

**Curso 2° SMR**  
**Módulo: SOR**  
**Sesión 4: Valores "lógicos"**

**¿Qué pretendemos conseguir?**

Operadores and, or, not.

Operadores ==, !=, <, >, >=, <=

**Desarrollo de la sesión**

Los posibles valores booleanos (lógicos) son `True` (verdadero) y `False` (falso).

El operador `and` devuelve verdadero si el valor booleano de la derecha y el de la izquierda son verdaderos. Ejemplo

**`(A > 7 ) and (A < 15)`**

Esta expresión será `True` si A tiene un valor comprendido entre 8 y 14.

El operador `or` devuelve `False` si el valor booleano de la derecha y el de la izquierda son falsos.

**`(A > 7) or (A < 15)`**

En este caso la expresión será verdadera siempre que A sea menor que 15 o mayor que 7. Intenta encontrar algún caso en el que la expresión sea `False`.

El operador `not` devuelve verdadero si el valor situado a su derecha es falso, y falso si el valor situado a su derecha es verdadero.

**`not (A > 15)`**

Será verdadero cuando A sea menor que 15 y falso cuando sea 15 o superior.

El operador `==` devuelve verdadero cuando los valores de izquierda y derecha son iguales. El operador `!=` devuelve verdadero cuando los valores izquierda derecha son diferentes. El operador `>` devuelve verdadero cuando el operador de la izquierda es mayor que el de la derecha. Una vez entendido esto el resto de operadores (`>=`, `<`, `>=`) creo que se explican por si solos.

Un error muy común, sobre todo cuando se está empezando a programar, es equivocarse entre los operadores de asignación(=) y de igualdad (==). Tenlo en cuenta cuando estés haciendo un programa.

## Comenzamos

Ejecutamos el comando python.

Introduce

A=15

B=30

Ejercicio1: ¿Cuál es el resultado de (A>15) or (B==30)

Ejercicio2: ¿Y si cambiamos el or por un and?

Ejercicio3: Si hacemos C=17 ¿Cuál es el resultado de not  
( ( A > C ) and ( B > C ) or True).

Ejercicio4: ¿Aparece algún error si introducimos en Python  
D==7? ¿Por qué?