

Condicionales en Python (if, elif, else)

1. Introducción a las decisiones en programación

1.1 ¿Qué son las condicionales?

En la vida cotidiana, constantemente tomamos decisiones basadas en condiciones:

- "Si tengo dinero, voy al cine"
- "Si llueve, llevo paraguas"
- "Si no tengo tarea, juego videojuegos"

En programación, las **condicionales** son estructuras que permiten que el código tome decisiones. Le permiten al programa ejecutar diferentes bloques de código dependiendo de si una condición es verdadera o falsa.

Las condicionales son fundamentales porque hacen que los programas sean **inteligentes** y puedan adaptarse a diferentes situaciones.

1.2 Ejemplo introductorio

Imagina un programa que verifica la edad de una persona para saber si puede entrar a una discoteca:

```
edad = 18

if edad >= 18:
    print("Puedes entrar a la discoteca")
else:
    print("Lo siento, eres muy joven")
```

¿Qué sucede aquí?

- Si `edad >= 18` es verdadero, ejecuta la primera línea de `print()` .
- Si es falso, ejecuta la segunda línea de `print()` .

2. Operadores de comparación

Para crear condiciones, necesitamos **comparar valores**. Python proporciona varios operadores de comparación que devuelven `True` o `False` .

2.1 Operadores de comparación básicos

Operador	Significado	Ejemplo	Resultado
==	Igual a	5 == 5	True
!=	No igual a	5 != 3	True
>	Mayor que	7 > 5	True
<	Menor que	3 < 5	True
>=	Mayor o igual que	5 >= 5	True
<=	Menor o igual que	3 <= 5	True

2.2 Ejemplos detallados

```
# Ejemplos con números
print(10 == 10) # True (10 es igual a 10)
print(10 != 5)  # True (10 no es igual a 5)
print(15 > 10)  # True (15 es mayor que 10)
print(8 < 10)   # True (8 es menor que 10)
print(10 >= 10) # True (10 es mayor o igual que 10)
print(8 <= 10)  # True (8 es menor o igual que 10)

# Ejemplos con variables
edad = 25
print(edad > 18) # True
print(edad == 25) # True
print(edad < 20) # False

# Ejemplos con strings
nombre = "Carlos"
print(nombre == "Carlos") # True
print(nombre != "Juan")   # True
print(nombre == "carlos") # False (Python distingue mayúsculas/minúsculas)
```

2.3 Punto importante: = vs ==

Error común: Confundir = con ==

- = es el **operador de asignación** (guarda un valor en una variable)
- == es el **operador de comparación** (compara dos valores)

```
edad = 25 # Asigna 25 a la variable edad (correcto)
if edad == 25: # Compara si edad es igual a 25 (correcto)
    print("Tienes 25 años")
```

```
if edad = 25: # ✗ ERROR: no puedes usar = en una condición
    print("Esto dará error")
```

3. Estructura if (si)

3.1 Sintaxis básica del if

La estructura más simple de una condicional es el `if` (si):

```
if condición:
    # Bloque de código que se ejecuta si la condición es True
    print("La condición es verdadera")
```

Puntos clave:

- La palabra `if` es seguida por una condición.
- Después de la condición va dos puntos `:` (obligatorio).
- El código dentro del `if` debe estar **indentado** (con espacios o tabulaciones).

3.2 Indentación: El concepto más importante

En Python, la **indentación** (espacios al inicio de las líneas) es fundamental. Define qué código pertenece a qué bloque.

```
if edad >= 18:
    print("Eres mayor de edad")      # Esta línea está indentada (dentro del if)
    print("Puedes votar")           # Esta línea también está indentada
print("Fin del programa")           # Esta línea NO está indentada (fuera del if)
```

¿Cómo indentar correctamente?

- Usa 4 espacios (o una tabulación) para indentar.
- La mayoría de editores de código (Replit, VS Code, etc.) lo hacen automáticamente cuando presionas Enter después de los dos puntos.

