECTO

Si no envías el argumento, usa el valor por defecto.

```
1 def saludar(nombre="Invitado"):  
2     return f"Hola {nombre}"  
3  
4 saludar()  
5 saludar("Luis")
```

# PARÁMETROS POR POSICIÓN

El orden importa.

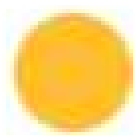


```
1 def dividir(a, b):  
2     return a / b  
3  
4 dividir(10, 2)
```



# PARÁMETROS POR NOMBRE (KEYWORD ARGUMENTS)

El orden no importa si nombras los parámetros.

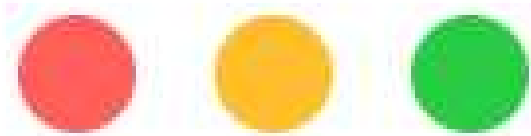


```
1  def dividir(a, b):  
2      return a / b  
3  
4  dividir(b=2, a=10)
```

# PARÁMETROS VARIABLES

## \*ARGS

Permiten recibir muchos argumentos sin saber cuántos.



```
1 def sumar(*numeros):  
2     return sum(numeros)  
3  
4 sumar(1, 2, 3, 4)
```

# PARÁMETROS VARIABLES

## **\*\*KWARGS**

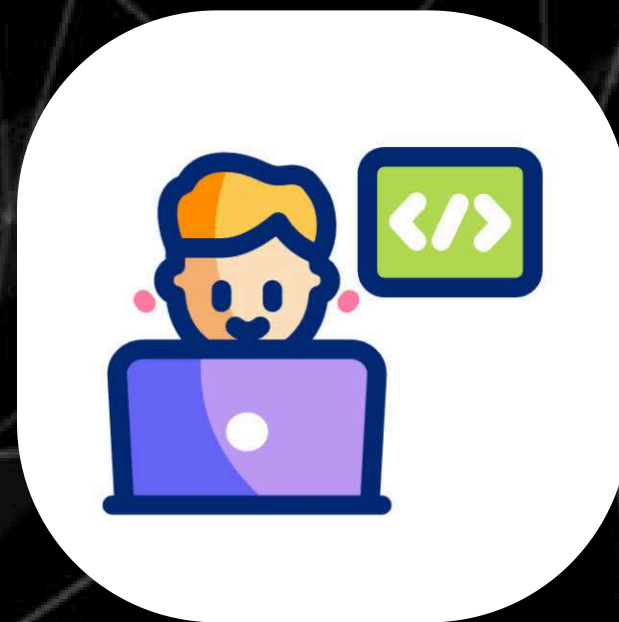
Permiten recibir argumentos con nombre en forma de diccionario.

```
1 def mostrar_info(**datos):  
2     for clave, valor in datos.items():  
3         print(clave, valor)  
4  
5 mostrar_info(nombre="Ana", edad=25)
```



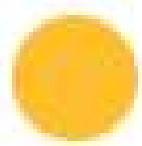
# VALORES DE RETORNO (RETURN)

El valor de retorno es el resultado que la función devuelve al finalizar.





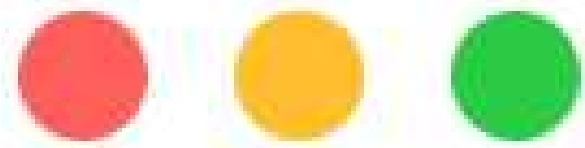
# FUNCIÓN CON RETORNO SIMPLE



```
1 def cuadrado(x):  
2     return x * x  
3  
4 resultado = cuadrado(5)
```

# FUNCIÓN SIN RETORNO

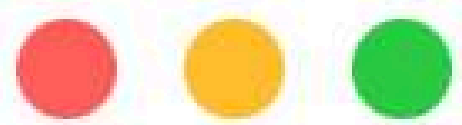
Si no usas return, la función devuelve None.



```
1  def mensaje():  
2      print("Hola")  
3  
4  print(mensaje())  # None
```

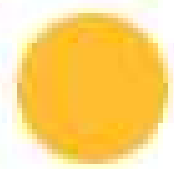
# RETORNAR MÚLTIPLES VALORES

Python devuelve una tupla.



```
1 def operaciones(a, b):  
2     return a + b, a - b, a * b  
3  
4 suma, resta, multiplicacion = operaciones(10, 5)
```

# RETORNO CONDICIONAL



```
1  def es_mayor(edad):  
2      if edad >= 18:  
3          return True  
4      return False
```





# BUENAS PRÁCTICAS

✓ Usa return para resultados, no print

✓ No mezcles demasiadas responsabilidades en una función

✓ Usa nombres claros para parámetros

✓ Documenta funciones importantes

# ■ RESUMEN RÁPIDO

- Parámetros: Datos que recibe la función
- Argumentos: Valores enviados
- \*args: Muchos parámetros por posición

# ■ RESUMEN RÁPIDO

- `**kwargs`: Muchos parámetros con nombre
- `return`: Devuelve un valor
- Retornos múltiples: Se devuelven como tupla

**NOS VEMOS EN UN PRÓXIMO  
VIDEO DE ESTE CURSO,  
SALUDOS 🚀**

