

Fundamentos de Programación Moderna

Variables, control de flujo y lógica 📊


Probabilidad y Estadística

Aprendiendo con diversión 🎓

25 de noviembre de 2025

¿Qué vamos a aprender?

Nota importante

¡Hoy empezamos a programar de verdad! 

Objetivo 1: Entender qué son las variables y tipos de datos

Objetivo 2: Escribir expresiones y sentencias

Objetivo 3: Controlar el flujo del programa con decisiones


Objetivo 4: Usar ciclos para repetir tareas

Ejemplo

Como organizar cajas con diferentes objetos y decidir qué hacer con cada una

Variables: Las Cajas de Memoria

Nota importante

Una variable es como una caja etiquetada donde guardamos datos 

Cada variable tiene un nombre (etiqueta)

Cada variable almacena un valor

Ejemplo

```
nombre_empresa = "Mi Empresa S.A."  
saldo = 1500000  
activo = True
```

✓ La computadora reserva memoria para guardar estos valores

Tipos de Datos Primitivos

Tipo	Uso	Ejemplo
Enteros (int) ↑↓	IDs, cantidades	<code>id_usuario = 12345</code>
Flotantes (float) \$	Dinero, precios	<code>precio = 19.99</code>
Textos (string) 📄	Nombres, emails	<code>nombre = "Juan"</code>
Booleanos (bool) ✓	Estados sí/no	<code>activo = True</code>

💡 Nota importante

Cada tipo de dato se guarda diferente en memoria

Operadores Aritméticos

Permiten hacer cálculos matemáticos

Operador	Operación	Ejemplo
+	Suma	$10 + 5 = 15$
-	Resta	$20 - 8 = 12$
*	Multiplicación	$6 * 7 = 42$
/	División	$15 / 3 = 5$

Ejemplo

$\text{total} = \text{precio} * \text{cantidad} + \text{iva}$

Operadores Lógicos

Permiten comparar y combinar condiciones

Operador	Significado	Ejemplo
==	Igual a	edad == 18
!=	Diferente de	saldo != 0
>	Mayor que	precio > 100
>=	Mayor o igual	edad >= 18
<	Menor que	stock < 10
<=	Menor o igual	nota <= 7

Ejemplo

Si saldo >= precio entonces aprobar compra

Expresiones vs. Sentencias

💡 Nota importante

¿Cuál es la diferencia? ❓

Expresión	Sentencia
Tiene un valor $5 + 3$	Ejecuta una acción $x = 5 + 3$
<code>precio * 1.19</code> <code>edad <= 18</code>	<code>print(mensaje)</code> <code>if edad <= 18:</code>

✏️ Ejemplo

Expresión: `precio * 1.19` (calcula IVA) → Sentencia: `total = precio * 1.19` (asigna resultado)

Ejemplo Práctico: Cálculo de IVA

🎯 Problema

Calcular el precio total de un producto con IVA (19

$$precio_{neto} = 100000$$

$$iva = precio_{neto} * 0,19 = 19000$$

$$precio_{total} = precio_{neto} + iva = 119000$$

✎ Ejemplo

Control de Flujo: IF/ELSE

💡 Nota importante

¡La computadora puede tomar decisiones! 🤖

IF = Si se cumple una condición, ejecuta un bloque de código

ELSE = Si no, ejecuta otro bloque

✂ Ejemplo

```
if saldo >= costo:
    aprobar_transaccion()
    restar_saldo(costo)
else:
    rechazar_transaccion()
    mostrar_error("Saldo insuficiente")
```

Ejemplo: Validación de Transacción

🎯 Problema

Aprobar o rechazar una compra según el saldo disponible

✏️ Ejemplo

```
saldo = 50000
precio_producto = 30000

if saldo >= precio_producto:
    print("Compra aprobada")
    saldo = saldo - precio_producto
    print("Nuevo saldo:", saldo)
else:
    print("Saldo insuficiente")
```

