

OPERADORES LÓGICOS

“

Os **operadores** nos permitem **manipular o valor** das variáveis, realizar **operações** e **comparar** seus valores



DE ATRIBUIÇÃO

Atribui o valor da direita na variável da esquerda.

```
{ }
```

```
var idade = 35; // Atribuo o número 35 à idade
```

ARITMÉTICOS

Nos permitem fazer operações matemáticas, retornando o resultado da operação.

```
{ }
```

```
10 + 15 // Soma → 25
```

```
10 - 15 // Subtração → -5
```

```
10 * 15 // Multiplicação → 150
```

```
15 / 10 // Divisão → 1.5
```

ARITMÉTICOS (CONTINUAÇÃO)

{ }

`15++` // Aumento, é igual a $15 + 1 \rightarrow 16$

`15--` // Diminuição, é igual a $15 - 1 \rightarrow 14$

{ }

`15 % 5` // Módulo, o resto de dividir 15 entre 5 $\rightarrow 0$

`15 % 2` // Módulo, o resto de dividir 15 entre 2 $\rightarrow 1$

O operador de módulo (%) nos retorna o resto de uma divisão.

$$\begin{array}{r} 15 \overline{) 5} \\ \underline{0} \\ 3 \end{array}$$

$$\begin{array}{r} 15 \overline{) 2} \\ \underline{1} \\ 7 \end{array}$$

“

Os **operadores** aritméticos sempre **retornarão** o **resultado numérico** da **operação** que se está realizando.



DE COMPARAÇÃO SIMPLES

Compara dois valores, retornando verdadeiro ou falso.

{ }

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

DE COMPARAÇÃO ESTRITA

Compara 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 casos, mas os tipos de dados são number e string. Como estamos pedindo que ambas coisas sejam iguais, o resultado é **false**.

DE COMPARAÇÃO (CONTINUAÇÃO)

Compara dois valores, retornando verdadeiro ou falso.

{ }

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



Sempre devemos escrever o símbolo maior (>) o menor (<) antes do igual (>= o <=). Se fizermos o contrário (=> o =<) JavaScript lê primeiro o operador de atribuição = e então ele não sabe 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, e o resultado também retorna um booleano.

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

AND (&&) **todos** os valores devem avaliar como **true**.

{ }

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



{ }

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



LÓGICOS (CONTINUAÇÃO)

OR (||) ao menos um valor deve avaliar como **true**.

{ }

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



{ }

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



NOT (!) nega a condição, se era true, será false e viceversa

{ }

```
!false // true
```

```
!(10 > 15) // true
```

“

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



CONCATENAÇÃO

Utilizamos para unir strings. Devolve outra cadeia de texto.

{}

```
var nome = 'Teodoro';  
var sobrenome = 'García';  
var nomeCompleto = nome + ' ' + sobrenome;
```

Se misturarmos outros tipos de dados, eles serão convertidos em cadeias de texto.

{}

```
var linha = 'M';  
var assento = 7;  
var localização = linha + assento; // 'M7' como  
string
```