

Estructuras Condicionales en Python

Introducción

Las estructuras condicionales son fundamentales en la programación, ya que permiten que un programa tome decisiones y ejecute diferentes bloques de código en función de si una condición es verdadera o falsa. Python, como lenguaje de programación, ofrece una sintaxis clara y concisa para implementar estas estructuras.

Este informe tiene como objetivo explicar las estructuras condicionales principales en Python: `if`, `elif` (else if) y `else`, junto con ejemplos prácticos.

La Sentencia `if`

La sentencia `if` es la estructura condicional más básica. Permite ejecutar un bloque de código **solo si** una condición específica es verdadera (True).

Sintaxis General:if condicion:

```
# Bloque de código a ejecutar si la condición es Verdadera (True)

instruccion_1

instruccion_2

# ...
```

Puntos Clave:

- La `condicion` es una expresión que se evalúa como `True` o `False`.
- El bloque de código que pertenece al `if` debe estar **indentado** (generalmente con 4 espacios). La indentación es crucial en Python para definir bloques de código.
- El `if` termina con dos puntos (`:`).

Ejemplo:edad = 20

```
if edad >= 18:
```

```
print("Es mayor de edad.")
```

La Sentencia **if** - **else**

A menudo, necesitamos ejecutar un bloque de código si la condición es verdadera y **otro** bloque de código si es falsa. Para esto, utilizamos la sentencia **else**.

Sintaxis General:if condicion:

```
# Bloque 1: Se ejecuta si la condición es Verdadera
```

```
instruccion_a
```

```
else:
```

```
# Bloque 2: Se ejecuta si la condición es Falsa
```

```
instruccion_b
```

Ejemplo:temperatura = 15

```
if temperatura > 25:
```

```
    print("Hace calor.")
```

```
else:
```

```
    print("No hace tanto calor.")
```

La Sentencia **if** - **elif** - **else**

Cuando hay **múltiples** condiciones a evaluar, se utiliza la sentencia **elif** (abreviatura de "else if"). Python evalúa las condiciones en orden. Tan pronto como una condición **if** o **elif** se evalúa como verdadera, se ejecuta el bloque de código asociado y se omiten todas las comprobaciones restantes. Si ninguna condición es verdadera, se ejecuta el bloque **else** (si existe).

Sintaxis General:if condicion_1:

```
# Bloque A: si condicion_1 es Verdadera
```

```
elif condicion_2:
```

```
    # Bloque B: si condicion_1 es Falsa Y condicion_2 es Verdadera
```

```
elif condicion_3:
```

```
    # Bloque C: si condicion_1 y condicion_2 son Falsas Y condicion_3 es Verdadera
```

```
else:
```

```
    # Bloque D: si todas las condiciones anteriores son Falsas
```

Ejemplo de Clasificación de Notas:

Nota	Clasificación
>= 90	Excelente
>= 70	Bueno
>= 60	Suficiente
< 60	Reprobado

El código para clasificar la nota sería: `nota = 75`

```
if nota >= 90:
```

```
    print("Excelente")
```

```
elif nota >= 70:
```

```
    print("Bueno")
```

```
elif nota >= 60:
```

```
    print("Suficiente")
```

```
else:
```

```
print("Reprobado")
```

Condiciones Múltiples (Operadores Lógicos)

Para evaluar múltiples sub-condiciones dentro de una única sentencia `if`, utilizamos los operadores lógicos `and`, `or`, y `not`.

Operador	Significado	Uso
<code>and</code>	Y lógico	Ambas sub-condiciones deben ser True.
<code>or</code>	O lógico	Al menos una sub-condición debe ser True.
<code>not</code>	NO lógico	Invierte el valor de verdad (True se vuelve False, y viceversa).

Ejemplos de Operadores Lógicos:

1. **`and` (Y):** Solo se otorga el descuento si se cumplen ambas condiciones.
`compra = 120`

```
es_cliente_premium = True
```

```
if compra > 100 and es_cliente_premium:
```

```
    print("¡Felicitaciones! Recibe un descuento.")
```

2. **`or` (O):** Se permite la entrada si se cumple al menos una condición.
`tiene_invitacion = False`

```
es_organizador = True
```

```
if tiene_invitacion or es_organizador:
```

```
    print("Adelante.")
```

```
else:
```

```
print("No puede ingresar.")
```