

# OPERADORES LÓGICOS

Los operadores lógicos son elementos fundamentales en programación que se utilizan para combinar o modificar expresiones booleanas. Estos operadores permiten evaluar múltiples condiciones y determinar si una expresión lógica es verdadera o falsa. Los operadores lógicos más comunes son **and**, **or** y **not**.

## Operador and (Y lógico)

El operador **and** devuelve **True** si y solo si ambas expresiones que conecta son verdaderas. En otras palabras, la expresión compuesta es verdadera solo si todas las condiciones individuales son verdaderas.

```
python Copy code  
  
x = 5  
y = 10  
if x > 0 and y < 20:  
    print("Ambas condiciones son verdaderas.")
```

## Operador or (O lógico)

El operador **or** devuelve **True** si al menos una de las expresiones que conecta es verdadera. En otras palabras, la expresión compuesta es verdadera si al menos una de las condiciones individuales es verdadera.

```
python Copy code  
  
x = 5  
y = 25  
if x == 5 or y == 10:  
    print("Al menos una de las condiciones es verdadera.")
```

## Operador not (negación lógica):

El operador **not** se utiliza para invertir el valor de una expresión booleana. Si la expresión original es verdadera, **not** la convierte en falsa, y viceversa.

```
python Copy code  
  
x = 5  
if not x == 10:  
    print("La condición es falsa.")
```

## Tabla de verdad

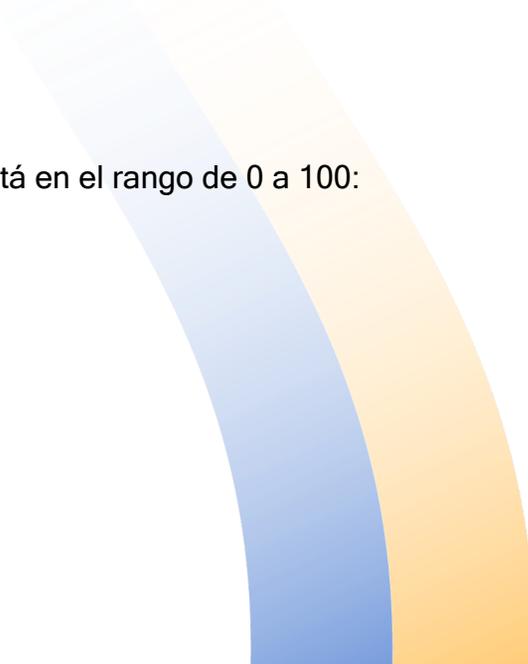
Los operadores lógicos pueden representarse mediante tablas de verdad que muestran todas las combinaciones posibles de valores de entrada y su resultado correspondiente.

Por ejemplo:

```
r Copy code  
  
x   y   x and y   x or y  
-----  
T   T       T       T  
T   F       F       T  
F   T       F       T  
F   F       F       F
```

## Ejemplo práctico

Supongamos que queremos verificar si un número es par y está en el rango de 0 a 100:



```
python Copy code  
  
num = 36  
if num % 2 == 0 and num >= 0 and num <= 100:  
    print("El número es par y está en el rango de 0 a 100.")
```

## Conclusiones:

Los operadores lógicos son herramientas poderosas en programación para combinar expresiones booleanas y controlar el flujo del programa. Al comprender cómo funcionan los operadores **and**, **or** y **not**, los programadores pueden escribir condiciones más complejas y lógicamente sólidas en sus programas.

## Referencias:

- Gaddis, T. (2019). Starting Out with Python. Pearson.
- Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. Green Tea Press.
- Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press.

