



## Laboratório 01

Este Laboratório tem como objetivo a prática da linguagem JavaScript, usando o console do *browser*. Obs.: é permitida a utilização do VS Code ou do console do Node.JS, sob responsabilidade do estudante.

### I Instruções

O envio deve conter, para cada questão, um arquivo com o código exigido com extensão `.js` e uma demonstração de funcionamento do código.

Para fazer a leitura de entradas do usuário, utilize o comando `prompt()`. Exemplo:

```
const entrada = prompt('Informe um número inteiro');
```

Conversões podem ser feitas com `parseInt`, `parseFloat`, etc. Exemplo:

```
let outraEntrada = parseInt(prompt('Informe um número inteiro'));
```

### II Entrega

A entrega da atividade será on-line através do SIGAA. O estudante deve enviar um único arquivo ZIP contendo as suas implementações. O arquivo ZIP deve ser nomeado com a matrícula do estudante.

**Data limite para entrega: 26/03/2024, 19h00m.**

### III Atividades

1. Escreva um algoritmo em JS que peça o número de horas trabalhadas e o valor da hora de um determinado funcionário. Em seguida, calcule o salário do funcionário. O cálculo do salário deve ser feito dentro de uma função, enquanto a leitura e a impressão dos resultados deve ser feita no programa principal. Caso o funcionário tenha trabalhado mais de 200 horas, o salário final é acrescido de 5%. Exemplos de execução:

Horas trabalhadas: 120  
Valor da hora: 100,00  
Salário: 12000.00

Horas trabalhadas: 210  
Valor da hora: 50.00  
Salário: 11025.00

2. Uma loja vende produtos à vista e a prazo (pagamento 30 dias depois da compra). À vista tem um desconto de 5% e a prazo um acréscimo de 10%. Escreva uma *arrow function* em JS que peça o preço do produto e a forma de pagamento: 1 para à vista; e 2 para a prazo. Depois apresente o preço final do produto. Exemplos de execução:

Preço do produto: 120.00  
Forma de pagamento: 1  
Preço a vista: 114.00

Preço do produto: 200.00  
Forma de pagamento: 2  
Preço a prazo: 220.00

3. Faça uma função JS que calcule a duração de um evento qualquer. Para tal, o programa pede a hora de início e hora de fim (sem os minutos), as horas serão informadas de 0 a 23. Perceba que um evento pode começar em um dia e acabar em outro. Os eventos nunca duram mais de 24 horas. Exemplos de execução:

Início: 12

Fim: 4

Duração: 16 horas

Início: 10

Fim: 15

Duração: 5 horas

Início: 21

Fim: 6

Duração: 9 horas

4. Faça uma função JS que simule a multiplicação através de adições. Para tal serão pedidos os dois operandos. Por exemplo se for informado 3 e 4, deverá ser calculado, através de soma,  $3 * 4$ , ou seja, 12. Este cálculo é feito somando o primeiro valor informado por ele mesmo o número de vezes representada pelo segundo número. Nesse exemplo, o três seria somado quatro vezes:  $3+3+3+3$ , resultado 12.
5. Faça uma função usando a sintaxe *arrow function*, que peça um valor e imprima a soma de todos os números de 1 até o valor informado. Por exemplo, se o valor informado for 6, o resultado será 21, ou seja,  $1 + 2 + 3 + 4 + 5 + 6$ .

6. Dada uma sequência de números (um número menor ou igual à 0 finaliza a sequência), apresentar o percentual de números informados que são maior ou igual à 10 e menor ou igual à 20. Exemplo:

5  
6  
11  
21  
0  
% entre 10 e 20: 25.00%

7. Faça uma função recursiva em JS para calcular o fatorial de um número dado. O fatorial de um número é  $n \times (n - 1) \times (n - 2) \times \dots \times 1$ . Por definição, o fatorial de 0 e 1 são 1. Por exemplo, o fatorial de 5 é 120, ou seja,  $5 \times 4 \times 3 \times 2 \times 1$  (perceba que não é necessário fazer a última multiplicação já que 1 é o elemento neutro da multiplicação).
8. Faça uma função JS que peça 4 números inteiros. Em seguida, apresente quantos números informados são negativos e quantos são positivos (considere o 0 como positivo). Exemplos de execução:

|               |               |
|---------------|---------------|
| N1: 12        | N1: -1        |
| N2: 4         | N2: -20       |
| N3: -3        | N3: -7        |
| N4: 5         | N4: -11       |
| 3 (+) e 1 (-) | 0 (+) e 4 (-) |