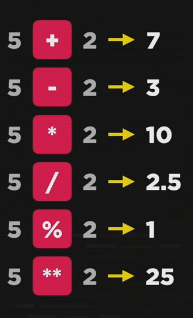
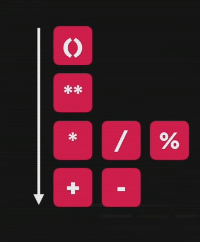
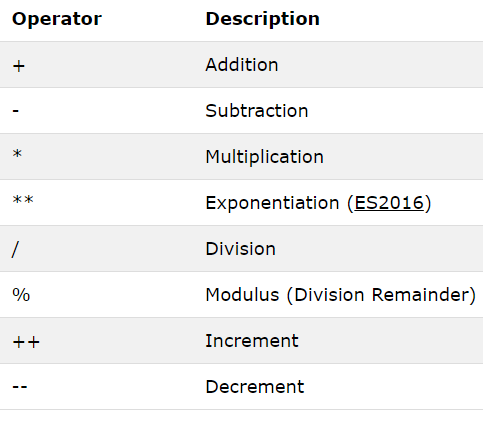
**Operadores do JavaScript**

**Operadores Aritméticos**



**Nota:** % é o resto da divisão como todas as outras linguagens.

**Nota 2:** \*\* é exponenciação

**Atribuição de valores no JavaScript**

Existem dos tipos de atribuições para Operadores Aritméticos:

* **Atribuição simples(=): indica que a variável recebe o valor.**

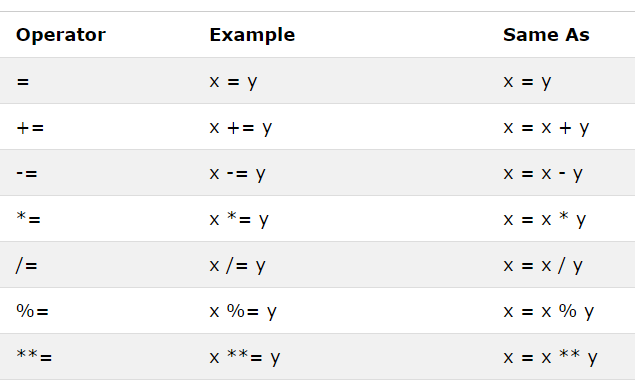
**Ex.: var a = 8+1 -> a recebe 8+1**

* **Auto atribuição:** quando a variável recebe si mesma;
  + **Forma normal**

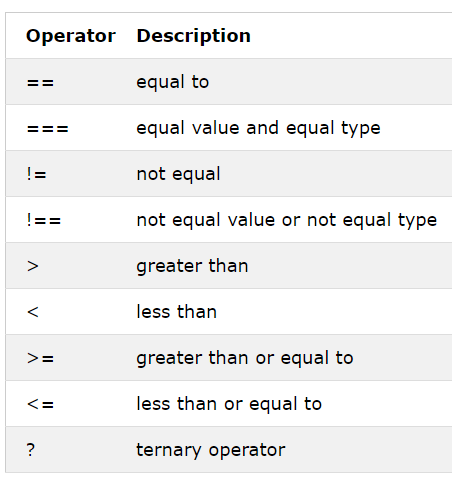
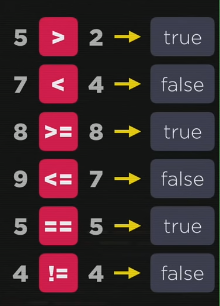
**Ex.: var a = a + 1 -> indica que ‘a recebe a + 1’;**

* + **Forma simplificada**

**Ex.: var a += 1 -> mesma coisa que dizer que ‘a recebe a + 1’**

**Alguns casos comuns:**

* **Pós-incremento:** mostra variável atual primeiro e soma/subtrai depois.
* **a += 1 pode ser escrito como a++**
* **a -= 1 pode ser escrito como a--**
* **Pré-incremento:** soma/subtrai primeiro e mostra a variável depois.
* **a += 1 pode ser escrito como ++a**
* **a -= 1 pode ser escrito como - - a**

**Operadores Relacionais**

**Nota:** se o número estiver no formato em string, o javascript considera como número.

Ex.: ***5 == ‘5’ -> true***

**Operador de identidade** (===): verifica se os valores são do mesmo valor e do mesmo valor.

Ex.: ***5===’5’ -> false***

**Operadores Ternários****:** chama-se ‘ternário’ porque usa três operandos.

* Teste -> um teste lógico qualquer
* Depois da **?** -> o que vai acontecer **caso o valor seja verdadeiro**;
* Depois do **:** -> indica o que vai acontecer **caso o valor seja falso**.

**Ex.:**

***Var a = 10***

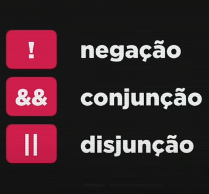
***a >= 10 ? ’sim’ : ’não’ -> return ‘sim’***

Se **a** for **maior ou igual a 10**, retorna **‘sim’**, caso contrário, retorna **‘não’**

***Ex 2.:*** é possível armazenar esse resultado dentro de uma variável.

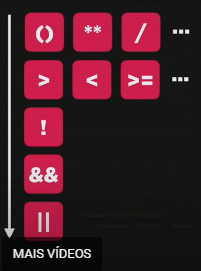


Caso a **variável dist** seja menor ou igual a 200, a **variável custo** recebe (dist x 0,5); caso contrário, recebe (dist x 0,45)

**Operadores Lógicos**

* **Negação seria not.**
* **Conjunção seria and.**
* **Disjunção seria or.**

**Ordem de Precedência dos Operadores**

* **Primeiro: operadores aritméticos;**
* **Segundo: operadores relacionais;**
* **Terceiro: operadores lógicos com a seguinte ordem de precedência**
  + **Not(!)**
  + **And(&&)**
  + **Or(||)**