Tipo 2 - Questões de preenchimento de lacuna

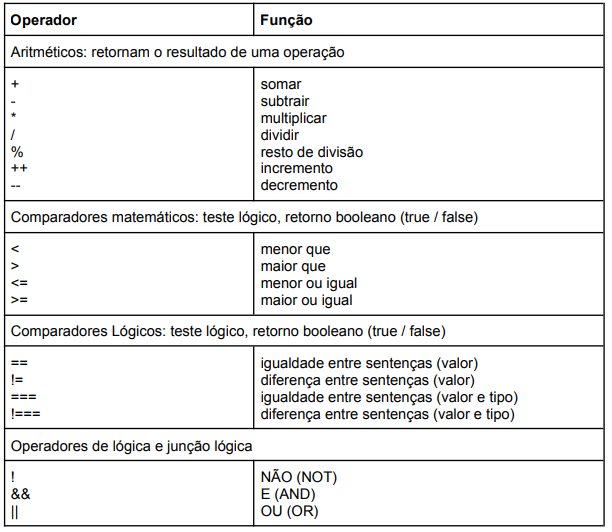
Tipo 1 - Quizzes

MÓDULO 14

JavaScript II

**Primeira parte**

Olá, estudante, 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:



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”;

As imagens demonstram as operações no Node. As operações foram feitas depois de responder as questões para confirmar os resultados.

**Questão 1**

1. Resolva as operações:

10 + 15 = 25 (number)

“10” + 2 = “102” (string)

“10” \* 2 = “1010” (string) **(20 – number)**

“10” / 3 = 3,333 (number)

“10” % 3 = 1 (number)

10 + true = 11 (number)

10 == ”10” = 1 (Boolean)

10 === “10” = 0 (Boolean)

10 < 11 = 1 (Boolean)

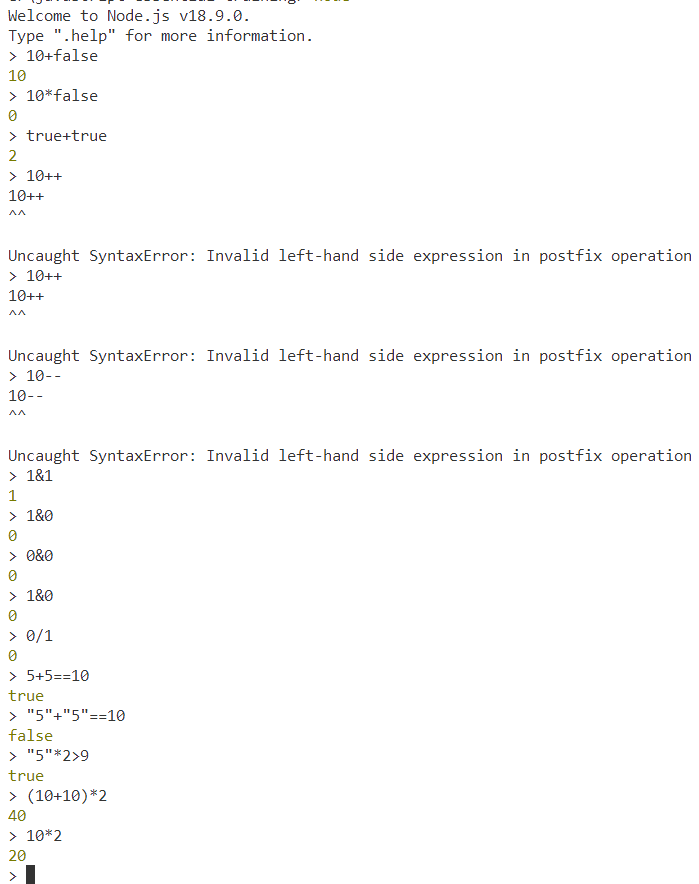
10 > 12 = 0 (Boolean)

10 <= 10.1 = 0 (Boolean)

10 > 9.99 = 1 (Boolean)

10 != “dez” = 1 (Boolean)

10 + true = 11 (number)

“dez” + true = “dez1” (string)

