

Exercícios de Javascript

1. Elabore uma função "saudação" que ao ser chamada seja capaz de retornar uma saudação, concatenada com um argumento do tipo string e ponto de exclamação no final.

Ex: saudacao("Maria"); // Retornará "Olá Maria!"

2. Faça uma função capaz de receber um inteiro com a idade de uma pessoa em anos e faça a conversão dessa mesma idade para dias.

Ex: converteldadeParaDias(20); // Retornará 7305

3. Crie uma função que receba um número entre 1 e 7 e quando chamada retorne o dia da semana correspondente por extenso.

Ex: retornaDiaPorExtenso(1); // Retornará "Domingo"

4. No mercado do desenvolvimento de software é muito comum encontrar vagas internacionais que informam o salário anual em dólar. Sabendo disso, faça uma função capaz de converter um salário anual em dólar para um modelo mensal em reais.

Ex: converteSalario(40000); // Retornará 18333.33

5. Crie uma função que ao receber dois números irá retornar um booleano informando se o primeiro é maior ou igual ao segundo.

Ex: verificaMaiorOuIgual(1, 0); // Retornará true

6. Escreva uma função que seja capaz de receber um valor booleano ou numérico e retornar o oposto do mesmo. Mas se o parâmetro de entrada não for dos tipos acima a função deverá retornar uma mensagem informando o erro de entrada.

Ex: retornaOposto(1000); // Retornará -1000

Ex: retornaOposto(true); // Retornará false

Ex: retornaOposto("testando"); // Retornará "A função espera um parâmetro do tipo "bool" ou "number", porém o valor informado é do tipo "string".

7. Crie uma função que dado dois valores (passados como parâmetros) mostre no console a soma, subtração, multiplicação e divisão desses valores.

Ex: imprimeOperacoesMatematicasBasicas(5, 4); // Imprimirá no console

5 + 4 = 9

5 - 4 = 1

5 x 4 = 20

5 / 4 = 1.25

8. Crie uma função de potenciação que, assim como o método Math.Pow, deverá ser capaz de receber dois parâmetros (base, expoente) e retornará a base elevada ao expoente.

Ex: calculaPotencia(5, 2); // Retornará 25

9. Crie uma função capaz de retornar um booleano após verificar se o número informado é par.

Ex: ehPar(2) // Retornará true

Ex: ehPar(15) // Retornará false

10. Agora, utilizando a função feita acima, faça uma outra função capaz de encontrar todos os pares entre dois números informados.

Ex: retornaParesEntreValores(1, 10); // Retornará 2, 4, 6, 8

11. Faça uma função que receba os parâmetros de peso(kg) e altura(cm) e seja capaz de retornar o IMC (índice de massa corporal).

Ex: calculaIMC(60, 173); // Retornará 20.05

12. Faça uma função capaz de receber um vetor de inteiros, percorrê-lo através de uma estrutura de repetição e retornar o maior valor dessa lista.

Ex:

```
let listaQualquer = [10, -13, 55, 0, 1, 24, 132, -5000]
```

```
retornaMaiorValor(listaQualquer); // Retornará 132
```

13. Faça uma função capaz de retornar a soma de valores de uma lista numérica.

Ex:

```
let valores = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
retornaSoma(valores) // Retornará 66
```

14. Faça uma função que será capaz de retornar a média aritmética dos valores de um vetor de inteiros. Utilize a função “retornaSoma” para facilitar o seu desenvolvimento.

Ex:

```
let valores = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
retornaMedia(valores) // Retornará 6
```

15. Faça uma biblioteca para calculos geométricos que conterà as seguintes funções:

```
calculaAreaQuadrado(lado);
```

```
calculaAreaRetangulo(base, altura);
```

```
calculaAreaTriangulo(base, altura);
```

```
calculaComprimentoCircunferencia(raio);
```

```
calculaAreaCircunferencia(raio);
```

16. Crie uma função que receba um texto pelo parâmetro e conte a quantidade de palavras contidas no mesmo.

Ex: `contaPalavras("O rato roeu a roupa do rei")` // Retornará 7

17. Crie uma função capaz de receber um caractere e uma string e retorne a quantidade de vezes que este caractere aparece no texto informado.

Ex: `contaCaracteres("r", "O rato roeu a roupa do rei de roma")`; // Retornará 5

18. Implemente um algoritmo simples de “mecanismo de busca”, isto é, uma função que irá receber dois parâmetros: O primeiro será a palavra a ser pesquisada na lista, o segundo uma lista de strings. Caso a palavra pesquisada seja encontrada, a função deverá retornar `true`, do contrário retornará `false`.

Ex: `buscaPalavras('r', ['rato', 'roeu', 'roupa', 'rei', 'roma'])`; // Retornará `false`

Ex: `buscaPalavras('roma', ['rato', 'roeu', 'roupa', 'rei', 'roma'])`; // Retornará `true`