



Universidad Tecnológica Equinoccial  
Tecnología Superior en Desarrollo de  
Software

Programación I



Nombre: Mikaela Zurita

Paralelo: "B"

## Estructuras de control: Secuencia: Ejecución lineal de instrucciones.

### 1. ¿Qué es la Secuencia?

La secuencia en programación es la estructura más simple. Las instrucciones se ejecutan una tras otra, de arriba hacia abajo, sin tomar decisiones ni repeticiones. En Python, esto es lo que ocurre por defecto: cuando escribes un conjunto de instrucciones, las ejecuta de forma secuencial, sin necesidad de estructuras adicionales.

En esta función **nombre** es el primer parámetro y **edad** es el segundo. Al llamar a la función el primer valor "**pedro**" se asigna a nombre y el segundo valor **24** se asigna a la edad.

### 2. Características de la Secuencia en Python:

- Flujo lineal: Las instrucciones se ejecutan en el mismo orden en que están escritas.
- No hay saltos ni decisiones: A diferencia de otras estructuras como if o for, no hay condiciones ni bifurcaciones.
- Ejecución única: Cada instrucción se ejecuta una sola vez, a menos que sea llamada dentro de un bucle o función.

### 3. Importancia de la secuencia en Python.

- **Base de todo programa:** Toda ejecución en Python comienza con una secuencia. Es el primer paso antes de que se añadan estructuras más complejas como bucles o condicionales.
- **Claridad:** El flujo secuencial facilita la lectura y comprensión del código, especialmente para programas pequeños o con tareas simples.

- **Facilidad de uso:** La secuencia es suficiente para resolver problemas simples sin la necesidad de estructuras complejas. Aquí los parámetros tienen un valor predeterminado. Es decir, si no se pasa un valor al llamarla, la función usará el valor que fue definido por defecto.

#### 4. Relación con otras estructuras de control en Python.

La secuencia se combina con otras estructuras de control como las condicionales y los bucles para crear programas más complejos. Sin embargo, siempre comienza con secuencia. Cuando usamos if, for, o while, estamos cambiando el flujo de la secuencia para tomar decisiones o repetir acciones.

#### 5. Estructura básica.

En Python, la estructura secuencial se ve en programas sencillos sin ninguna condición ni repetición:

```

1  # Paso 1: Declarar las variables
2  numero1 = 5
3  numero2 = 10
4
5  # Paso 2: Realizar una operacion
6  suma = numero1 + numero2
7
8  # Paso 3: Imprimir el resultado
9  print("La suma es:", suma)
10

```

#### - Flujo de ejecución de manera secuencial

- Se asignan valores a las variables numero1 y numero2. (**asignación de valores**)
- Se suma el valor de numero1 y numero2, almacenando el resultado en la variable suma. (**operación suma**)
- Finalmente, se imprime el valor de suma. (**imprime el resultado**)

#### 6. Ejemplo: Promedio tres números

```

D:\PROGRAMACION > PROMEDIO TRES NUMEROS (SECUENCIA).py > ...
1
2  numero1 = float(input("Ingresa el primer numero: "))
3  numero2 = float(input("Ingresa el segundo numero: "))
4  numero3 = float(input("Ingresa el tercer numero: "))
5
6  promedio = (numero1 + numero2 + numero3) / 3
7
8
9  print("El promedio es:", promedio)
10

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\PROGRAMACION\ProyectoProgramacion> & C:/Users/Det-Pc/AppData/Local/P
Ingresa el primer numero: 8
Ingresa el segundo numero: 9
Ingresa el tercer numero: 10
El promedio es: 9.0
PS D:\PROGRAMACION\ProyectoProgramacion>

```

- **Flujo secuencial:**
  - Se solicitan tres números al usuario.
  - Se calcula el promedio de esos tres números.
  - Se imprime el resultado del promedio.

Este programa se ejecuta de forma secuencial porque no hay condicionales ni bucles. Cada instrucción sigue a la anterior sin desvíos.

