

Operadores encadenados

Una de las peculiaridades más interesantes de Python, y que otros lenguajes no ofrecen, es la capacidad de encadenar múltiples expresiones.

Normalmente para encadenar expresiones las unimos utilizando el operador lógico **and**:

```
1 < 2 and 2 < 3

True
```

La condición para poder encadenar con operadores es encontrar un punto en común entre ambas expresiones:

```
1 < 2 < 3

True
```

La cual también se puede expresar de la siguiente forma:

```
3 > 2 > 1

True
```

RECORDATORIO: Al utilizar operadores encadenados, estos se basan en comprobar el resultado de cada mínima expresión y relacionarlos con AND. No confundir con la comparación del resultado de cada expresión con el siguiente:

Lógica correcta	Lógica incorrecta
3 > 2 > 1	3 > 2 > 1
(3 > 2) and (2 > 1)	(3 > 2) > 1
True and True	True > 1
True	False

Veamos en ejemplo mucho más útil, donde queremos comprobar si un número se encuentra entre 0 y 100 (ambos incluidos):

```
numero = 35
if numero >= 0 and numero <= 100:
    print("El número {} se encuentra entre 0 y 100".format(numero) )
```

```
else:  
    print("El número {} no se encuentra entre 0 y 100".format(numero) )
```

El número 35 se encuentra entre 0 y 100

Utilizando operadores encadenados podemos simplificar la sintaxis readaptando la lógica:

```
numero = 35  
if 0 <= numero <= 100:  
    print("El número {} se encuentra entre 0 y 100".format(numero) )  
else:  
    print("El número {} no se encuentra entre 0 y 100".format(numero) )
```

El número 35 se encuentra entre 0 y 100