



## Clase 9 – Funciones simples en Python

### Fundamentación

Introducir funciones es un paso clave para que los estudiantes organicen su código, reduzcan repeticiones y empiecen a pensar en módulos reutilizables. Conocer la estructura básica de def y cómo llamarla será la base para proyectos más complejos.

### Objetivo General

Comprender el concepto de función y su utilidad en la programación.

### Objetivos Particulares

- Definir funciones simples sin parámetros.
- Ejecutar funciones desde el programa principal.
- Identificar la diferencia entre declarar y llamar una función.

### Contenidos

- **Conceptuales:**
  - Definición de función.
  - Sintaxis de def.
  - Llamado a una función.
- **Procedimentales:**
  - Crear y ejecutar una función que imprima un saludo.
  - Crear una función que dibuje un cuadrado con Turtle.
- **Actitudinales:**
  - Valorar la organización y limpieza del código.
  - Seguir buenas prácticas de indentación.



## PYTHON INICIAL

---

### Definir funciones en Python

Las funciones se pueden crear en cualquier punto de un programa, escribiendo su definición.

La primera línea de la definición de una función contiene:

- la palabra reservada `def`
- el nombre de la función (la guía de estilo de Python recomienda escribir todos los caracteres en minúsculas separando las palabras por guiones bajos)
- paréntesis (que pueden incluir los argumentos de la función, como se explica más adelante)

Las instrucciones que forman la función se escriben con sangría con respecto a la primera línea.

Por comodidad, se puede indicar el final de la función con la palabra reservada **`return`** (más adelante se explica el uso de esta palabra reservada), aunque no es obligatorio.

Para poder utilizar una función en un programa se tiene que haber definido antes. Por ello, normalmente las definiciones de las funciones se suelen escribir al principio de los programas

```
def nombre_de_la_funcion():  
    # Instrucciones que ejecuta la función
```

#### Ejemplo 1 – Función saludo

```
def saludo():  
    | print("Hola, bienvenido a la clase de Python.")  
    | print("Hoy aprenderemos funciones.")  
  
# Llamado a la función  
saludo()
```

#### Puntos a explicar:

- `def saludo()`: crea la función.
- Las instrucciones dentro están indentadas.
- `saludo()` ejecuta lo que está adentro.



## PYTHON INICIAL

---

### Ejemplo 2 – Reutilización

```
def saludo():  
    print("¡Hola!")  
    print("Espero que estés listo para aprender.")  
  
print("Llamada 1:")  
saludo()  
  
print("Llamada 2:")  
saludo()
```

Muestra que la misma función puede usarse varias veces sin reescribir las instrucciones.

### Ejemplo 3 – Función que dibuja con Turtle

```
import turtle  
  
def cuadrado():  
    for i in range(4):  
        turtle.forward(100)  
        turtle.right(90)  
  
cuadrado()  
turtle.done()
```

#### Puntos a explicar:

- La función `cuadrado()` dibuja un cuadrado.
- Se puede llamar varias veces para hacer varios cuadrados.
- Cambiar el valor 100 cambia el tamaño.



## PYTHON INICIAL

---

### Ejemplo 4 – Varias funciones en el mismo programa

```
import turtle

def cuadrado():
    for i in range(4):
        turtle.forward(100)
        turtle.right(90)

def triangulo():
    for i in range(3):
        turtle.forward(100)
        turtle.right(120)

cuadrado()
triangulo()
turtle.done()
```

Aquí ven que podemos tener más de una función y decidir cuál ejecutar.

### Llamando una función

Una vez que se ha definido una función, podemos llamarla desde cualquier parte del código. Al llamar una función, se ejecuta el bloque de código dentro de la función y se puede devolver un resultado si es necesario.

En Python, para llamar una función, se utiliza el nombre de la función seguido de paréntesis. Por ejemplo, si tenemos una función llamada "suma", podríamos llamarla de la siguiente manera: saludo()

### Actividad práctica guiada

- **Objetivo:** crear dos funciones diferentes, una de texto y una con Turtle.
- **Instrucciones:**
  1. Crear una función presentacion() que imprima el nombre del alumno y su color favorito.
  2. Crear una función estrella() que dibuje una estrella de 5 puntas con Turtle.
  3. Llamar a ambas funciones desde el programa principal.