# Operadores lógicos

Como comparar valores booleanos.

# Operadores booleanos

| | |
|---|---|
| **OU** | \|\| |
| **E** | && |
| **Não** | ! |

Editor

```
1   false || false // false
2   true  || false // true
3   false || true  // true
4   true  || true  // true
5
```

O comparador OU (||) resulta em **true** quando **qualquer valor** for **verdadeiro**. Caso contrário, resulta em false.

Editor

```
1   false && false // false
2   true  && false // false
3   false && true  // false
4   true  && true  // true
5
```

O comparador E (&&) resulta em **true** apenas quando **todos os valores são verdadeiros**. Caso contrário, resulta em false.

# Comparadores lógicos

## OU (||)

| | | |
|---|---|---|
| F | F | F |
| T | F | T |
| F | T | T |
| T | T | T |

## E (&&)

| | | |
|---|---|---|
| F | F | F |
| T | F | F |
| F | T | F |
| T | T | T |

É possível utilizar **mais de um comparador na mesma expressão**?


Editor
1 false || false || true // true
2 false && false && true // false
3 false || true && true // true
4

# Tabela básica de precedência dos operadores lógicos

| | |
|---|---|
| **E** | && |
| **OU** | \|\| |


Editor
1 (false || true) && true // true
2

E o operador NOT!?

Editor

```
1  !false // true
2  !true  // false
3
4  !(10 < 20) // !true = false
5
```

Tudo que é true **se torna false** e vice versa.

Apesar de se parecerem complexos, os **operadores lógicos são muito especiais na programação**.

Editor

```
(1 < 2) || (3 > 4) // true || false = true
(1 > 2) || (3 > 4) // false || false = false

(10 < 20) && (20 < 30) // true && true = true
(20 < 30) && (30 > 40) // true && false = false

(20 < 30) && !(30 > 40) // true && !false = true

```

Editor

```
1  var a = 10
2  var b = 20
3  var c = 30
4
5  (a < b) && (b < c) // true && true = true
6
```