

01 - Variables y Tipos de Datos

Mostrar información con print()

Antes de trabajar con variables, necesitamos saber cómo mostrar información en pantalla. En Python usamos la función `print()`:

```
print("Hola, mundo!")
print("Mi primer programa en Python")
print(42)
print(3.14)
```

Características de print(): - Muestra texto o números en la pantalla - Cada `print()` aparece en una línea nueva
- El texto va entre comillas (" o ') - Los números van sin comillas

```
# Mostrar diferentes tipos de datos
print("Esto es texto")
print(123)
print(45.67)
print(True)

# Mostrar varias cosas en un solo print()
print("El resultado es:", 10 + 5)
print("Tu edad es", 25, "años")
```

Diferentes formas de imprimir

1. Print básico

```
print("Hola")
print(42)
```

2. Print con múltiples valores

```
nombre = "Ana"
edad = 25
print("Nombre:", nombre, "Edad:", edad)
# Salida: Nombre: Ana Edad: 25
```

3. F-strings (forma moderna y recomendada)

```
nombre = "Ana"
edad = 25
print(f"Mi nombre es {nombre} y tengo {edad} años")
# Salida: Mi nombre es Ana y tengo 25 años
```

```
precio = 19.99
print(f"El precio es {precio:.2f} euros")
# Salida: El precio es 19.99 euros
```

4. Método .format()

```

nombre = "Ana"
edad = 25
print("Mi nombre es {} y tengo {} años".format(nombre, edad))
# Salida: Mi nombre es Ana y tengo 25 años

```

5. Concatenación de strings

```

nombre = "Ana"
edad = "25" # Debe ser string para concatenar
print("Mi nombre es " + nombre + " y tengo " + edad + " años")
# Salida: Mi nombre es Ana y tengo 25 años

```

6. Print con separadores personalizados

```

print("rojo", "verde", "azul", sep="-")
# Salida: rojo-verde-azul

print("Línea 1", end=" ")
print("Línea 2")
# Salida: Línea 1 Línea 2 (en la misma línea)

```

Recomendación: Usa **f-strings** siempre que puedas, es la forma más clara y eficiente.

¿Qué es una variable?

Una variable es como una **caja** donde guardamos información para usarla después. En Python no hace falta declararlas, solo asignar un valor.

Tipos de datos básicos

1. Strings (texto)

Texto entre comillas simples o dobles:

```

nombre = "Ana"
ciudad = 'Madrid'
mensaje = "Hola, mundo!"

```

2. Integers (números enteros)

Números sin decimales:

```

edad = 25
puntos = 100
temperatura = -5

```

3. Floats (números decimales)

Números con decimales:

```

altura = 1.75
precio = 19.99
pi = 3.14159

```

4. Booleans (verdadero/falso)

Solo dos valores posibles: True o False

```

es_estudiante = True
tiene_carnet = False
esta_lloviendo = False

```

Ejemplos básicos

```
# Crear variables
nombre = "Carlos"
edad = 30
altura = 1.80
es_programador = True

# Usar variables
print(nombre)
print(edad)

# Cambiar el valor de una variable
edad = 31
print(edad)

# Usar variables en operaciones
precio = 10
cantidad = 3
total = precio * cantidad
print(total) # 30
```

Ver el tipo de una variable con type()

Usa type() para saber qué tipo de dato es una variable:

```
nombre = "Ana"
edad = 25
altura = 1.65
es_estudiante = True

print(type(nombre))      # <class 'str'>
print(type(edad))        # <class 'int'>
print(type(altura))      # <class 'float'>
print(type(es_estudiante)) # <class 'bool'>

# Útil para debugging
dato = "25"
print(type(dato)) # <class 'str'> - es texto, no número!
```

Pedir datos al usuario con input()

Puedes pedir al usuario que escriba datos usando input():

```
# Pedir el nombre (siempre devuelve string)
nombre = input("¿Cómo te llamas? ")
print(f"Hola, {nombre}!")

# Pedir un número (convertir de string a int)
edad_str = input("¿Cuántos años tienes? ")
edad = int(edad_str) # Convertir a entero
print(f"El próximo año tendrás {edad + 1} años")

# Forma corta (convertir directamente)
altura = float(input("¿Cuánto mides en metros? "))
print(f"Mides {altura} metros")
```

Recuerda: `input()` siempre devuelve un string. Si necesitas un número, usa `int()` o `float()`.

