

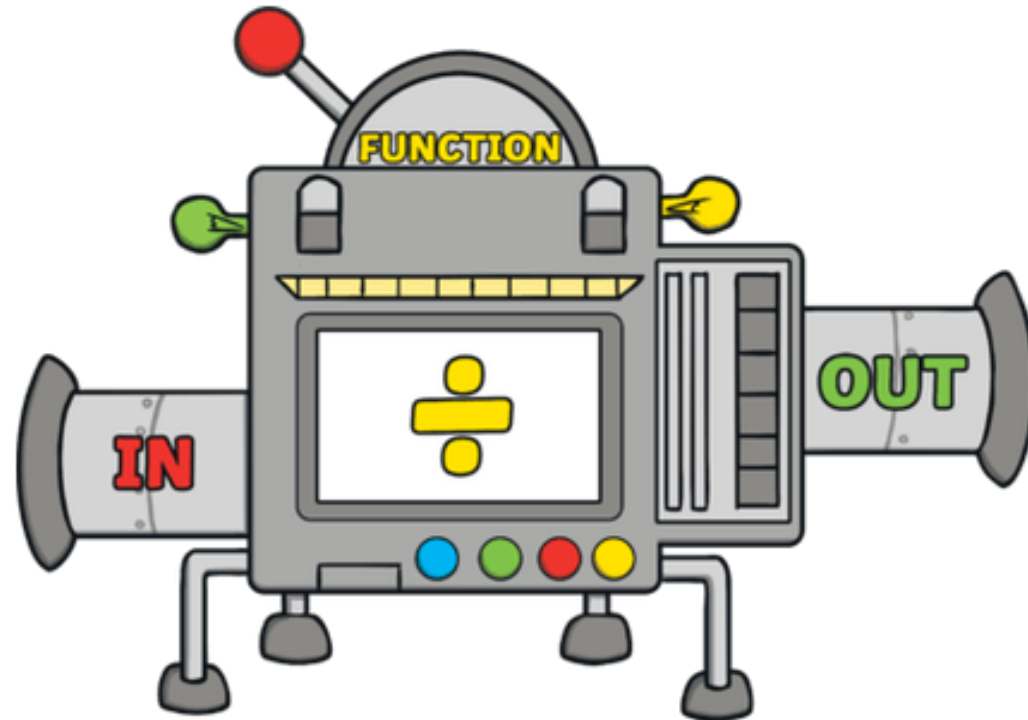
# EIN413B

## Programación



# UVA 5

## Funciones



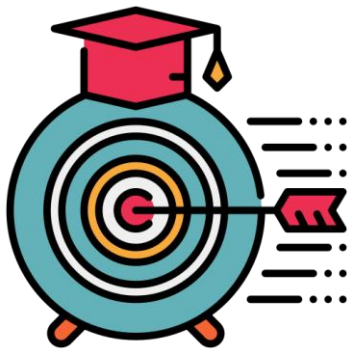
# ¿Dónde estamos?



EIN413B Programación / ELI109\_A Introducción a la Programación  
Sede Viña del Mar  
**Planificación Asignatura 2025-1**

SEMANA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
1	10-03-2025	11-03-2025	12-03-2025	13-03-2025	14-03-2025	15-03-2025	16-03-2025
2	17-03-2025	18-03-2025	19-03-2025	20-03-2025	21-03-2025	22-03-2025	23-03-2025
3	24-03-2025	25-03-2025	26-03-2025	27-03-2025	28-03-2025	29-03-2025	30-03-2025
4	31-03-2025	01-04-2025	02-04-2025	03-04-2025	04-04-2025	05-04-2025	06-04-2025
5	07-04-2025	08-04-2025	09-04-2025	10-04-2025	11-04-2025	12-04-2025	13-04-2025
6	14-04-2025	15-04-2025	16-04-2025	17-04-2025	18-04-2025	19-04-2025	20-04-2025
7	21-04-2025	22-04-2025	23-04-2025	24-04-2025	25-04-2025	26-04-2025	27-04-2025
8	28-04-2025	29-04-2025	30-04-2025	01-05-2025	02-05-2025	03-05-2025	04-05-2025
9	05-05-2025	06-05-2025	07-05-2025	08-05-2025	09-05-2025	10-05-2025	11-05-2025
10	12-05-2025	13-05-2025	14-05-2025	15-05-2025	16-05-2025	17-05-2025	18-05-2025
11	19-05-2025	20-05-2025	21-05-2025	22-05-2025	23-05-2025	24-05-2025	25-05-2025
12	26-05-2025	27-05-2025	28-05-2025	29-05-2025	30-05-2025	31-05-2025	01-06-2025
13	02-06-2025	03-06-2025	04-06-2025	05-06-2025	06-06-2025	07-06-2025	08-06-2025
14	09-06-2025	10-06-2025	11-06-2025	12-06-2025	13-06-2025	14-06-2025	15-06-2025
15	16-06-2025	17-06-2025	18-06-2025	19-06-2025	20-06-2025	21-06-2025	22-06-2025
16	23-06-2025	24-06-2025	25-06-2025	26-06-2025	27-06-2025	28-06-2025	29-06-2025
17	30-06-2025	01-07-2025	02-07-2025	03-07-2025	04-07-2025	05-07-2025	06-07-2025
18	07-07-2025	08-07-2025	09-07-2025	10-07-2025	11-07-2025	12-07-2025	13-07-2025