Resultado : Compra aprobada Nuevo saldo : 20000

Operadores Lógicos: AND, OR, NOT

Operador	Uso	Ejemplo
AND	Ambas condiciones verdaderas	$edad_i = 18$ AND $tiene_l$ licencia
OR	Al menos una verdadera	$pago_e$ efectivo OR $pago_t$ arjeta
NOT	Niega la condición	NOT $usuario_b$ bloqueado

Ejemplo

```
if saldo >= precio AND usuario_activo:  
    procesar_compra()
```

Nota importante

Las validaciones de seguridad usan muchos AND para verificar múltiples condiciones

Estructuras SWITCH/MATCH

Para elegir entre múltiples opciones

Ejemplo

Menú de opciones en un ERP:

```
match opcion: case 1: crear_factura() case2 : consultar_saldo() case3 :  
generar_reporte() case: print(" Opcion invalida")
```

Nota importante

Más limpio que múltiples IF/ELIF cuando hay muchas opciones

Ciclos: WHILE (Mientras)

💡 Nota importante

¡Repetir código mientras se cumpla una condición! ↺

Se usa cuando no sabemos cuántas veces repetir

🔧 Ejemplo

```
intentos = 0
conectado = False

while not conectado and intentos < 5:
    conectado = intentar_conexion()
    intentos = intentos + 1

if conectado:
    print("Conexion exitosa")
else:
    print("Error de conexion")
```

Ciclos: FOR (Para Cada)

Se usa cuando sabemos cuántas veces repetir

Ideal para recorrer listas de datos

Ejemplo

Pagar nómina a todos los empleados:

```
for empleado in empleados: procesar_pago(empleado) enviar_comprobante(empleado) print(" Pag
```

Ejemplo Práctico: Calcular Total de Ventas

🎯 Problema

Sumar todas las ventas del día

✏️ Ejemplo

```
ventas = [15000, 23000, 8500, 12000, 31000]
total = 0

for venta in ventas:
    total = total + venta

print("Total del día:", total)
```

Resultado : Total del día : 89500

Prevención de Bucles Infinitos



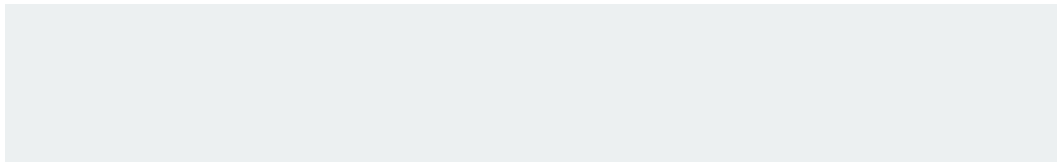
Nota importante

⚠ ¡Cuidado con los bucles que nunca terminan!



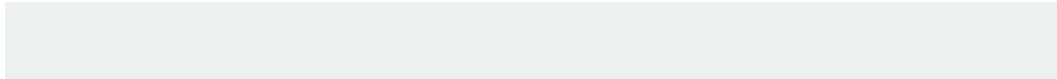
Ejemplo

Bucle infinito (ERROR):



Ejemplo

Bucle correcto:



¡Tu Turno de Practicar!

Problema

Crear un programa que valide un descuento especial

Condiciones:

- Si la compra es mayor a 100,000B10
- Si además es cliente VIP → 15
- Calcular el precio final


Nota importante

Usa IF/ELSE y operadores AND para resolverlo

¡Resumen de lo Aprendido!

- ✓ Variables guardan datos en memoria (como cajas etiquetadas)
- ✓ Tipos de datos: Enteros, Flotantes, Textos, Booleanos
- ✓ Operadores aritméticos para cálculos (+, -, *, /,
- ✓ Operadores lógicos para comparaciones (==, !=, <, >, <=, >=)
- ✓ IF/ELSE permite tomar decisiones en el código
- ✓ WHILE repite mientras se cumpla una condición
- ✓ FOR recorre listas y repite un número fijo de veces

Nota importante

¡Con estos fundamentos ya puedes crear programas reales! 

¡Sigue aprendiendo! 🚀