3.3 Ejemplos de if simple

Ejemplo 1: Verificar si alguien es mayor de edad

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

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

Si el usuario ingresa 20, se mostrará: Eres mayor de edad Si el usuario ingresa 15, no se mostrará nada (porque la condición es falsa).

Ejemplo 2: Verificar si un número es positivo

```
numero = int(input("Ingresa un número: "))  
  
if numero > 0:  
    print("El número es positivo")
```

Ejemplo 3: Verificar pertenencia a un grupo

```
nombre = input("¿Cuál es tu nombre? ")  
  
if nombre == "Carlos":  
    print("¡Hola Carlos! Bienvenido")
```

4. Estructura if-else (si-sino)

4.1 Sintaxis del if-else

A menudo queremos hacer algo si una condición es verdadera, y algo diferente si es falsa. Para esto usamos `else`:

```
if condición:  
    # Bloque que se ejecuta si la condición es True  
    print("La condición es verdadera")  
else:  
    # Bloque que se ejecuta si la condición es False  
    print("La condición es falsa")
```

4.2 Ejemplos de if-else

Ejemplo 1: Mayor o menor de edad

```
edad = int(input("¿Cuántos años tienes? "))  
  
if edad >= 18:  
    print("Eres mayor de edad")
```

```
else:  
    print("Eres menor de edad")
```

Si el usuario ingresa 20: Eres mayor de edad Si el usuario ingresa 15: Eres menor de edad

Ejemplo 2: Número par o impar

```
numero = int(input("Ingresa un número: "))  
  
if numero % 2 == 0:  
    print("El número es par")  
else:  
    print("El número es impar")
```

Explicación: `numero % 2` devuelve el residuo de la división. Si es 0, el número es par; si es 1, es impar.

Ejemplo 3: Contraseña correcta

```
contrasena = input("Ingresa la contraseña: ")  
  
if contrasena == "python123":  
    print("Acceso permitido")  
else:  
    print("Contraseña incorrecta")
```

Ejemplo 4: Número positivo, negativo o cero

```
numero = int(input("Ingresa un número: "))  
  
if numero > 0:  
    print("El número es positivo")  
else:  
    if numero < 0:  
        print("El número es negativo")  
    else:  
        print("El número es cero")
```

Aunque esto funciona, hay una forma mejor: usar `elif`.

5. Estructura if-elif-else (si-pero si-sino)

5.1 Sintaxis del if-elif-else

Cuando tienes **múltiples condiciones** que verificar, usa `elif` (else if). Te permite escribir código más limpio y legible.

```
if condición1:
    # Se ejecuta si condición1 es True
    print("Primera condición verdadera")
elif condición2:
    # Se ejecuta si condición1 es False y condición2 es True
    print("Segunda condición verdadera")
elif condición3:
    # Se ejecuta si condición1 y condición2 son False, pero condición3 es True
    print("Tercera condición verdadera")
else:
    # Se ejecuta si todas las condiciones anteriores son False
    print("Ninguna condición fue verdadera")
```

Puntos importantes:

- Puedes tener **tantos** `elif` como necesites (0, 1, 2, 3, etc.).
- Se evalúan **de arriba a abajo**.
- Solo se ejecuta el **primer bloque** que cumple su condición.
- El `else` es **opcional**.

5.2 Ejemplo: Clasificación de notas

Este es un ejemplo clásico que verás mucho en educación:

```
nota = float(input("Ingresa tu nota (0 a 20): "))

if nota < 0 or nota > 20:
    print("Nota fuera de rango")
elif nota < 10:
    print("Reprobado")
elif nota <= 13:
    print("En recuperación")
elif nota <= 16:
    print("Aprobado")
else:
    print("Excelente")
```

Flujo de ejecución:

- Si digitas 5: Se ejecuta `print("Reprobado")`
- Si digitas 12: Se ejecuta `print("En recuperación")`
- Si digitas 15: Se ejecuta `print("Aprobado")`
- Si digitas 19: Se ejecuta `print("Excelente")`

5.3 Ejemplo: Sistema de clasificación por edad

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

if edad < 13:
    print("Eres un niño")
elif edad < 18:
    print("Eres un adolescente")
elif edad < 65:
    print("Eres un adulto")
else:
    print("Eres un adulto mayor")
```

5.4 Ejemplo: Determinar el tipo de triángulo