Reglas para nombres de variables

Permitido: - Letras (a-z, A-Z) - Números (pero no al inicio) - Guión bajo (`_`)

No permitido: - Empezar con número - Espacios - Caracteres especiales (`@`, `#`, `$`, etc.)

```
# Bien
nombre = "Ana"
edad_usuario = 25
precio_total = 100
dato1 = "valor"

# Mal
1nombre = "Ana"      # Empieza con número
edad usuario = 25     # Tiene espacio
precio-total = 100    # Guión medio no permitido
```

Buenas prácticas

- Usa nombres **descriptivos**: `edad` mejor que `e`
 - Usa **minúsculas** con guiones bajos: `nombre_completo`
 - Sé **consistente** con tu estilo
-

Ejercicios

Ejercicio 1: Tus datos personales con `input()`

Pide al usuario: - Su nombre - Su edad (convertir a `int`) - Su altura en metros (convertir a `float`) - Si es estudiante (responder “si” o “no”)

Imprime toda la información y el tipo de cada variable con `type()`.

Ejemplo de ejecución:

```
¿Cómo te llamas? Ana
¿Cuántos años tienes? 25
¿Cuánto mides en metros? 1.65
¿Eres estudiante? (si/no): si

Tus datos:
Nombre: Ana (tipo: <class 'str'>)
Edad: 25 (tipo: <class 'int'>)
Altura: 1.65 (tipo: <class 'float'>)
Estudiante: True (tipo: <class 'bool'>)
```

Tips

- Para ver el tipo de una variable usa: `type(variable)`
- Para imprimir con formato usa f-strings: `print(f"Edad: {edad}")`
- Python es sensible a mayúsculas: `Nombre` y `nombre` son diferentes
- `input()` siempre devuelve string, conviértelo si necesitas número

Resumen

- Las variables guardan información
- Tipos principales: `str`, `int`, `float`, `bool`
- `type()` muestra el tipo de dato
- `input()` pide datos al usuario (siempre devuelve `string`)
- Usa `int()` o `float()` para convertir strings a números

02 - Operaciones Matemáticas

Operaciones básicas

Python puede hacer cálculos como una calculadora.

Suma (+)

```
resultado = 10 + 5 # 15
print(resultado)
```

Resta (-)

```
resultado = 10 - 5 # 5
print(resultado)
```

Multiplicación (*)

```
resultado = 10 * 5 # 50
print(resultado)
```

División (/)

```
resultado = 10 / 5 # 2.0 (siempre devuelve float)
print(resultado)
```

Operaciones especiales

División entera (//)

Divide y elimina los decimales:

```
resultado = 10 // 3 # 3 (no 3.333...)
resultado = 17 // 5 # 3 (no 3.4)
```

Módulo (%) - Resto de la división

```
resultado = 10 % 3 # 1 (10 dividido 3 da 3, y sobra 1)
resultado = 15 % 4 # 3 (15 dividido 4 da 3, y sobran 3)
```

```
# Útil para saber si un número es par
numero = 8
resto = numero % 2 # 0 → es par
```

Potencia (**)

```
resultado = 2 ** 3    # 8 (2 elevado a 3)
resultado = 5 ** 2    # 25 (5 al cuadrado)
resultado = 10 ** 0   # 1 (cualquier número elevado a 0 es 1)
```

Orden de operaciones

Python sigue las reglas matemáticas (PEMDAS): 1. **P**aréntesis 2. **E**xponentes (potencias) 3. **M**ultiplicación / **D**ivisión 4. **A**dición / **S**ustracción

```
resultado = 2 + 3 * 4    # 14 (primero 3*4=12, luego 2+12=14)
resultado = (2 + 3) * 4   # 20 (primero paréntesis 2+3=5, luego 5*4=20)
resultado = 10 / 2 + 3    # 8.0 (primero 10/2=5, luego 5+3=8)
```

Operaciones con variables

```
precio = 100
descuento = 20
precio_final = precio - descuento
print(precio_final)    # 80
```

```
base = 5
altura = 3
area = base * altura
print(area)    # 15
```

Modificar variables

Forma larga

```
puntos = 10
puntos = puntos + 5    # Suma 5 a los puntos actuales
print(puntos)    # 15
```

