

# Operadores relacionais:

Que nome difícil! Mas, você verá que é fácil de entender!

Como vimos, a condicional `if` é uma estrutura que executa um código caso a condição seja verdadeira, dependendo sempre da condição requerida.

Já estamos familiarizados com o sinal de atribuição (`=`), que usamos regularmente na declaração de nossas variáveis, agora vamos entender mais sobre os operadores relacionais.

Os **operadores relacionais** são utilizados para comparar diversos tipos de dados, em que os valores comparados podem ser inseridos diretamente ou através de variáveis.

Geralmente retornam um valor booleano, neste caso: `true`(verdadeiro) ou `false`(falso):

Operador	Significado
<code>&gt;</code>	Maior
<code>&lt;</code>	Menor
<code>=</code>	Igual
<code>==</code>	Comparação de valor
<code>===</code>	Comparação de valor e tipo de dado
<code>&gt;=</code>	Maior ou igual
<code>&lt;=</code>	Menor ou igual

Mas, como esses operadores funcionam na prática? Um dos pontos que aprendemos é que o seu retorno será `true` ou `false`, mas isso não significa que o resultado de nossas comparações precisam ser necessariamente um deles.

Vamos ver os exemplos para compreender melhor:

```
var A = 20  
var B = 10
```

```
A > B    // o resultado da comparação é true (verdadeiro)  
  
A < B    // o resultado da comparação é false (falso)  
  
A >= B   // o resultado da comparação é true (verdadeiro)  
  
A <= B   // o resultado da comparação é false (falso)
```

No caso dos sinais que comparam igualdade, precisamos ter muita atenção. Nesse caso, os sinais são semelhantes, mas as avaliações são distintas.

Vamos ver alguns exemplos para esclarecer a diferença de cada sinal. Teremos duas variáveis:

```
var x = "1"  
var y = 1
```

### Sinal de atribuição (=)

Neste caso, estamos apenas atribuindo um valor a variável x:

```
x = y
```

### Sinal de comparação de valor (==)

Neste caso, apenas o valor da variável que é levado em conta independente do tipo de dado:

```
x == y // "1" == 1 o retorno é true (verdadeiro)
```

## Sinal de comparação de valor e tipo de dado (===)

Neste caso, levamos em consideração o valor da variável levando em conta o tipo do dado e o valor do mesmo

```
x === y // "1" === 1 o retorno é false (falso)
```