```
# Supongamos que ya tenemos los tres lados
lado1 = 5
lado2 = 5
lado3 = 5

if lado1 == lado2 == lado3:
    print("Es un triángulo equilátero")
elif lado1 == lado2 or lado2 == lado3 or lado1 == lado3:
    print("Es un triángulo isósceles")
else:
    print("Es un triángulo escaleno")
```

6. Operadores lógicos: and, or, not

6.1 El operador AND (y)

El operador `and` devuelve `True` solo si **todas** las condiciones son verdaderas.

Sintaxis:

```
if condición1 and condición2:
    print("Ambas condiciones son verdaderas")
```

Tabla de verdad del AND:

condición1	condición2	resultado
True	True	True

condición1	condición2	resultado
True	False	False
False	True	False
False	False	False

Ejemplos:

```
edad = 25
tiene_licencia = True

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

Salida: Puedes conducir (porque ambas condiciones son verdaderas)

```
# Verificar si un número está en un rango
numero = 15

if numero > 10 and numero < 20:
    print("El número está entre 10 y 20")
else:
    print("El número está fuera del rango")
```

Salida: El número está entre 10 y 20

```
# Verificar acceso a un sistema
usuario = "admin"
contrasena = "1234"
es_administrador = True

if usuario == "admin" and contrasena == "1234" and es_administrador:
    print("Acceso a panel administrativo concedido")
else:
    print("Acceso denegado")
```

6.2 El operador OR (o)

El operador `or` devuelve `True` si **al menos una** de las condiciones es verdadera.

Sintaxis:


```
if condición1 or condición2:  
    print("Al menos una condición es verdadera")
```

Tabla de verdad del OR:

condición1	condición2	resultado
True	True	True
True	False	True
False	True	True
False	False	False

Ejemplos:

```
es_fin_de_semana = True  
es_feriado = False  
  
if es_fin_de_semana or es_feriado:  
    print("Hoy no hay clases")  
else:  
    print("Hoy hay clases")
```

Salida: Hoy no hay clases (porque una de las condiciones es verdadera)

```
# Verificar si alguien es estudiante o profesor  
role = "profesor"  
  
if role == "estudiante" or role == "profesor":  
    print("Puedes acceder a la plataforma educativa")  
else:  
    print("Acceso denegado")
```

Salida: Puedes acceder a la plataforma educativa

```
# Verificar edad para descuento  
edad = 70  
  
if edad < 12 or edad > 65:  
    print("Tienes descuento")  
else:  
    print("Precio regular")
```

Salida: Tienes descuento

6.3 El operador NOT (no)

El operador `not` **niega** una condición. Convierte `True` en `False` y viceversa.

Sintaxis:

```
if not condición:
    print("La condición es falsa")
```

Tabla de verdad del NOT:

condición	not condición
True	False
False	True

Ejemplos:

```
es_lluvia = False

if not es_lluvia:
    print("Vamos al parque")
else:
    print("Nos quedamos en casa")
```

Salida: Vamos al parque (porque "no lluvia" es verdadero)

```
# Verificar si no está activo
cuenta_activa = False

if not cuenta_activa:
    print("Tu cuenta está desactivada")
else:
    print("Tu cuenta está activa")
```

Salida: Tu cuenta está desactivada

```
# Combinando not con otras condiciones
edad = 16
tiene_permiso = False

if edad >= 18 and not tiene_permiso:
    print("Eres mayor pero necesitas permiso")
elif edad < 18 and not tiene_permiso:
    print("Eres menor y necesitas permiso de padres")
```

```
else:  
    print("Puedes pasar")
```

6.4 Combinando operadores lógicos

Puedes combinar múltiples operadores lógicos para crear condiciones complejas:

```
# Ejemplo: Validar credenciales de un usuario  
usuario = "carlos"  
contrasena = "pass123"  
es_admin = True  
cuenta_activa = True  
  
if (usuario == "carlos" and contrasena == "pass123") and (es_admin or cuenta_activa):  
    print("Acceso concedido")  
else:  
    print("Acceso denegado")
```

Salida: Acceso concedido (porque todas las condiciones se cumplen)

7. Ejercicio guiado: Sistema de clasificación de notas

Ahora crearemos un programa completo que clasifique las notas de un estudiante.

