



OFICIOS DIGITALES

PROGRAMACIÓN PYTHON

Operadores



LIC. GUILLERMO ALFARO



Unidad 2

Operadores

Un operador es un símbolo que realiza una operación.

Dependiendo del lenguaje, existen diferentes tipos de operadores que se pueden agrupar de la siguiente manera:

- Operadores aritméticos
- Operadores relacionales
- Operadores de asignación
- Operadores lógicos
- Operadores de pertenencia
- Operador binario

1 – Operadores aritméticos

Estos operadores los vimos en detalle cuando estudiamos los tipos de datos numéricos.

```
a = 6  
print("a > 10: {}".format(a > 10))
```

```
a > 10: False
```

2 – Operadores relacionales

Este tipo de operadores sirven para relacionar dos variables o para ser más precisos, analizar la condición de veracidad de una determinada relación, y nos retornan una respuesta o variable del tipo booleana.

Veamos un ejemplo simple para determinar si un valor dado es mayor que 10:



Al igual que los operadores aritméticos, son de uso muy intuitivo. El siguiente cuadro contiene un ejemplo de uso de cada uno:

Símbolo	Significado	Ejemplo	Resultado
<code>==</code>	Igual que	<code>5 == 7</code>	<code>False</code>
<code>!=</code>	Distinto que	<code>rojo != verde</code>	<code>True</code>
<code><</code>	Menor que	<code>8 < 12</code>	<code>True</code>
<code>></code>	Mayor que	<code>12 > 7</code>	<code>True</code>
<code><=</code>	Menor o igual que	<code>12 <= 12</code>	<code>True</code>
<code>>=</code>	Mayor o igual que	<code>4 >= 5</code>	<code>False</code>

> Las variables booleanas se representan numéricamente como 0 (falso) y 1 (verdadero).

3 – Operadores de asignación

Son operadores que se utilizan para asignar o modificar una variable.

Ya vimos que el operador de asignación es el símbolo `=`, veamos algunas variantes de su uso para modificar variables que ya tienen valores asignados.

```
# Suma y asignación.  
var = 5  
var += 2  
print("Suma y asignación: {}".format(var))
```

Suma y asignación: 7

```
# Resta y asignación.  
var = 5  
var -= 2  
print("Resta y asignación: {}".format(var))
```

Resta y asignación: 3

> Probar este método con todos los operadores aritméticos.



4 – Operadores lógicos

Los operadores lógicos booleanos, nos permiten armar una lógica más compleja a nuestros programas, como, por ejemplo, anidar o concatenar varios operadores relacionales para obtener un único resultado booleano sobre una proposición compleja que contiene más de un operador relacional.

Tenemos tres operadores de este tipo para usar en Python, veamos primero los dos que nos permiten concatenar dos expresiones relaciones, AND y OR.

Operador	Ejemplo	Explicación	Resultado
and	5 == 7 and 7 < 12	False and False	False
and	9 < 12 and 12 > 7	True and True	True
and	9 < 12 and 12 > 15	True and False	False
or	12 == 12 or 15 < 7	True or False	True
or	7 > 5 or 9 < 12	True or True	True

Y un par de ejemplos para ver cómo funcionan.

```
# AND
a=5
b=10
print("AND es V solo si ambas expresiones son verdaderas: {}".format(a==5 and b==10))
```

AND es V solo si ambas expresiones son verdaderas: True

```
# OR
a=5
b=10
print("OR es V solo si una de las expresiones es verdadera: {}".format(a==7 or b==10))
```

OR es V solo si una de las expresiones es verdadera: True

Adicionalmente tenemos el operador de negación NOT:

```
# NOT, negación
a = 2
print("NOT modifica el valor booleanos de una expresión: {}".format(not a == 2))
```

NOT modifica el valor booleanos de una expresión: False



5 – Operadores de pertenencia.

Estos operadores se utilizan para determinar la pertenencia o no de un elemento en una lista o secuencia de elementos.

Son los operadores IN y NOT IN.

```
lista = [1,2,3]
print(1 in lista)
```

True

```
lista = [1,2,3]
print(8 in lista)
```

False

Ya vimos que en Python las cadenas de texto se almacenan como listas, por lo tanto, estos operadores también son aplicables a dichas cadenas de texto.

```
lista = [1,2,3]
print(1 in lista)
```

True

```
lista = [1,2,3]
print(8 in lista)
```

False

> Probar estos operadores en una cadena de texto.

6 – Operadores binarios

Estos operadores se utilizan para hacer operaciones de BITS, es decir transforman las variables a números binarios, y luego realizan la operación bit a bit.