10 + false = 10 (number)

10 \* false = false (Number)

true + true = 2 (number)

10++ = 11 (number)

10-- = 9 (number)

1 & 1 = 1 (number)

1 & 0 = ?? (1 number)

0 & 0 = ?? (0 number)

1 & 0 = 0 (number)

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 \* 2 =20 (number)

RESPOSTAS:

10 + 15 = 25 (number)

“10” + 2 = '102' (string)

“10” \* 2 = 20 (number)

“10” / 3 = 3.3333333333333335 (float)

“10” % 3 = 1 (number)

10 + true = 11 (number)

10 == ”10” = true (boolean)

10 === “10” = false (boolean)

10 < 11 = true (boolean)

10 > 12 = false (boolean)

10 <= 10.1 = true (boolean)

10 > 9.99 = true (boolean)

10 != “dez” = true (boolean)

10 + true = 11 (number)

“dez” + true = 'deztrue' (string)

10 + false = 10 (number)

10 \* false = 0 (number)

true + true = 2 (number)

10++ = 11 (number)

10-- = 9 (number)

1 & 1 = 1 (number)

1 & 0 = 0 (number)

0 & 0 = 0 (number)

1 & 0 = 0 (number)

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)

**Feedback**

* Acima de 70%: Parabéns! Você acertou **86%** do exercício. Continue progredindo!
* De 50 a 70%: Quase lá! Você teve \_\_\_\_% de acertos. É uma boa pontuação, mas sabemos que você pode ter um melhor desempenho!
* Menos de 50%: Que pena! Você teve apenas\_\_\_\_\_% de acertos. Que tal rever a teoria para um melhor desempenho da próxima vez?

**Questão 2**

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;

a) branco == “branco” false

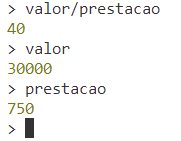
b) branco == cinza false

c) carro === branco true

d) var cavalo = carro == “preto” ? “cinza” : "marrom"; (“cinza”)

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

40 prestações



f) Somando as variáveis de cores é formada uma string de quantos caracteres?

21 caracteres



RESPOSTAS:

a) branco == “branco” (false)

b) branco == cinza (false)

c) carro === branco (false)

d) var cavalo = carro == “preto” ? “cinza” : “marrom”; ("cinza")

e) Quantas prestações são necessárias para pagar o valor do carro com uma entrada

de 3.000? Demonstre a operação.

var entrada = 3000

(valor - entrada) / prestacao = 36 prestações

f) Somando as variáveis de cores é formada uma string de quantos caracteres

21 caracteres

**Feedback**

* Acima de 70%: Parabéns! Você acertou **83%** do exercício. Continue progredindo!
* De 50 a 70%: Quase lá! Você teve \_\_\_\_% de acertos. É uma boa pontuação, mas sabemos que você pode ter um melhor desempenho!
* Menos de 50%: Que pena! Você teve apenas\_\_\_\_\_% de acertos. Que tal rever a teoria para um melhor desempenho da próxima vez?

**Tutoria**

Ainda ficou com dúvidas nas operações ou variáveis? Mande sua pergunta para nossos tutores.

**Quiz**

1. O operador lógico correspondente ao “não” é o:
2. &&
3. !
4. ||
5. !=

|  |
| --- |
| Feedback  O operador de lógica e junção lógica responsável pelo não é o &&. |

1. Quando definimos que a != b queremos dizer que:.
2. O valor e o tipo de x são diferentes de y.
3. O valor de a é diferente de b.
4. O valor de a não é igual ao valor de b.
5. Nenhuma das alternativas.

|  |
| --- |
| Feedback  O sinal de != indica diferença, então A é diferente de B. |

1. Quando utilizamos === queremos dizer que existe:
2. Igualdade entre sentenças (valor).
3. Diferença entre sentenças (valor).
4. Igualdade entre sentenças (valor e tipo).
5. Diferença entre sentenças (valor e tipo).

|  |
| --- |
| Feedback  O operador === indica a igualdade entre sentenças de valor e tipo. |