7.1 Objetivo

Crear un programa que:

1. Pida la nota de un estudiante (número de 0 a 20).
2. Valide que la nota esté en el rango correcto.
3. Clasifique la nota y muestre el resultado.
4. Opcionalmente, indique si el estudiante necesita estudio extra.

7.2 Planificación

1. Entrada: Pedir la nota al usuario
2. Validación: Verificar que esté entre 0 y 20
3. Clasificación:
 - Menor a 10: Reprobado
 - 10 a 13: En recuperación
 - 14 a 16: Aprobado
 - 17 a 20: Excelente
4. Salida: Mostrar clasificación y consejo

7.3 Código completo con explicación

```
# Sistema de clasificación de notas

print("=== CLASIFICADOR DE NOTAS ===")
print()

# PASO 1: Pedir nota al usuario
nota = float(input("Ingresa tu nota (0 a 20): "))

# PASO 2: Validar que esté en rango
if nota < 0 or nota > 20:
    print("❌ Nota fuera de rango. Debe estar entre 0 y 20")
# PASO 3: Clasificar la nota
elif nota < 10:
    print(f"Tu nota es: {nota}")
    print("Clasificación: ❌ REPROBADO")
    print("Recomendación: Necesitas estudiar más. Consulta con tu profesor.")
elif nota <= 13:
    print(f"Tu nota es: {nota}")
    print("Clasificación: ⚠️ EN RECUPERACIÓN")
    print("Recomendación: Debes mejorar. Intenta en el próximo examen.")
elif nota <= 16:
    print(f"Tu nota es: {nota}")
    print("Clasificación: ✅ APROBADO")
    print("Recomendación: ¡Bien hecho! Sigue así.")
else: # nota <= 20
    print(f"Tu nota es: {nota}")
    print("Clasificación: 🌟 EXCELENTE")
    print("Recomendación: ¡Felicidades! Eres un estudiante destacado.")
```

7.4 Salida del programa

Si el usuario ingresa 15:

```
=== CLASIFICADOR DE NOTAS ===

Ingresa tu nota (0 a 20): 15
Tu nota es: 15.0
Clasificación: ✅ APROBADO
Recomendación: ¡Bien hecho! Sigue así.
```

Si el usuario ingresa 5:

```
=== CLASIFICADOR DE NOTAS ===

Ingresa tu nota (0 a 20): 5
Tu nota es: 5.0
```

Clasificación: **✗** REPROBADO

Recomendación: Necesitas estudiar más. Consulta con tu profesor.

Si el usuario ingresa 25:

```
=== CLASIFICADOR DE NOTAS ===
```

Ingresa tu nota (0 a 20): 25

✗ Nota fuera de rango. Debe estar entre 0 y 20

7.5 Variación: Sistema de clasificación con rangos de letras

```
# Versión alternativa usando letras
```

```
print("=== CLASIFICADOR CON LETRAS ===")  
print()
```

```
nota = float(input("Ingresa tu nota (0 a 20): "))
```

```
if nota < 0 or nota > 20:  
    print("Nota fuera de rango")  
elif nota < 10:  
    print(f"Nota: {nota} → Letra: F (Reprobado)")  
elif nota < 12:  
    print(f"Nota: {nota} → Letra: D (Deficiente)")  
elif nota < 14:  
    print(f"Nota: {nota} → Letra: C (Regular)")  
elif nota < 17:  
    print(f"Nota: {nota} → Letra: B (Bueno)")  
else:  
    print(f"Nota: {nota} → Letra: A (Excelente)")
```

8. Ejercicio de evaluación

8.1 Enunciado

Crea un programa que determine si una persona puede **solicitar un préstamo bancario** basado en los siguientes criterios:

1. **Edad:** Debe ser mayor o igual a 18 años.
2. **Ingresos mensuales:** Deben ser mayores a \$500.
3. **Historial crediticio:** Debe tener un buen historial (sin deudas previas).
4. **Monto del préstamo:** El programa debe validar que el monto sea razonable (menor a 10 veces los ingresos mensuales).

