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”;

# 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:

## Resolva as operações:

* + 10 + 15 = 25 (number)
  + “10” + 2 = 1010 ( string)
  + “10” \* 2 = 20 (number)
  + “10” / 3 = 3.3 ( float)
  + “10” % 3 = 1 ( number)
  + 10 + true = 11(number)
  + 10 == ”10” = true ( boolean)
  + 10 === “10” = false (boolean)
  + 10 < 11 = false (boolean)
  + 10 > 12 = false (boolean)
  + 10 <= 10.1 = true(boolean)
  + 10 > 9.99 = true(boolean)
  + 10 != “dez” = 10!dez(string)
  + 10 + true = 11(number)
  + “dez” + true = deztrue (string)
  + 10 + false = 9 (number)
  + 10 \* false = -10 (number)
  + true + true = 2 (number)
  + 10++ = 12(number)
  + 10-- = 8 (number)
  + 1 & 1 = 1 (number)
  + 1 & 0 = 1 (number)
  + 0 & 0 = 0 (number)
  + 1 & 0 = 0 (number)
  + 0 / 1 = 0 (number)
  + 5 + 5 == 10 = sim (number)
  + “5” + ”5” == 10 = não (string)
  + “5” \* 2 > 9 = 55 ( string)
  + (10 + 10) \* 2 = 40 ( number)
  + 10 + 10 \* 2 = 30 (number)

1. 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” /( Não ) branco =preto
2. branco == cinza / ( não ) branco = preto
3. carro === branco / (não ) Carro = preto
4. var cavalo = carro == “preto” ? “cinza” : “marron” / (sim) carro = preto
5. Quantas prestações são necessárias para pagar o valor do carro com uma entrada de 3.000? Demonstre a operação.

30.000 – 3000 = 27,000

27,000 / 750 = 36

36 Prestações

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

R: 4