# aula 8 – operadores (parte 2)

Breve revisão:

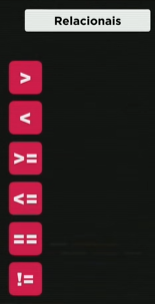
Pra que serve o operador % no JS? Serve para calcular uma divisão e exibir o resto da divisão calculada.

6 + 4 / 2 é igual a 4 ou 8? 8

Se uma variável n está valendo 10 e executamos um n += 5, qual será seu novo valor? O valor será 15, pois quando usamos o += estamos somando com o valor da própria variável.

O que acontece quando é colocado mais ++ e -- em uma variável? Vamos supor que var n = 5, e depois fazemos o seguinte comando, n ++, a variável n irá valer 6 pois esse comando adiciona mais um ao valor da variável, se fizermos o inverso, n --, isso vai subtrair um da variável.

• operadores relacionais

 Os operadores relacionais do JS são como na maioria das linguagens de programação:

• > maior que ...

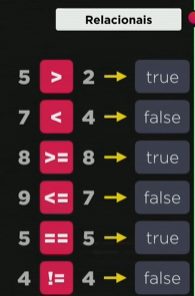
• < menor que ...

• >= maior ou igual à ...

• <= menor ou igual à ...

• == igual à ...

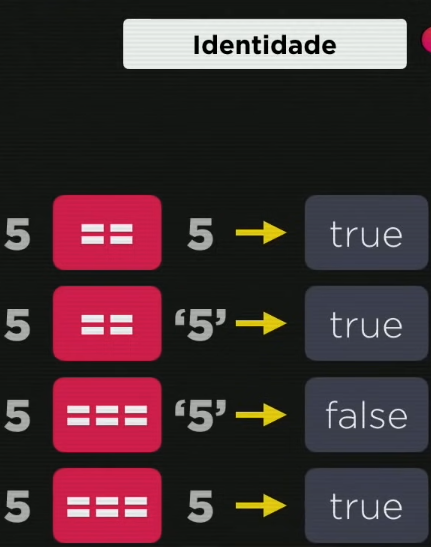
• != diferente de ...



Sempre que for feita uma expressão usando operadores relacionais, seu resultado será um valor **Booleano!**

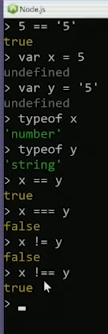
Alguns exemplos de como cada operador funciona:

Também é possível comparar valores com os operadores de identidade:



• Repare que no segundo exemplo, o número está em valor string, mas pro JS o sinal de == reconhece apenas o valor independente se é string ou não;

• No terceiro exemplo vemos que o sinal de === diferencia o valor numérico e string, e mostra que não é igual pq ele compara as restrições de cada valor



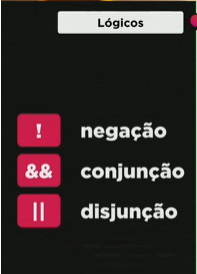
Alguns exemplos no node.js:

• Observe que quando comparamos

x != y, estamos pedindo se os valores são diferentes, mas o valor é o mesmo, e quando digitamos x !== y estamos comparando as restrições que aí sim são diferentes!

• Operadores lógicos

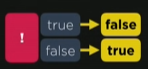
São apenas três operadores:



• Como funcionam?

Usando os exemplos do prof. Guanabara, vamos supor que vamos pedir canetas com os operadores de lógicos:

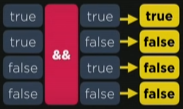
• Negação (!): Eu quero uma caneta, mas ela não pode ser azul, pode ser qualquer outra cor menos azul (bloqueia uma coisa obrigatoriamente). O operador de negação é um operador unário, ou seja, ele só tem um operando que fica depois da exclamação, que ou vai ser true e ou vai ser false.



Se não é true, é false;

Se não é false, é true;

• Conjunção (&&): Eu quero uma caneta azul e uma vermelha, mas tem que ser as duas! (aceita duas coisas obrigatoriamente). O operador de conjunção é um operador binário, ou seja, temos dois valores de cada lado dos &&.



• Disjunção (||): Eu quero uma caneta azul ou uma vermelha, mas se quiser me entregar as duas? Aceito. Se quiser me entregar só a azul? Aceito. Se quiser me entregar só a vermelha? Aceito. Mas se não me der nenhuma? Aí não aceito. (Ou uma coisa, ou outra, ou as duas, nunca nenhuma). Tem dois operadores lógicos, e basta que um valor seja verdadeiro para que ele “aceite”.



• mas e quando temos operadores aritméticos, operacionais e lógicos na mesma expressão?

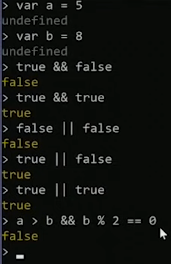
A ordem de precedência fica assim:

1° - Aritméticos

2° - Relacionais

3° - Lógicos

Exemplo:



• b % 2 é igual a 0, então é true;

• 5 não é maior que 8 (a > b), o que significa que ele é false;

• como temos um valor true e false (true && false), com o operador de conjunção, o resultado dessa expressão será false, pois os dois valores teriam que ser true para que o operador de conjunção “aceitasse”

• e se tivermos negação, conjunção e disjunção na mesma expressão?

A ordem de precedência fica:

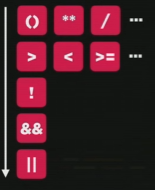
1° - Negação;

2° - Conjunção;

3° - Disjunção;

(primeiro o “não”, depois o “e”, e por último o “ou”)

• ordem de precencia com todos os operadores!!!

1° – Aritméticos (de acordo com a precedência dos aritméticos)

2° – Relacionais (sem ordem de precedência, apenas da esquerda pra direta como se lê)

3 °– Lógicos (primeiro negação, depois conjunção e por ultimo disjunção)

• Ternários

São apenas dois operadores, dentro de uma expressão:



Ele se chama ternário pois tem três partes dentro de uma expressão, junta três operando dentro de uma expressão:



Na pratica, ficaria como no exemplo abaixo:



Se a média for maior ou igual a 7.0, o resultado será aprovado(true), se for menor o resultado será reprovado(false).