Forma corta (operadores compuestos)

```
puntos = 10
puntos += 5    # Equivale a: puntos = puntos + 5
print(puntos)    # 15
```

```
puntos -= 3    # Equivale a: puntos = puntos - 3
puntos *= 2    # Equivale a: puntos = puntos * 2
puntos /= 4    # Equivale a: puntos = puntos / 4
```

Usando input() con operaciones

```
# Calculadora simple con input()
num1 = float(input("Primer número: "))
num2 = float(input("Segundo número: "))

suma = num1 + num2
resta = num1 - num2
multiplicacion = num1 * num2
division = num1 / num2
```

```
print(f"Suma: {suma}")
print(f"Resta: {resta}")
print(f"Multiplicación: {multiplicacion}")
print(f"División: {division}")
```

Ejercicios

Ejercicio 1: Calculadora básica con input()

Pide al usuario dos números. Calcula e imprime: - La suma - La resta - La multiplicación - La división

Ejercicio 2: Calculadora de propina con input()

Pide al usuario el total de una cuenta de restaurante. Pide el porcentaje de propina que quiere dejar (ej: 15 para 15%).

Calcula: - La propina ($\text{cuenta} \times \text{porcentaje} / 100$) - El total a pagar ($\text{cuenta} + \text{propina}$)

Ejercicio 3: Conversor de temperatura con input()

Pide al usuario una temperatura en Celsius. Conviértela a Fahrenheit usando la fórmula:

$\text{Fahrenheit} = (\text{Celsius} \times 9/5) + 32$

Ejercicio 4: Área y perímetro de un círculo con input()

Pide al usuario el radio de un círculo. Usa $\pi = 3.14159$

Calcula: - Área = $\pi \times \text{radio}^2$ - Perímetro = $2 \times \pi \times \text{radio}$

Ejercicio 5: Repartir en partes iguales con input()

Pide al usuario: - Cantidad total de caramelos - Número de amigos

Calcula: - ¿Cuántos caramelos recibe cada amigo? (división entera) - ¿Cuántos caramelos sobran? (módulo)

Ejercicio 6: Calculadora de potencias con input()

Pide al usuario: - Un número base - Un exponente

Calcula y muestra el resultado de base elevado al exponente.

Ejercicio 7: Precio final con IVA con input()

Pide al usuario el precio de un producto (sin IVA). Calcula el precio con IVA del 21% ($\text{precio} \times 1.21$). Muestra ambos precios.

Ejercicio 8: Contador de puntos con operadores compuestos

Crea una variable `puntos = 0`.

Pide al usuario cuántos puntos gana en cada ronda (3 rondas). Usa el operador `+=` para ir sumando los puntos.

Muestra los puntos después de cada ronda y el total final.

Ejemplo:

```
Puntos iniciales: 0
¿Puntos ganados en ronda 1? 10
Puntos actuales: 10
¿Puntos ganados en ronda 2? 25
Puntos actuales: 35
¿Puntos ganados en ronda 3? 15
Puntos finales: 50
```

Tips

- La división (`/`) siempre devuelve un float, incluso si el resultado es entero
- Usa paréntesis cuando tengas dudas sobre el orden
- El operador `%` es muy útil para saber si un número es par: `numero % 2 == 0`
- Los operadores compuestos (`+=`, `-=`, etc.) hacen el código más corto

Resumen

Operador	Operación	Ejemplo	Resultado
<code>+</code>	Suma	<code>5 + 3</code>	8
<code>-</code>	Resta	<code>5 - 3</code>	2
<code>*</code>	Multipliación	<code>5 * 3</code>	15
<code>/</code>	División	<code>5 / 2</code>	2.5
<code>//</code>	División entera	<code>5 // 2</code>	2
<code>%</code>	Módulo (resto)	<code>5 % 2</code>	1
<code>**</code>	Potencia	<code>5 ** 2</code>	25
<code>+=</code>	Incremento	<code>x += 5</code>	<code>x = x + 5</code>

03 - Strings y Texto

¿Qué es un string?

Un **string** (cadena de texto) es una secuencia de caracteres. En Python se escriben entre comillas.

```
nombre = "Ana"
ciudad = 'Barcelona'
mensaje = "Hola, mundo!"
```

Puedes usar comillas simples (') o dobles ("), pero sé consistente.

