**Operadores**

**Operadores Aritméticos**

O símbolo de + e menos você já sabe o que faz, o \* multiplica valores e a / faz a divisão comum.

Para fazer a divisão inteira se usa %, e o computador vai te mostrar somente o valor que sobra. Por exemplo:

5 % 2 = 2 e sobra 1

Para fazer a potência, se usa \*\*.

­­­­­­­­­

**Ordem de precedência**

Em Java Script, as contas seguem uma ordem de qual conta é feita primeira. É a seguinte:

1° ( ) - É a primeira coisa que o programa roda, independente da conta feita dentro dela.

2° (\*\*) em 3° (/, % e \*) e por último (+ e -).

**Operadores de atribuição**

Nas atribuições, você coloca o resultado das contas como variáveis. Por exemplo:

Var R = 2 + 2 (a variável R agora vale 2)

**Auto atribuição**

Você pode usar os operadores junto do = quando a variável da conta se torna também o resultado. Por exemplo:

Var N = 4

Agora N é igual a 4

N = n + 5

A partir de agora N foi atualizado, passando a ser igual a 9 pois 4 + 5 é 9

Usando com foi dito no início, a mesma conta pode ser escrita desta forma:

N += 5

**­­­­­­­** Com outros operadores: -=, \*=, /=, %= e \*\*=

decrementos

Os decrementos são usados para simplificar ainda mais seu código, somando ou diminuindo 1 ao valor original. São eles ++ e --. Ex: var x = 10 x++ ou ++x é igual a 11

**Operadores Relacionais**

Os operadores relacionais são comparativos, ou seja, vão dizer se a expressão é verdadeira ou falsa. Por exemplo:

7 < 4. Resultado false.

Na expressão acima, a expressão perguntou se 7 era menor que 4 e logicamente, o resultado é false afinal 7 não é menor que quatro, é maior.

No total existem 7 operadores. São eles:

**Operador de identidade.**

O operador de identidade (===) vai dizer se um símbolo é idêntico a outro. Ex:

5 === ‘5’

Resultado False

Isso por que o 5(Número) não é igual ao ‘5’ (String)

* < (menor que);
* > (maior que);
* <= (menor ou igual);
* >= (maior ou igual);
* == (igual);
* !> (diferente).

O resultado das expressões com operadores relacionais será sempre booleano (True ou False).

**Operadores lógicos**

São eles:

* ! (negação);
* && (conjunção);
* || (disjunção);

Nas linhas de precedência eles são os penúltimos a serem considerados, e servem para comparar valores True e false.

**!-** a negação serve para negar algum valor, portanto o valor pode ser qualquer um, exceto o valor indicado. (!5 irá aceitar qualquer valor, menos o 5).

**Op. Ternário**

É somente ? :

Neste operador, usamos os dois símbolos para mostrar uma variável

Por exemplo:

Idade >= 18 ? ‘maior’ : ‘menor’

Variável vai se tornar um valor booleano, e vai ser executado,

Se seu resultado for True, vai ser mostrado na tela o texto antes dos dois pontos, E se for false, vai mostrar o segundo.

**&&-** A conjunção irá mostrar o booleano True somente se todos os valores pedidos forem entregues. Ex: ‘5 && 7’ O valor será True se 5 **E** 7 forem dados.

**||-** Por fim, a disjunção irá mostrar o booleano True somente se qualquer um dos valores pedidos forem entregues. Ex: ‘2 || 9’ O valor será True se 2 **OU** 9 forem dados.

**DOM (Document object model)**