Requisitos:

- El programa debe pedir: edad, ingresos mensuales, si tiene deudas (sí/no), y el monto solicitado.
- Mostrar un mensaje claro indicando si el préstamo es **aprobado** o **rechazado**, y la razón.

Ejemplo de ejecución:

```
=== SOLICITUD DE PRÉSTAMO ===
```

```
¿Cuántos años tienes? 25
¿Cuáles son tus ingresos mensuales? 1500
¿Tienes deudas previas? (sí/no) no
¿Cuánto deseas pedir prestado? 5000
```

```
Verificando elegibilidad...
```

```
✅ Préstamo APROBADO
```

```
Razón: Cumples todos los requisitos.
```

```
Monto máximo que podías pedir: $15000 (10 × ingresos)
```

8.2 Pistas para resolver

- Necesitarás variables para: `edad`, `ingresos`, `tiene_deudas`, `monto_solicitado`.
- Para leer "sí/no", usa `input()` y compara con strings.
- Usa `if-elif-else` para cada validación.
- Usa `and` para combinar múltiples condiciones.

9. Solución del ejercicio de evaluación

```
# Sistema de aprobación de préstamo bancario
```

```
print("=== SOLICITUD DE PRÉSTAMO ===")
print()
```

```
# PASO 1: Pedir información al usuario
```

```
edad = int(input("¿Cuántos años tienes? "))
ingresos = float(input("¿Cuáles son tus ingresos mensuales? $"))
tiene_deudas = input("¿Tienes deudas previas? (sí/no): ")
monto_solicitado = float(input("¿Cuánto deseas pedir prestado? $"))
```

```
print()
print("Verificando elegibilidad...")
print()
```

```
# PASO 2: Validar cada criterio
```

```
monto_maximo = ingresos * 10
deudas_limpio = tiene_deudas.lower() != "sí" and tiene_deudas.lower() != "si"
```

```
# PASO 3: Determinar si se aprueba o rechaza
if edad < 18:
    print("❌ Préstamo RECHAZADO")
    print("Razón: Debes ser mayor de 18 años para solicitar un préstamo.")
elif ingresos <= 500:
    print("❌ Préstamo RECHAZADO")
    print("Razón: Tus ingresos son muy bajos. Mínimo requerido: $500")
elif not deudas_limpio:
    print("❌ Préstamo RECHAZADO")
    print("Razón: Tienes deudas previas. Debes liquidarlas primero.")
elif monto_solicitado > monto_maximo:
    print("❌ Préstamo RECHAZADO")
    print(f"Razón: El monto solicitado excede tu capacidad de pago.")
    print(f"Monto máximo permitido: ${monto_maximo:.2f} (10 × tus ingresos)")
else:
    print("✅ Préstamo APROBADO")
    print("Razón: Cumples todos los requisitos.")
    print(f"Monto aprobado: ${monto_solicitado:.2f}")
    print(f"Monto máximo que podías pedir: ${monto_maximo:.2f}")
    print()
    print("Felicidades! Tu préstamo ha sido aprobado.")
    print("Un agente se pondrá en contacto contigo en 24 horas.")
```

9.1 Explicación de la solución

- Línea 7-10: Recibimos toda la información necesaria.
- Línea 13: Calculamos el monto máximo (10 veces los ingresos).
- Línea 14: Creamos una variable booleana que es `True` si no tiene deudas.
 - `tiene_deudas.lower()` convierte a minúsculas para comparar sin importar mayúsculas.
- Línea 17-34: Sistema de validaciones:
 - Si `edad < 18`: rechaza.
 - Si `ingresos <= 500`: rechaza.
 - Si tiene deudas: rechaza.
 - Si `monto > monto máximo`: rechaza.
 - Si pasa todas las validaciones: aprueba.

