

Módulo 1 - Unidad 2 - Lección 1

# Actividad 6

Funciones en Python



# Funciones en Python

Definición de funciones y su importancia para la modularidad del código.

Parámetros, valores de retorno y ámbito de las variables.

Las funciones en Python son bloques de código reutilizables que realizan una tarea específica, permiten dividir un programa en partes más pequeñas y modulares, lo que facilita la comprensión, mantenimiento y reutilización del código. Aquí tienen una explicación detallada sobre funciones en Python:

## Definición de Funciones:

```
def nombre_de_la_funcion(parametros):  
    # Cuerpo de la función  
    # ...  
    return resultado
```

`nombre_de_la_funcion`: es el nombre que se le da a la función. Debe seguir las reglas de nomenclatura de Python.

# Funciones en Python

**Parámetros:** son los valores que la función acepta como entrada (opcional).

**Return:** Es la palabra clave utilizada para devolver un resultado desde la función (opcional).

**Ejemplo Básico de Función:**

```
def saludar(nombre):  
    mensaje = f"Hola, {nombre}!"  
    return mensaje  
  
# Llamada a la función  
resultado_saludo = saludar("Juan")  
print(resultado_saludo)
```

En este ejemplo, la función saludar toma un parámetro (nombre), crea un mensaje de saludo y lo devuelve. Luego, la función se llama con el argumento "Juan" y el resultado se imprime.

# Funciones en Python

## Parámetros de Función:

Las funciones pueden aceptar cero o más parámetros, pueden ser parámetros posicionales (sin nombre) o parámetros con nombre.

```
def suma(a, b):  
    return a + b  
  
resultado_suma = suma(3, 5)  
print(resultado_suma) # Salida: 8
```

En este ejemplo, la función suma toma dos parámetros (a y b) y devuelve la suma de ellos.

# Funciones en Python

## Parámetros con Nombre:

```
def saludar(nombre, saludo="Hola"):  
    mensaje = f"{saludo}, {nombre}!"  
    return mensaje  
  
resultado_saludo = saludar("María", saludo="Buen día")  
print(resultado_saludo)
```

Aquí, la función saludar tiene un parámetro con nombre (saludo) con un valor predeterminado. Al llamar a la función, se puede proporcionar un valor específico para ese parámetro.

# Funciones en Python

## Return y Valores de Retorno:

Las funciones pueden devolver un valor utilizando la palabra clave return. Si no hay una declaración de return o si no se proporciona ningún valor después de return, la función devuelve None.

## Alcance de las Variables:

Las variables definidas dentro de una función son locales a esa función, a menos que se utilice la palabra clave global, esto significa que no están disponibles fuera de la función.

## Funciones Integradas:

Python tiene muchas funciones integradas que pueden utilizar sin necesidad de definir las, como print(), len(), max(), min(), entre otras.

```
longitud = len("Hola")  
print(longitud) # Salida: 4
```



***Las funciones son esenciales para organizar y estructurar código en Python. Facilitan la modularidad y la reutilización del código, contribuyendo a un diseño más claro y mantenible.***

# Ejercicios

*Saludo  
Personalizado*

*Suma de  
Cuadrados*

*Verificación de  
Número Primo*

*Conversión de Celsius  
a Fahrenheit*

*Estos ejercicios cubren diferentes aspectos del manejo de funciones, desde saludos personalizados hasta cálculos más complejos.*



## Ejercicio Conversión de Celsius a Fahrenheit

- Crea una función llamada 'celsius\_a\_fahrenheit' que toma una temperatura en grados Celsius y la convierte a Fahrenheit.
- La fórmula es:  $Fahrenheit = (Celsius * 9/5) + 32$ .
- Luego, llama a la función con una temperatura e imprime el resultado.

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



## Ejercicio Verificación de Número Primo

- Define una función llamada 'es\_primo' que toma un número como parámetro y devuelve True si es primo, False si no.
- Luego, llama a la función con un número e imprime el resultado.

```
def es_primo(numero):  
    if numero < 2:  
        return False  
    for i in range(2, int(numero**0.5) + 1):  
        if numero % i == 0:  
            return False  
    return True  
  
numero_primo = 17  
print(es_primo(numero_primo))
```



## Ejercicio Suma de Cuadrados

- Crea una función llamada 'suma\_cuadrados' que toma dos números como parámetros y devuelve la suma de sus cuadrados.
- Luego, llama a la función con dos números e imprime el resultado.

```
def suma_cuadrados(a, b):  
    return a**2 + b**2  
  
numero1 = 3  
numero2 = 4  
print(suma_cuadrados(numero1, numero2))
```



## Ejercicio Saludo Personalizado

- Define una función llamada 'saludar' que toma un nombre como parámetro y devuelve un saludo personalizado.
- Luego, llama a la función con tu nombre e imprime el resultado.

```
def saludar(nombre):  
    return f"Hola, {nombre}!"  
  
nombre_usuario = "Juan"  
print(saludar(nombre_usuario))
```



**TIC**

▶ **TALENTO**  
**TECH**

**AZ** | **PROYECTOS**  
**EDUCATIVOS**

