

# Lógica Booleana

Toda la informática se basa en trabajar con el sistema binario y la lógica booleana. Los valores de la lógica binaria "Verdadero" y "Falso" en Python existen como `True` y `False` respectivamente.

```
True
True
False
False
type(True)
bool
```

## Operadores Booleanos

### Negación (`not`)

La negación devuelve `False` cuando es aplicada sobre un `True` y viceversa. Es un operador unario. Es decir, sólo necesita un operando para devolver un resultado.

```
not False
True
not True
False
```

### Suma lógica (`or`)

La suma lógica devuelve `True` siempre que alguno de sus operandos sea `True`. O lo que es lo mismo, devuelve `False` cuando ambos operando son `False`

```
True or False
True
True or True
True
False or True
True
```

```
False or False
```

```
False
```

## Producto lógico (and)

El producto lógico devuelve `True` sólo si ambos operando son `True`. O lo que es lo mismo, si alguno de sus operandos es `False` el producto lógico devuelve `False`

```
True and True
```

```
True
```

```
True and False
```

```
False
```

```
False and False
```

```
False
```

```
False and True
```

```
False
```

## Operadores Relacionales

Evalúan la relación entre los operandos y devuelven `True` o `False` según corresponda. Son 6:

1. Igual: `==`
2. Mayor: `>`
3. Menor: `<`
4. No igual: `!=`
5. Mayor o igual: `>=`
6. Menor o igual: `<=`

Guardemos en la variable "a" el valor 10 y veamos algunos ejemplos:

```
a == 100
```

```
False
```

```
a > 1
```

```
True
```

```
a < 9
```

```
False
```

```
a != 5
```

```
True
```

```
a >= 10
```

```
True
```

```
a <= 10
```

```
True
```

Para todos los operadores se pueden usar expresiones y combinaciones entre ellos.

¿Qué devuelve la siguiente expresión?

```
a = 10
```

```
b = 2
```

```
flag = False
```

```
(a * b <= a + 2) and flag
```

```
False
```