9.2 Alternativa más compacta

Si prefieres una versión más corta usando `and` :

```
print("=== SOLICITUD DE PRÉSTAMO ===")
print()

edad = int(input("¿Cuántos años tienes? "))
ingresos = float(input("¿Cuáles son tus ingresos mensuales? $"))
tiene_deudas = input("¿Tienes deudas previas? (sí/no): ")
```

```
monto_solicitado = float(input("¿Cuánto deseas pedir prestado? $"))

print()
monto_maximo = ingresos * 10
deudas_limpio = tiene_deudas.lower() not in ["sí", "si"]

if edad >= 18 and ingresos > 500 and deudas_limpio and monto_solicitado <= monto_maximo:
    print("✅ Préstamo APROBADO")
    print(f"Monto aprobado: ${monto_solicitado:.2f}")
else:
    print("❌ Préstamo RECHAZADO")
    if edad < 18:
        print("- Edad insuficiente (mínimo 18 años)")
    if ingresos <= 500:
        print("- Ingresos insuficientes (mínimo $500)")
    if not deudas_limpio:
        print("- Tienes deudas previas")
    if monto_solicitado > monto_maximo:
        print(f"- Monto solicitado excede el límite: ${monto_maximo:.2f}")
```

10. Ejercicios adicionales para practicar

10.1 Ejercicio: Sistema de autenticación

Crea un programa que:

- Pida un usuario y contraseña.
- Valide contra credenciales predefinidas.
- Muestre un mensaje diferente si el usuario existe pero la contraseña es incorrecta, o si ninguno de los dos es correcto.

10.2 Ejercicio: Calculadora de IMC (Índice de Masa Corporal)

Crea un programa que:

- Pida peso (en kg) y altura (en metros).
- Calcule IMC = peso / (altura²).
- Clasifique según:
 - IMC < 18.5: Bajo peso
 - 18.5 ≤ IMC < 25: Peso normal
 - 25 ≤ IMC < 30: Sobrepeso
 - IMC ≥ 30: Obesidad

10.3 Ejercicio: Juego: Adivina el número

Crea un programa que:

- Tenga un número secreto (por ejemplo, 42).
- Pida al usuario que adivine.
- Diga si es mayor, menor o igual al número secreto.
- Diga "Ganaste" si acierta.

10.4 Ejercicio: Validar una contraseña

Crea un programa que valide que una contraseña cumpla:

- Mínimo 8 caracteres.
- Contiene al menos una letra mayúscula.
- Contiene al menos un número.
- Si cumple todo, muestra "Contraseña fuerte".

11. Tabla resumen de condicionales

Estructura	Sintaxis	Uso
if	<code>if condición:</code>	Ejecuta si verdadero
if-else	<code>if condición: ... else:</code>	Verdadero o falso
if-elif-else	<code>if cond1: ... elif cond2: ... else:</code>	Múltiples opciones
and	<code>cond1 and cond2</code>	Verdadero si TODAS son verdaderas
or	<code>cond1 or cond2</code>	Verdadero si AL MENOS UNA es verdadera
not	<code>not condición</code>	Niega la condición

12. Errores comunes al usar condicionales

Error 1: Olvidar los dos puntos

```
if edad >= 18 # ❌ Falta los dos puntos
    print("Eres mayor de edad")
```

Solución:

```
if edad >= 18: # ✅ Dos puntos presentes
    print("Eres mayor de edad")
```

Error 2: Indentación incorrecta

```
if edad >= 18:
print("Eres mayor de edad") # ❌ Sin indentación
```

Solución:

```
if edad >= 18:
    print("Eres mayor de edad") # ✅ Indentado correctamente
```

Error 3: Confundir = con ==

```
if edad = 18: # ❌ Asigna en lugar de comparar
    print("Tienes 18 años")
```

Solución:

```
if edad == 18: # ✅ Compara correctamente
    print("Tienes 18 años")
```

Error 4: Lógica invertida con not

```
if not edad > 18: # Confuso
    print("Eres menor de edad")
```


Mejor:

```
if edad <= 18: # ✅ Más claro
    print("Eres menor de edad")
```

Error 5: Múltiples condiciones sin and/or

```
if edad >= 18, ingresos > 1000: # ❌ Sintaxis incorrecta
    print("Puedes solicitar crédito")
```

Solución:

```
if edad >= 18 and ingresos > 1000: #  Usa and
    print("Puedes solicitar crédito")
```

13. Flujo de ejecución: Visualizando cómo funciona

Es importante entender cómo Python ejecuta las condicionales paso a paso.

