

Let, Const, Arrow function

1 - Uma fruteira está vendendo frutas com a seguinte tabela de preços:

	Até 5 Kg	Acima de 5 Kg
Morango	R\$ 2,50 por Kg	R\$ 2,20 por Kg
Maçã	R\$ 1,80 por Kg	R\$ 1,50 por Kg

Se o cliente comprar mais de 8 Kg em frutas ou o valor total da compra ultrapassar R\$ 25,00, receberá ainda um desconto de 10% sobre este total. Escreva um algoritmo em JavaScript para ler a quantidade (em Kg) de morangos e a quantidade (em Kg) de maçãs adquiridas e retornar no console, utilizando **Interpolação**, o valor a ser pago pelo cliente. Utilize const e/ou let nas declarações de variáveis.

2 - Escreva um algoritmo em JavaScript para ler 2 valores e se o segundo valor informado for ZERO, deve ser lido um novo valor, ou seja, para o segundo valor não pode ser aceito o valor zero e imprimir, utilizando **Interpolação**, o resultado da divisão do primeiro valor lido pelo segundo valor lido. (utilizar a estrutura Do while). Utilize const e/ou let nas declarações de variáveis

3 - Escreva um algoritmo em JavaScript para ler 10 valores e escrever quantos desses valores lidos estão no intervalo [10,20] (incluindo os valores 10 e 20 no intervalo) e quantos deles estão fora deste intervalo. Utilize const e/ou let nas declarações de variáveis

4 - A prefeitura de uma cidade deseja fazer uma pesquisa entre seus habitantes. Faça um algoritmo em JavaScript para coletar dados sobre o salário e número de filhos de cada habitante e após as leituras, escreva no console, utilizando **Interpolação**:

- a) Média de salário da população
- b) Média do número de filhos
- c) Maior salário dos habitantes
- d) Percentual de pessoas com salário menor que R\$ 1000,00

Obs.: O final das leituras dos dados se dará com a entrada de um “salário negativo”. Utilize const e/ou let nas declarações de variáveis

5 - Escreva um algoritmo em JavaScript para ler um vetor **A** de 10 números. Após, o algoritmo deverá ler mais um número e guardar em uma variável **X**. Armazenar em um vetor **M** o resultado de cada elemento de **A** multiplicado pelo valor **X**. Imprima, no console utilizando **Interpolação**, o vetor **M**. Utilize const e/ou let nas declarações de variáveis.

6 - Faça um programa em JavaScript que recebe três números do usuário, e identifica o maior através de uma função e o menor número através de outra função. Utilize `const` e/ou `let` nas declarações de variáveis e a sintaxe Arrow function na criação das funções. Utilize **Interpolação** para imprimir no console.

7 - Crie uma função que recebe um inteiro positivo e teste para saber se ele é primo ou não. Faça um script que recebe um inteiro n e mostra todos os primos, de 1 até n . Utilize `const` e/ou `let` nas declarações de variáveis e a sintaxe Arrow function na criação das funções. Utilize **Interpolação** para imprimir no console.

8 - Faça um programa para imprimir no console :

```
1
2  2
3  3  3
...
n  n  n  n  n  n  ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n -ésima linha. Utilize `const` e/ou `let` nas declarações de variáveis e a sintaxe Arrow function na criação das funções. Utilize **Interpolação** para imprimir no console.

9 - Faça uma função que informe a quantidade de dígitos de um determinado número inteiro informado. Utilize `const` e/ou `let` nas declarações de variáveis e a sintaxe Arrow function na criação das funções. Utilize **Interpolação** para imprimir no console.

10 - Faça um programa que recebe três números do usuário, e identifica o maior através de uma função e o menor número através de outra função. Utilize `const` e/ou `let` nas declarações de variáveis e a sintaxe Arrow function na criação das funções. Utilize **Interpolação** para imprimir no console.

11 – Crie uma função que recebe uma string e verifica se o texto é um palíndromo ou não, retornando “É um palíndromo” caso seja, ou “Não é um palíndromo” caso contrário.

12 – Faça uma função que verifica se um número é primo ou não. Retornando “Não é um número primo” caso não seja, ou “É um número primo” caso seja.

13 – Escreva uma função que retorna o inverso de um número fornecido.

Exemplo:

forçando o número 12345 Retorno: 54321.

14 – Escreva uma função capaz de somar uma quantidade indeterminada de números.

Exemplo:

Somar(3, 5, 6) retorna 14

Somar(2, 5) retorna 7

15 – Escreva uma função que recebendo uma frase e uma letra retorna quantas vezes a letra esteve presente na frase.

Exemplo:

ContarLetra("frase de exemplo", "e") Retorno: 4

16 – Escreva uma função que mostre um retângulo sólido composto do mesmo caractere. A função deve receber como parâmetro a quantidade de linhas e colunas e exibir o retângulo com os seguintes parâmetros.

Exemplo:

geraRetangulo(4, 6) geraria o seguinte resultado no console:

```
VVVVVV
VVVVVV
VVVVVV
VVVVVV
```