Concatenar (unir) strings

Método 1: Con el operador +

```
nombre = "Carlos"
apellido = "García"
nombre_completo = nombre + " " + apellido
print(nombre_completo) # Carlos García

saludo = "Hola, " + nombre + "!"
print(saludo) # Hola, Carlos!
```

Con input()

```
nombre = input("¿Cómo te llamas? ")
edad = input("¿Cuántos años tienes? ")

# Usando concatenación con input
print("Hola " + nombre + ", tienes " + edad + " años")

# También puedes usar f-strings (ya visto en capítulo 1)
print(f"Hola {nombre}, tienes {edad} años")
```

Métodos útiles de strings

Los strings tienen funciones incorporadas llamadas “métodos”:

Cambiar mayúsculas/minúsculas

```
texto = "hola mundo"

print(texto.upper()) # HOLA MUNDO
print(texto.lower()) # hola mundo
print(texto.capitalize()) # Hola mundo
print(texto.title()) # Hola Mundo
```

Reemplazar texto

```
texto = "Me gusta Python"
nuevo = texto.replace("Python", "programar")
print(nuevo)  # Me gusta programar

frase = "hola hola hola"
nueva = frase.replace("hola", "adiós")
print(nueva)  # adiós adiós adiós
```

Otras operaciones útiles

```
texto = "Python es genial"

# Contar caracteres
longitud = len(texto)
print(longitud)  # 16

# Comprobar si contiene algo
contiene = "Python" in texto
print(contiene)  # True

contiene_java = "Java" in texto
print(contiene_java)  # False

# Dividir en palabras
palabras = texto.split()
print(palabras)  # ['Python', 'es', 'genial']
```

Uso práctico con input()

```
# Convertir input a mayúsculas para comparar
respuesta = input("¿Quieres continuar? (SI/NO): ")
respuesta = respuesta.upper()

if respuesta == "SI":
    print("Continuando...")
else:
    print("Finalizando...")

# O en una línea
respuesta = input("¿Quieres continuar? (SI/NO): ").upper()
```

Caracteres especiales

```
# Salto de línea: \n
print("Primera línea\nSegunda línea")

# Tabulación: \t
print("Nombre:\tCarlos")

# Comillas dentro de strings
print("Dijo: \"Hola\"")  # Dijo: "Hola"
print('Dijo: "Hola"')    # Más fácil con comillas simples
```

Ejercicios

Ejercicio 1: Presentación personalizada con `input()`

Pide al usuario: - Su nombre - Su ciudad - Su edad

Usando concatenación o f-strings, muestra: “Hola, soy [nombre], tengo [edad] años y vivo en [ciudad]”

Ejercicio 2: Manipular texto con `input()`

Pide al usuario una frase. 1. Conviértela a mayúsculas 2. Cuenta cuántos caracteres tiene 3. Reemplaza los espacios por guiones bajos (`_`)

Muestra el resultado de cada operación.

Ejercicio 3: Información de película con `input()`

Pide al usuario: - Título de una película - Año de estreno - Puntuación (0-10)

Muestra con concatenación o f-strings: “La película [titulo] ([año]) tiene una puntuación de [puntuacion]/10”

Ejercicio 4: Formatear nombre con `input()`

Pide al usuario su nombre completo (puede estar en mayúsculas, minúsculas o mezclado).

Formátalo para que cada palabra empiece con mayúscula. Ejemplo: “juan garcía perez” → “Juan García Pérez”

Pista: Usa el método `.title()`

Ejercicio 5: Creador de email con `input()`

Pide al usuario: - Su nombre - Su apellido - Dominio del email (ej: “gmail.com”)

Crea un email con el formato: `nombre.apellido@dominio` Todo en minúsculas.

Ejemplo: “Ana García” + “hotmail.com” → “ana.garcia@hotmail.com”

Ejercicio 6: Búsqueda en texto con `input()`

Pide al usuario: - Una frase - Una palabra a buscar en la frase

Comprueba si la palabra está en la frase (usa `in`). Muestra “La palabra [palabra] SÍ está en la frase” o “NO está”.

Ejercicio 7: Contador de vocales con `input()`

Pide al usuario una palabra. Cuenta cuántas veces aparece la letra ‘a’ (usa el método `.count()`). Muestra el resultado.