Ejemplo: Diagrama de flujo mental

```
edad = 16

if edad < 13:
    print("Eres un niño")           # ← NO se ejecuta (edad no es < 13)
elif edad < 18:
    print("Eres adolescente")      # ← SÍ se ejecuta (edad es < 18)
elif edad < 65:
    print("Eres adulto")           # ← NO se ejecuta (ya se ejecutó elif anterior)
else:
    print("Eres adulto mayor")     # ← NO se ejecuta (ya se ejecutó elif anterior)
```

Salida: Eres adolescente

Puntos clave:

- Python evalúa **de arriba hacia abajo**.
- Se ejecuta el **primer bloque** que cumple su condición.
- Los bloques posteriores **se ignoran**.

14. Resumen de conceptos clave del PDF 2

Concepto	Descripción	Ejemplo
Condición	Expresión que es verdadera o falsa	edad > 18
if	Ejecuta si la condición es verdadera	if edad >= 18:
else	Ejecuta si todas las condiciones anteriores son falsas	else:
elif	Otra condición si la anterior fue falsa	elif edad < 13:
==	Operador de comparación: igual	nombre == "Carlos"

Concepto	Descripción	Ejemplo
<code>!=</code>	Operador de comparación: no igual	<code>edad != 18</code>
<code>></code>	Mayor que	<code>precio > 100</code>
<code><</code>	Menor que	<code>edad < 65</code>
<code>>=</code>	Mayor o igual	<code>nota >= 10</code>
<code><=</code>	Menor o igual	<code>edad <= 20</code>
<code>and</code>	Verdadero si TODAS las condiciones son verdaderas	<code>edad >= 18 and ingresos > 500</code>
<code>or</code>	Verdadero si AL MENOS UNA condición es verdadera	<code>es_fin_semana or es_feriado</code>
<code>not</code>	Niega una condición	<code>not tiene_deudas</code>
Indentación	Espacios que marcan el bloque de código	Usar 4 espacios

15. Preguntas frecuentes del PDF 2

P: ¿Puedo tener más de 3 elif?

R: Sí, puedes tener tantos `elif` como necesites. No hay límite.

P: ¿Es obligatorio el else?

R: No. `else` es opcional. Puedes usar solo `if` o `if-elif` sin `else`.

P: ¿Cuál es la diferencia entre `and` y `or`?

R: `and` requiere que TODAS las condiciones sean verdaderas. `or` requiere que AL MENOS UNA sea verdadera.

P: ¿Puedo anidar condicionales (un `if` dentro de otro)?

R: Sí, pero puede hacerse confuso. Es mejor usar `and` u `or` cuando es posible.

P: ¿Qué pasa si no indento correctamente?

R: Python dará un error de indentación. Es muy importante respetar los espacios.

P: ¿Cómo comparo strings sin que importe mayúsculas/minúsculas?

R: Usa `.lower()` o `.upper()` : `if nombre.lower() == "carlos":`

Conclusión

Felicidades por completar el PDF 2. Ahora entiendes:

- Cómo usar operadores de comparación (`==`, `!=`, `>`, `<`, `>=`, `<=`).
- La estructura `if` para decisiones simples.
- La estructura `if-else` para dos opciones.
- La estructura `if-elif-else` para múltiples opciones.
- Los operadores lógicos `and`, `or`, `not`.
- Cómo combinar condiciones complejas.
- La importancia de la indentación en Python.

En el siguiente PDF (Bucles), aprenderás a **repetir código** usando `while` y `for`, lo que es fundamental para procesar colecciones de datos y automatizar tareas.