
 No description has been provided for this image No description has been provided for this image

Python Basics I

Contenidos

- [Operaciones con booleanos](#)

Operaciones con booleanos

[al índice](#)

Todas las operaciones que realizabamos en el apartado anterior devolvían un tipo de dato concreto: un booleano. `True` o `False`. Pero ¿cómo harías si se tienen que cumplir 3 condiciones, o solo una de esas tres, o que no se cumplan 5 condiciones? Para este tipo de operaciones recurrimos al [Álgebra de Boole](#). Se trata de una rama del álgebra que se utiliza en electrónica, pero que tiene un sin fin de aplicaciones, no solo técnicas, sino aplicables a la vida cotidiana. Estas matemáticas pueden llegar a ser muy complejas aún utilizando únicamente dos valores: `True` y `False`. Las operaciones más comunes son **AND**, **OR**, **NOR**. En las siguientes tablas tienes todos los posibles resultados de las puertas AND, OR, NOR, dependiendo de sus inputs.

 imagen

Puede parecer complejo pero a efectos prácticos, y sin meternos con otro tipo de puertas lógicas, te recomiendo seguir estas reglas:

- **AND**: Se tienen que cumplir ambas condiciones para que sea un `True`
- **OR**: Basta que se cumpla al menos una condicion para que sea `True`
- **NOR**: Lo contrario de lo que haya

Veamos un ejemplo práctico para aclarar estos conceptos. Imaginemos que queremos comprar un ordenador, pero nos cuesta decidirnos. Eso sí, tenemos claras las siguientes condiciones a la hora de elegir

- La RAM me vale que tenga 16, 32 o 64 GB

- En cuanto al procesador y disco duro, la combinación que mejor me viene es un i3 con 500GB de disco.
- Precio: que no pase de los 800 €

```
In [1]: # Primer ordenador
ram1 = 32
process1 = "i5"
disco1 = 500
precio1 = 850

# Segundo ordenador
ram2 = 8
process2 = "i5"
disco2 = 500
precio2 = 600

# Tercer ordenador
ram3 = 32
process3 = "i3"
disco3 = 500
precio3 = 780
```

Veamos cómo implemento esto mediante operaciones booleanas

```
In [ ]: # Primero, calculamos el valor de estas condiciones por separado
# para el primer ordenador
```

```
In [4]: cond_ram1 = (ram1 == 16 or ram1 == 32 or ram1 == 64) #OR
cond_process1 = (process1 == "i3" and disco1 == 500) #and obliga a toda
cond_precio1 = (precio1 < 800)
cond_total1 = cond_ram1 and cond_process1 and cond_precio1
print(cond_total1, cond_ram1, cond_process1, cond_precio1)
```

False True False False

El primer ordenador cumple el requisito de ram, pero no los de precio y procesador/disco. Veamos los otros dos si los cumplen

```
In [ ]: # Calculamos las condiciones para los otros ordenadores
```

```
In [3]: cond_ram2 = (ram2 == 16 or ram2 == 32 or ram2 == 64) #OR
cond_process2 = (process2 == "i3" and disco2 == 500) #and obliga a toda
cond_precio2 = (precio2 < 800)
cond_total2 = cond_ram2 and cond_process2 and cond_precio2
print(cond_total2, cond_ram2, cond_process2, cond_precio2)
```

False False False True

```
In [5]: cond_ram3 = (ram3 == 16 or ram3 == 32 or ram3 == 64) #OR
cond_process3 = (process3 == "i3" and disco3 == 500) #and obliga a toda
cond_precio3 = (precio3 < 800)
```

```
cond_total3 = cond_ram3 and cond_process3 and cond_precio3  
print(cond_total3, cond_ram3, cond_process3, cond_precio3)
```

True True True True

¡Bingo! El tercer ordenador cumple todas las condiciones para ser mi futura compra. Verás en próximos notebooks que esto se puede hacer todavía más sencillo mediante bucles y funciones.

Si quieres aprender más sobre el **Álgebra de Boole**, te recomiendo [esta página](#)

In []:

In []: