

Puntos Clave

1. Los operadores de **comparación** (o también denominados operadores *relacionales*) se utilizan para comparar valores. La siguiente tabla ilustra cómo funcionan los operadores de comparación, asumiendo que `x=0`, `y=1` y `z=0`:

Operador	Descripción	Ejemplo
<code>==</code>	Devuelve si los valores de los operandos son iguales, y <code>False</code> de lo contrario.	<pre>x == y # False x == z # True</pre>
<code>!=</code>	Devuelve <code>True</code> si los valores de los operandos no son iguales, y <code>False</code> de lo contrario.	<pre>x != y # True x != z # False</pre>
<code>></code>	Devuelve <code>True</code> si el valor del operando izquierdo es mayor que el valor del operando derecho, y <code>False</code> de lo contrario.	<pre>x > y # False y > z # True</pre>
<code><</code>	Devuelve <code>True</code> si el valor del operando izquierdo es menor que el valor del operando derecho, y <code>False</code> de lo contrario.	<pre>x < y # True y < z # False</pre>
<code>>=</code>	Devuelve <code>True</code> si el valor del operando izquierdo es mayor o igual al valor del operando derecho, y <code>False</code> de lo contrario.	<pre>x >= y # False x >= z # True y >= z # True</pre>
<code><=</code>	Devuelve <code>True</code> si el valor del operando izquierdo es menor o igual al valor del operando derecho, y <code>False</code> de lo contrario.	<pre>x <= y # True x <= z # True y <= z # False</pre>

2. Cuando deseas ejecutar algún código solo si se cumple una determinada condición, puedes usar una **sentencia condicional**:

- Una única sentencia `if`, por ejemplo:

```
x = 10
```

```
if x == 10: # condición  
    print("x es igual a 10") # Ejecutado si la condición es  
    Verdadera.
```

- Una serie de sentencias `if`, por ejemplo:

```
x = 10

if x > 5: # primera condición
    print("x es mayor que 5") # Ejecutado si la primera
condición es Verdadera.

if x < 10: # segunda condición
    print("x is less than 10") # Ejecutado si la segunda
condición es Verdadera.

if x == 10: # tercera condición
    print("x is equal to 10") # Ejecutado si la tercera
condición es Verdadera.
```

Cada sentencia `if` se prueba por separado.

- Una sentencia de `if-else`, por ejemplo:

```
x = 10

if x < 10: # Condición
    print("x es menor que 10") # Ejecutado si la condición es
Verdadera.

else:
    print("x es mayor o igual a 10") # Ejecutado si la
condición es Falsa.
```

- Una serie de sentencias `if` seguidas de un `else`, por ejemplo:

```
x = 10
```

```

if x > 5: # True
    print("x > 5")

if x > 8: # True
    print("x > 8")

if x > 10: # False
    print("x > 10")

else:
    print("se ejecutará el else")

```

Cada `if` se prueba por separado. El cuerpo de `else` se ejecuta si el último `if` es `False`.

- La sentencia `if-elif-else`, por ejemplo:

```

x = 10

if x == 10: # True
    print("x == 10")

if x > 15: # False
    print("x > 15")

elif x > 10: # False
    print("x > 10")

elif x > 5: # True
    print("x > 5")

else:
    print("else no será ejecutado")

```

Si la condición para `if` es `False`, el programa verifica las condiciones de los bloques `elif` posteriores: el primer `elif` que sea `True` es el que se ejecuta. Si todas las condiciones son `False`, se ejecutará el bloque `else`.

- Sentencias condicionales anidadas, ejemplo:

```

x = 10

```

```
if x > 5: # True
    if x == 6: # False
        print("anidado: x == 6")
    elif x == 10: # True
        print("anidado: x == 10")
    else:
        print("anidado: else")
else:
    print("else")
```