Semana	Tema	Control	SMOJ	Tarea
1	UVA 1: Python - Programas Secuenciales	-	-	-
2	UVA 1: Python - Programas Secuenciales	C1	UVA1	
3	UVA 2: Condicionales	C2	UVA2	TA1
4	UVA 3: Ciclos			
5	UVA 3: Ciclos	C3	UVA3	
6	UVA 4: Strings	C4	UVA4	
7	UVA 5: Funciones	C5	UVA5	TA2
8	Preparación Certamen 1			
9	Certamen 1 : Jueves 08/05 17:30 hrs.			
10	UVA 6: Listas	C6	UVA6	
11	Vacaciones Estudiantes			
12	UVA 6: Listas			TA3
13	UVA 7: Diccionarios	C7	UVA7	
14	UVA 8: Procesamiento de Texto y Archivos	C8	UVA8	TA4
15	Preparación Certamen 2			
16	Certamen 2 : Jueves 26/06 17:30 hrs.			
17	No hay más clases ni ayudantías			
18	Certamen Recuperativo : martes 08/07 17:30 hrs.			



# Objetivos

1. Implementar funciones que resuelvan subproblemas particulares dentro de un problema más amplio.
2. Comprender la diferencia entre print y return.
3. Descomponer problemas en subproblemas abordables mediante funciones.
4. Realizar seguimiento y ruteo de programas que utilizan funciones.

# ¿Qué es una Función?




Una función es un bloque de código reutilizable que se utiliza para realizar una tarea específica.

Se define utilizando la palabra clave **def** y puede recibir parámetros y retornar un valor.

```
1 def nombre_funcion(parámetros):  
2     # Cuerpo de la función  
3     return valor
```


## Parámetros

Son las entradas que recibe una función para operar. Se definen entre paréntesis después del nombre de la función.



```
1 def nombre_funcion(parámetros):  
2     # Cuerpo de la función  
3     return valor
```

## Valor de retorno



El resultado de la función se devuelve utilizando la palabra clave `return`.