Pista: `texto.count("a")` cuenta las apariciones de “a”

Tips

- **Concatenación** con `+` y **f-strings** son las formas principales para combinar texto y variables
- Los métodos de strings **no modifican** el original, devuelven uno nuevo
- `len()` funciona con cualquier string
- Usa `in` para comprobar si un texto contiene otro
- `.lower()` o `.upper()` son útiles para comparar strings sin importar mayúsculas

Resumen

Operación	Código	Resultado
Concatenar	"Hola" + " " + "Ana"	"Hola Ana"
f-string	f"Hola {nombre}"	"Hola Ana"
Mayúsculas	"hola".upper()	"HOLA"
Minúsculas	"HOLA".lower()	"hola"
Title Case	"hola mundo".title()	"Hola Mundo"
Reemplazar	"hola".replace("h", "H")	"Hola"
Longitud	len("hola")	4
Contiene	"la" in "hola"	True
Contar	"hola".count("o")	1

04 - Condicionales - Tomar Decisiones

¿Qué son los condicionales?

Los condicionales permiten que el programa tome **diferentes caminos** según las condiciones que establezcas.

Estructura if / else

Sintaxis básica

```
if condición:
    # Código que se ejecuta si la condición es True
else:
    # Código que se ejecuta si la condición es False
```

Ejemplo simple

```
edad = 18

if edad >= 18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
```

Con input()

```
edad = int(input("¿Cuántos años tienes? "))

if edad >= 18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
```

IMPORTANTE: La indentación

La **indentación** (espacios al inicio) es **obligatoria** en Python:

```
# CORRECTO
if edad >= 18:
    print("Mayor de edad") # 4 espacios (o 1 tab)
    print("Puedes votar")

# ERROR
if edad >= 18:
print("Mayor de edad") # Sin espacios → ERROR!
```

Múltiples condiciones: if / elif / else

Cuando tienes más de dos opciones:

```
nota = 7
```

```
if nota >= 9:
    print("Sobresaliente")
elif nota >= 7:
    print("Notable")
elif nota >= 5:
    print("Aprobado")
else:
    print("Suspenso")
```

- if: primera condición
- elif: condiciones adicionales (else if)
- else: si ninguna condición anterior fue True

Con input()

```
nota = float(input("Introduce tu nota (0-10): "))
```

```
if nota >= 9:
    print("Sobresaliente")
elif nota >= 7:
    print("Notable")
elif nota >= 5:
    print("Aprobado")
else:
    print("Suspenso")
```

Operadores de comparación

Operador	Significado	Ejemplo	Resultado
==	Igual que	5 == 5	True
!=	Diferente de	5 != 3	True
>	Mayor que	5 > 3	True
<	Menor que	5 < 3	False
>=	Mayor o igual	5 >= 5	True
<=	Menor o igual	5 <= 3	False

Ejemplos

```
x = 10
y = 5
```

```
print(x == y)  # False
print(x != y)  # True
print(x > y)   # True
print(x < y)   # False
print(x >= 10) # True
print(y <= 5)  # True
```

Cuidado: = vs ==

```
# = es asignación (dar valor)
edad = 18

# == es comparación (comprobar igualdad)
if edad == 18:
    print("Tienes 18 años")
```

Operadores lógicos

Permiten combinar varias condiciones:

AND (y)

Ambas condiciones deben ser True:

```
edad = int(input("¿Cuántos años tienes? "))
tiene_carnet = input("¿Tienes carnet? (si/no): ").lower()

if edad >= 18 and tiene_carnet == "si":
    print("Puedes conducir")
else:
    print("No puedes conducir")
```

Condición 1	Condición 2	Resultado
True	True	True
True	False	False
False	True	False
False	False	False

OR (o)

Al menos una condición debe ser True:

```
dia = input("¿Qué día es hoy? ").lower()

if dia == "sábado" or dia == "domingo":
    print("Es fin de semana!")
else:
    print("Es día laboral")
```

Condición 1	Condición 2	Resultado
True	True	True
True	False	True
False	True	True
False	False	False

NOT (no)

Invierte el valor (True → False, False → True):

```
llueve = input("¿Está lloviendo? (si/no): ").lower() == "si"

if not llueve:
    print("Puedes salir sin paraguas")
```



```
else:
    print("Lleva paraguas")
```

Combinar operadores

```
temperatura = float(input("Temperatura actual: "))
hace_sol = input("¿Hace sol? (si/no): ").lower() == "si"

if temperatura > 20 and hace_sol:
    print("Buen día para la playa")
elif temperatura > 20 and not hace_sol:
    print("Está caluroso pero nublado")
else:
    print("Hace frío")
```

