JavaScript

Para facilitar seu aprendizado, tente fazer os exercícios mentalmente e somente em caso de dúvidas utilize alguma ferramenta, como o terminal de comando com o Node.js rodando, Console do navegador ou sites como JS Fiddle:

# Operadores

|  |  |
| --- | --- |
| **Operador** | **Função** |
| Aritméticos: retornam o resultado de uma operação | |
| +  \*  /  %  ++  -- | somar  subtrair multiplicar  dividir  resto de divisão incremento decremento |
| Comparadores matemáticos: teste lógico, retorno booleano (true / false) | |
| <  >  <=  >= | menor que maior que menor ou igual maior ou igual |
| Comparadores Lógicos: teste lógico, retorno booleano (true / false) | |
| ==  !=  ===  !=== | igualdade entre sentenças (valor) diferença entre sentenças (valor) igualdade entre sentenças (valor e tipo) diferença entre sentenças (valor e tipo) |
| Operadores de lógica e junção lógica | |
| !  &&  || | NÃO (NOT)  E (AND)  OU (OR) |

O sinal de exclamação (!) é o operador NOT (não), utilizado para negar a sentença que vem na sequência.

Exemplos:

a != b // o valor de a é diferente de b x !=== y // o valor e o tipo de x são diferentes de y !a == b // o valor de a não é igual ao valor de b

As condições lógicas são convertidas em números binários:

**true** é equivalente a 1 **false** é equivalente a 0

Operador lógico de atribuição

Tem a capacidade de atribuir valor a uma variável a partir de uma condição lógica, economiza IFs

Exemplo:

var meuCarro = cor == “preto” ? “preto” : “branco”;

se var meuCarro = corr == “preto” {

“”

Exercícios:

Preencha os resultados das operações e o tipo de dado

**Exemplos**

8 + 6 = 14 (number)

“8” + “6” = “86” (string)

“8.6” + 4 = “8.64” (string)

“8” \* 4 = 32 (number)

“8” - 4 = 4 (number)

“8” / 3 = 2.6666666666666665 (float)

5 + true = 6 (number)

“teste” + true = “testetrue” (string)

“8” == 8 = true (boolean)

“8” == 4 = false (boolean)

8 === “8” = false (boolean)

8 !== “8” = true (boolean)

8 < 4 = false (boolean)

8 > 4 = true (boolean)

Exercícios:

1. **Resolva as operações:**
   * 10 + 15 =35 (number)
   * “10” + 2 = 102 (string)
   * “10” \* 2 =20 (number)
   * “10” / 3 = 3.3333333 (float)
   * “10” % 3 = 1 (number)
   * 10 + true = 11 (number)
   * 10 == ”10” = true (boolean)
   * 10 === “10” = false (boolean)
   * 10 < 11 = 10 > 12 = false (boolean)
   * 10 <= 10.1 = true (boolean)
   * 10 > 9.99 = true (boolean)
   * 10 != “dez” = true (boolean)
   * 10 + true = (boolean)
   * “dez” + true = “dez1” (string)
   * 10 + false = 10 (number)
   * 10 \* false = 0 (number)
   * true + true = true (boolean)
   * 10++ = 11 (number)
   * 10-- = 8 (number)
   * 1 & 1 = 1 (boolean)
   * 1 & 0 = 0 (boolean)
   * 0 & 0 = 0 (boolean)
   * 1 & 0 = 0 (boolean)
   * 0 / 1 = 0 (number)
   * 5 + 5 == 10 = true (boolean)
   * “5” + ”5” == 10 = false (boolean)
   * “5” \* 2 > 9 = true (boolean)
   * (10 + 10) \* 2 = 40 (number)
   * 10 + 10 \* 2 = 30 (number)
2. Responda as perguntas de acordo com as variáveis.

var branco = “preto”; var preto = “cinza”; var cinza = “branco”; var carro = “preto”; var valor = 30000; var prestacao = 750;

1. branco == “branco” true
2. branco == cinza false
3. carro === branco false
4. var cavalo = carro == “preto” ? “cinza” : “marron”;

if (var carro = “preto”) {

System.out.println(“preto”);

} else {

System.out.println(“branco”);

}

“preto”

1. Quantas prestações são necessárias para pagar o valor do carro com uma entrada de 3.000? Demonstre a operação.

var restante = var valor - 3000

var quantidade = var restante/ var prestação

var restante = 30000-3000

var restante = 27000

var quantidade = 27000/750

var quantidade = 36

1. Somando as variáveis de cores é formada uma string de quantos caracteres?

“pretocinzabrancopreto”

21 caracteres