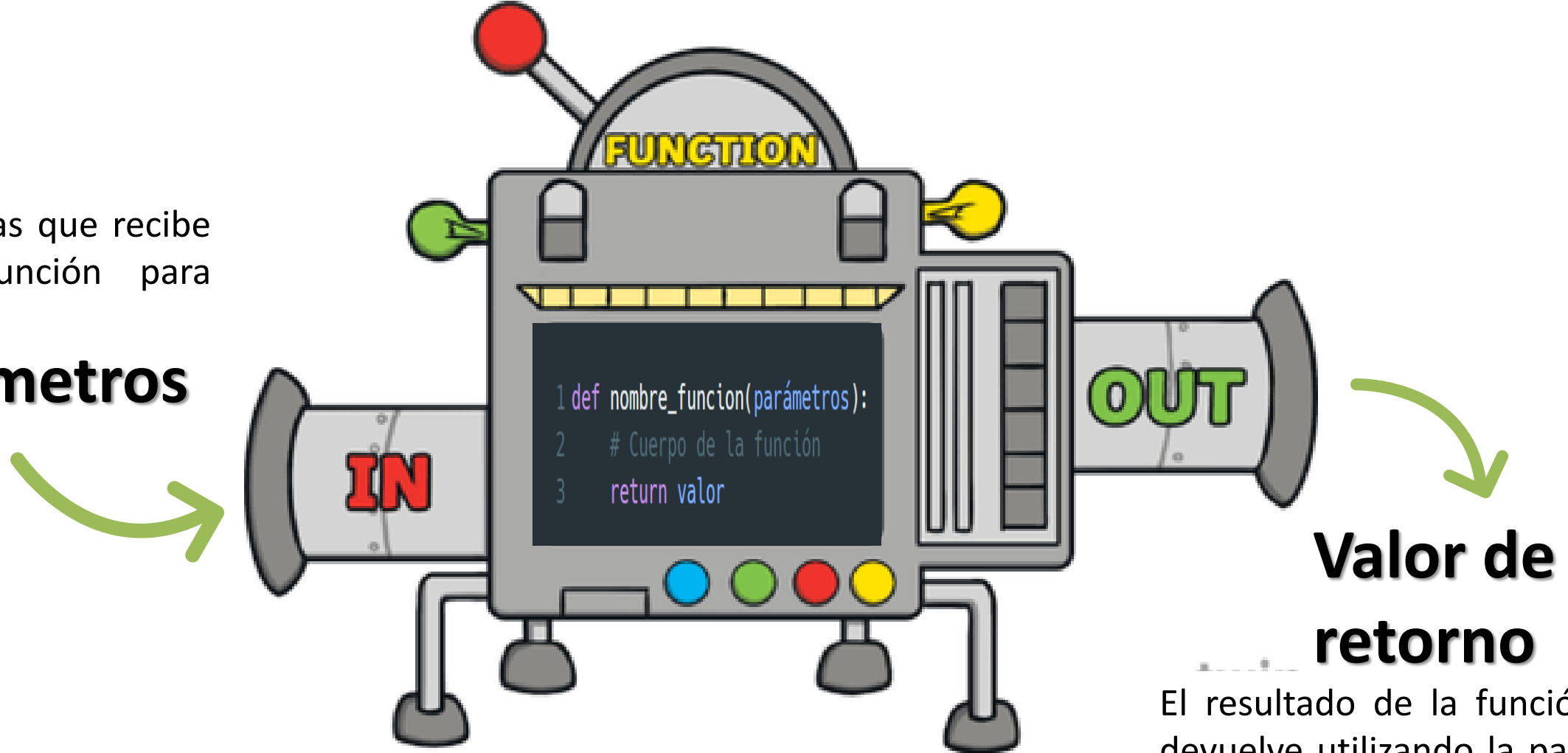
# Elementos de una Función



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

Entradas que recibe  
una función para  
operar

## Parámetros



El resultado de la función se devuelve utilizando la palabra clave **return**.

# Llamada a una Función



- Una llamada a una función es el proceso mediante el cual se ejecuta el código dentro de una función.
- Cuando se llama a una función, se envían argumentos (valores) a los parámetros definidos, y la función devuelve un valor de retorno.

```
1 def saludar(nombre):  
2     return f"Hola, {nombre}!"  
3  
4 mensaje = saludar("Bruno")  
5 print(mensaje) # Imprime: Hola, Bruno!
```



# La función principal

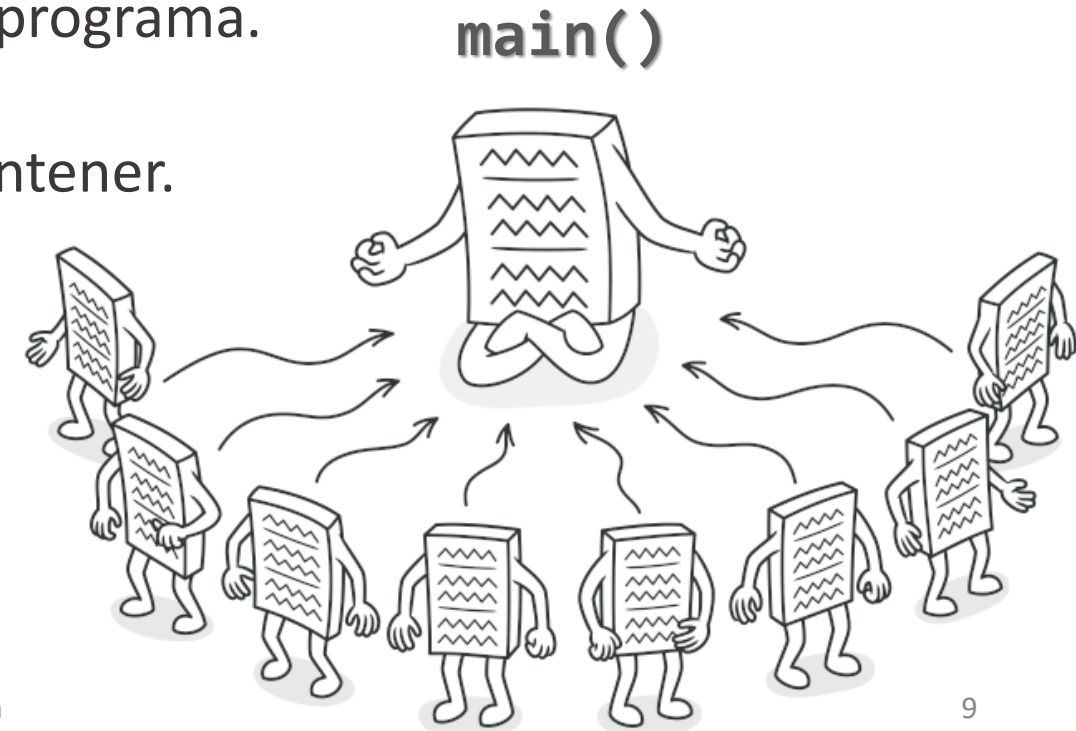


UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

La función principal es el punto de entrada desde donde comienza la ejecución del programa.