Condicionales con strings

```
color = input("¿Cuál es tu color favorito? ").lower()

if color == "rojo":
    print("El rojo es un color intenso")
elif color == "azul":
    print("El azul es relajante")
elif color == "verde":
    print("El verde es natural")
else:
    print(f"El {color} también es bonito")
```

Condicionales anidados

Puedes poner un if dentro de otro:

```
edad = int(input("¿Cuántos años tienes? "))

if edad >= 18:
    tiene_entrada = input("¿Tienes entrada? (si/no): ").lower()
    if tiene_entrada == "si":
        print("Puedes entrar")
    else:
        print("Necesitas una entrada")
else:
    print("Eres menor de edad, no puedes entrar")
```

Ejercicios

Ejercicio 1: Detector de par o impar con input()

Pide un número al usuario. Si es par (resto al dividir por 2 es 0), muestra “Es par”. Si no, muestra “Es impar”.

Pista: Usa el operador % (módulo)

Ejercicio 2: Mayor de edad con input()

Pide la edad al usuario. Si tiene 18 o más, muestra “Eres mayor de edad”. Si no, muestra “Eres menor de edad”.

Ejercicio 3: Clasificador de notas con input()

Pide una nota (0-10) al usuario. Clasifícala según: - 9-10: Sobresaliente - 7-8: Notable - 5-6: Aprobado - 0-4: Suspenso

Ejercicio 4: Control de acceso al concierto con input()

Pide al usuario: - Su edad - Si tiene entrada (si/no)

Puede entrar si tiene 18 o más años **Y** tiene entrada. Muestra “Puedes entrar” o “No puedes entrar” según corresponda.

Ejercicio 5: Descuento en tienda con input()

Pide el precio de un producto. Si el precio es mayor a 100 euros, aplica 20% de descuento. Si no, no hay descuento. Muestra el precio final.

Ejercicio 6: Día de la semana con input()

Pide al usuario un día de la semana. Si es “sábado” o “domingo”, muestra “Es fin de semana”. Si no, muestra “Es día laboral”.

Pista: Usa `.lower()` y `or`

Ejercicio 7: Verificador de contraseña con input()

Pide una contraseña al usuario. La contraseña correcta es “python2024”. Si la contraseña es correcta, muestra “Acceso permitido”. Si no, muestra “Contraseña incorrecta”.

Ejercicio 8: Calculadora con operación con input()

Pide al usuario: - Un número - Una operación (+, -, *, /) - Otro número

Realiza la operación correspondiente y muestra el resultado.

Pista: Usa `if/elif` para cada operación

Ejercicio 9: Clasificador de temperatura con input()

Pide la temperatura actual. Clasifícala según: - Menos de 10: “Hace frío” - 10-20: “Temperatura agradable” - 21-30: “Hace calor” - Más de 30: “Hace mucho calor”

Ejercicio 10: Decisión de salir con input()

Pide al usuario: - Si está lloviendo (si/no) - Temperatura actual

Decide: - Si llueve: “Lleva paraguas” - Si no llueve y temperatura > 25: “Hace buen tiempo para salir” - Si no llueve y temperatura <= 25: “Hace fresco, lleva chaqueta”

Tips

- La **indentación** es obligatoria (usa 4 espacios o 1 tab)
- Usa == para comparar, no =
- Para comparar strings ignorando mayúsculas: .lower()
- and requiere que **ambas** condiciones sean True
- or requiere que **al menos una** condición sea True
- Usa .lower() en inputs para aceptar “SI”, “Si”, “si”, etc.

Errores comunes

```
# ERROR 1: Usar = en vez de ==
if edad = 18: # ERROR!
if edad == 18: # CORRECTO

# ERROR 2: Olvidar los dos puntos :
if edad >= 18 # ERROR!
if edad >= 18: # CORRECTO

# ERROR 3: Sin indentación
if edad >= 18:
print("Mayor") # ERROR!

if edad >= 18:
    print("Mayor") # CORRECTO
```

Resumen

```
# Estructura básica
if condición:
    # código si True
else:
    # código si False

# Múltiples condiciones
if condición1:
    # código
elif condición2:
    # código
else:
    # código

# Operadores de comparación: ==, !=, >, <, >=, <=
# Operadores lógicos: and, or, not
```