



# Funciones en Programación

Exploraremos el concepto de funciones en programación, su importancia y cómo utilizarlas de manera efectiva en Python. Aprenderemos a definir, llamar y estructurar funciones para mejorar la modularidad, reutilización y claridad de nuestro código.

**Ing. Alfredo Colcha O., Mgs.**



# Objetivos de la Clase

1

## Entender Funciones

Conocer qué son las funciones y cuál es su propósito en la programación.

2

## Aprender Prototipos

Comprender el concepto de prototipos de función y su importancia.

3

## Desarrollar Habilidades

Adquirir las destrezas necesarias para crear funciones en Python.

# Introducción a Funciones

## Definición

Una función es un bloque de código reutilizable que realiza una tarea específica. Tienen un propósito y beneficios claros.

## Ejemplo Sencillo

```
def saludo():  
    print("Hola,  
    mundo")  
saludo()
```

## Beneficios

Las funciones mejoran la modularidad, reutilización y legibilidad del código.

python

```
def saludo():  
    print("Hola, mundo")  
saludo()
```

# Sintaxis Básica de Funciones

1

Definición

La estructura básica incluye el nombre de la función, parámetros y el cuerpo.

2

Llamado

Para usar la función, se la invoca por su nombre y se pasan los argumentos necesarios.

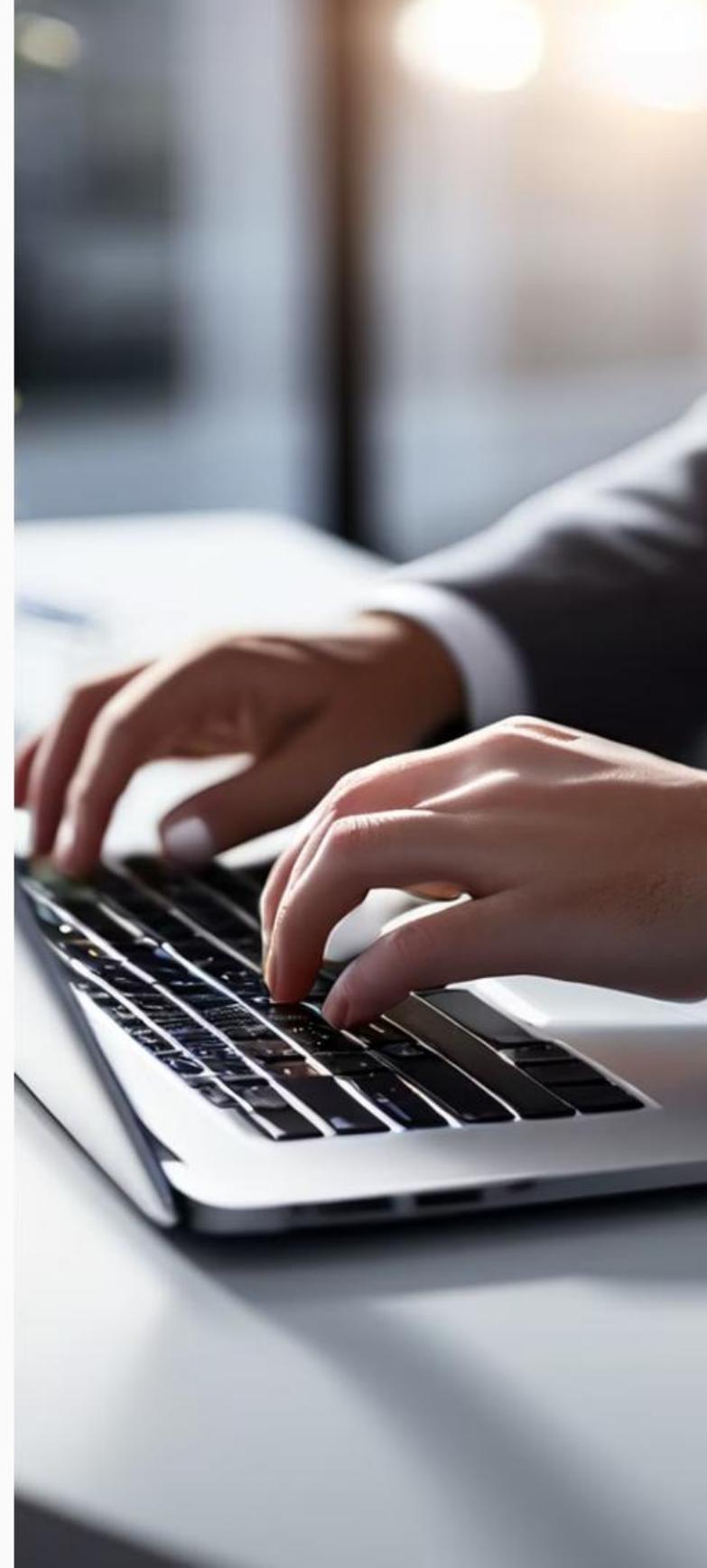
3

Retorno

Las funciones pueden devolver valores utilizando la palabra clave "return".

```
python
```

```
def nombre_funcion(parametros):  
    cuerpo_de_la_funcion  
nombre_funcion(argumentos)
```



# Ejemplo Práctico de Función

## Suma Sencilla

La función "suma" toma dos números como parámetros y devuelve su resultado.

## Código

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

## Resultado

La salida de este código sería "8".

```
python
```

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



# Prototipos de Función



## Definición

El prototipo de función define la estructura de la función, incluyendo el tipo de datos de sus parámetros y valor de retorno.



## Importancia

En lenguajes fuertemente tipados, los prototipos ayudan a verificar la consistencia del código.



## Python

En Python, no es necesario declarar explícitamente los prototipos.



# Elaboración de Funciones

1

## Identificar Tareas

Detectar patrones y acciones repetitivas en el código que puedan ser encapsuladas en funciones.

2

## Definir Parámetros

Determinar qué datos deben ser pasados a la función para lograr su objetivo.

3

## Implementar Cuerpo

Escribir el código que realizará la tarea específica de la función.

4

## Valores de Retorno

Definir los valores de retorno y su tipo de dato

# Buenas Prácticas



## Nombres Descriptivos

Utilizar nombres claros y significativos para las funciones, que reflejen su propósito.

```
python

def celsius_a_fahrenheit(celsius):
    return (celsius * 9/5) + 32
print(celsius_a_fahrenheit(25))
```



## Documentación

Agregar comentarios que expliquen el objetivo y uso de cada función.

```
python

def area_circulo(radio):
    return 3.1416 * radio ** 2
print(area_circulo(5))
```