## ¿Por qué es importante?

- Estructura el flujo de ejecución.
- Organiza las llamadas a las demás funciones del programa.
- Mejora la claridad y organización del código.
- Hace que el programa sea más fácil de leer y mantener.



# Diferencia entre print() y return



**print:** Imprime un valor en la consola, pero no lo devuelve para su uso posterior en el código.

**return:** Devuelve un valor a la parte del programa que llamó a la función, permitiendo su reutilización.

```
1 def sumar_con_print(a, b):  
2     print(a + b)  
3  
4 def sumar_con_return(a, b):  
5     return a + b
```

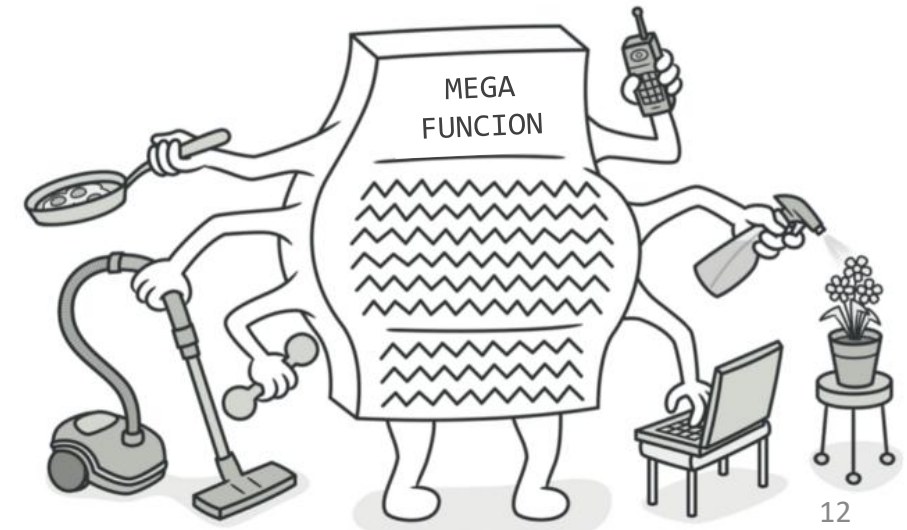
# Descomposición de Problemas



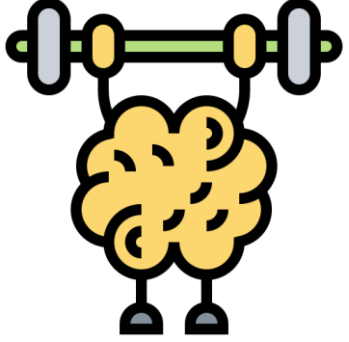
- Descomponer un problema grande en partes más pequeñas y manejables (subproblemas).
- Cada subproblema puede ser resuelto por una función, promoviendo la reutilización del código.

```
1 def area_circulo(radio):  
2     return 3.14 * radio ** 2  
3  
4 def perimetro_rectangulo(largo, ancho):  
5     return 2 * (largo + ancho)
```

- **Usar parámetros y valores de retorno:** Las funciones deben recibir toda la información necesaria a través de los parámetros.
- **Evitar variables globales:** Mantén el código modular, pasando los datos a través de los parámetros.
- No usar `print()` o `input()` dentro de funciones, a menos que la salida o entrada sea específica de la función.
- **Una función, una responsabilidad:**  
Cada función debe realizar una tarea clara y bien definida.



```
1 def multiplicar_por_dos(x):  
2     return x * 2  
3  
4 def sumar(a, b):  
5     return a + b  
6  
7 def calcular(a, b):  
8     resultado_suma = sumar(a, b)  
9     resultado_final =  
10    multiplicar_por_dos(resultado_suma)  
11    return resultado_final  
12  
13 def main():  
14     num1 = 3  
15     num2 = 4  
16     resultado = calcular(num1, num2)  
17     print(f"El resultado final es:  
18     {resultado}")  
19  
20 main()
```



# ¿Qué aprendimos?



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

1. Las funciones permiten estructurar el código de manera más clara, reutilizable y modular.
2. La descomposición de problemas es esencial para resolver problemas grandes de manera eficiente.
3. Es fundamental entender la diferencia entre print y return para implementar funciones correctas.