## 7. Secuencia en operaciones matemáticas

Ejemplo: Calcular el área de un triángulo

```
D: > PROGRAMACION > SECUENCIA OPERACIONES MATEMATICAS.py > ...
1
2 base = float(input("Ingresa la base del rectangulo: "))
3 altura = float(input("Ingresa la altura del rectangulo: "))
4 area = base * altura
5 print(f"El area del rectangulo es: {area}")
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\PROGRAMACION\ProyectoProgramacion> & C:/Users/Det-Pc/AppData/Local/Programs/Python/Python38-64/Python.exe D:\PROGRAMACION\ProyectoProgramacion\SECUENCIA OPERACIONES MATEMATICAS.py
Ingresa la base del rectangulo: 4
Ingresa la altura del rectangulo: 2
El area del rectangulo es: 8.0
PS D:\PROGRAMACION\ProyectoProgramacion>
```

### Flujo del Programa:

- Pide la base del rectángulo.
- Pide la altura del rectángulo.
- Calcula el área multiplicando base por altura.
- Muestra el resultado.

## 8. Ejemplo de uso de secuencia con condiciones simples (no alteran la secuencia)

Aunque las estructuras como **if** o **while** cambian el flujo, a veces pueden ser parte de una secuencia para realizar algo bajo condiciones específicas. Así se debería usar una condición simple junto a una secuencia:

```
1 edad = 21
2 edad = int(input("Ingresa tu edad: "))
3
4
5 if edad >= 18:
6     print("Eres mayor de edad.")
7 else:
8     print("Eres menor de edad.")
9

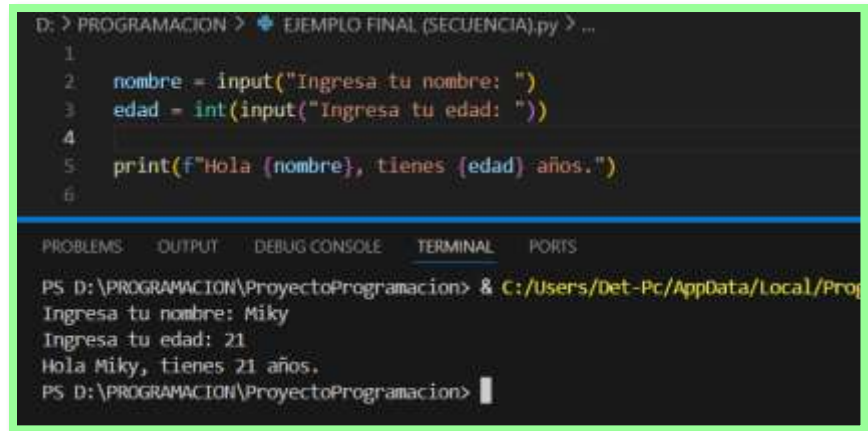
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\PROGRAMACION\ProyectoProgramacion> & C:/Users/Det-Pc/AppData/Local/Programs/Python/Python38-64/Python.exe D:\PROGRAMACION\ProyectoProgramacion\SECUENCIA OPERACIONES MATEMATICAS.py
Ingresa tu edad: 21
Eres mayor de edad.
PS D:\PROGRAMACION\ProyectoProgramacion>
```

El flujo sigue siendo secuencial, pero la condición de **if** simplemente cambia qué mensaje se imprime. En este ejemplo, el flujo sigue siendo secuencial, pero con un desvío para mostrar un mensaje diferente dependiendo de la condición.

## 9. Ejemplo final

Ejemplo donde se usan solo secuencias, ni bucles ni condiciones complejas:



```
D: > PROGRAMACION > EJEMPLO FINAL (SECUENCIA).py > ...
1
2  nombre = input("Ingresa tu nombre: ")
3  edad = int(input("Ingresa tu edad: "))
4
5  print(f"Hola {nombre}, tienes {edad} años.")
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\PROGRAMACION\ProyectoProgramacion> & C:/Users/Det-Pc/AppData/Local/Programs/Python/Python39-64/Python.exe EJEMPLO FINAL (SECUENCIA).py
Ingresa tu nombre: Miky
Ingresa tu edad: 21
Hola Miky, tienes 21 años.
PS D:\PROGRAMACION\ProyectoProgramacion>
```

Este programa sigue un flujo secuencial donde:

- Se recibe el nombre y la edad.
- Se imprime un mensaje con esos datos.
- No hay saltos, no hay condiciones. Todo sigue el orden en el que se escribió.

## 10. Conclusión.

La secuencia es un pilar importante de cualquier programa. En Python, este tipo de control se utiliza de manera predeterminada, permitiendo que las instrucciones se ejecuten en un orden específico. Aunque las estructuras más complejas agregan poder y flexibilidad al código, la secuencia siempre es la base sobre la que se construyen



