



# Operadores Lógicos



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



“

**Operadores lógicos** e de **comparação** nos ajudam a **controlar** certos aspectos de nossos aplicativos, além de torná-los mais **eficientes**.



”





## Comparação simples

Comparam dois valores, devolvem verdadeiro ou falso.

```
{}  
10 == 15 // Igualdade → false  
10 != 15 // Desigualdade → true
```

## Comparação estrita

Comparam o valor e o tipo de dado também.

```
{}  
10 === "10" // Igualdade estrita → false  
10 !== 15 // Desigualdade estrita → true
```

No primeiro caso, o valor é 10 em ambos os casos, mas os tipos de dados são número e string. Como estamos comparando que ambos (valor e tipo de dados) são iguais, o resultado é falso.



## Comparação (continuação)

Comparam dois valores, devolvem verdadeiro ou falso.

{}

```
15 > 15 // Maior que → false  
15 >= 15 // Maior ou igual a → true  
10 < 15 // Menor que → true  
10 <= 15 // Menor ou igual a → true
```



Sempre devemos escrever o símbolo maior (>) ou menor (<) antes do igual (>= ou <=). Se o fazemos ao contrário (=> ou =<), o JavaScript lê primeiro o operador de atribuição = e então não saberá o que fazer com o maior (>) ou o menor (<).



Os **operadores** de comparação sempre retornarão um **booleano**, ou seja, **true** ou **false**, como resultado.





## Lógicos

Permitem combinar valores booleanos, o resultado também retorna um booleano.

Existem três operadores **e** (and), **ou** (or), **negação** (not).

**AND** (&&) → **todos** os valores devem ser avaliados como **true (verdadeiro)** para que o resultado seja **verdadeiro**.

```
{ } (10 > 15) && (10 != 20) // false
```



```
{ } (12 % 4 == 0) && (12 != 24) // true
```





**OR ( || )** → **ao menos um** valor deve ser avaliado como **true** para que o resultado seja true.

```
{ } (10 > 15) || (10 != 20) // true
```



```
{ } (12 % 5 == 0) && (12 != 12) // false
```



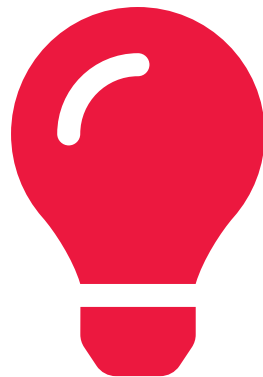
**NOT ( ! )** → **nega a condição**. Se era true, será false e vice-versa.

```
{ } !false // true  
! (20 > 15) // false
```



“

Assim como os operadores de comparação, os **operadores lógicos** sempre retornarão um **booleano**, ou seja, **true** ou **false**, como resultado.



”





DigitalHouse>  
Coding School