

### **Lista de Exercícios**

Antes de iniciar os exercícios, crie uma página HTML simples contendo campos (*inputs*), botões para executar as funções e quaisquer outros elementos que julgar necessário para mostrar os resultados (*h1*, *span*, *p* etc.).

1. Receba um número e exiba se ele é par ou ímpar.
2. Dado dois números e uma operação (+, -, \*, /), retorne o resultado usando *if / else*.  
Desafio: reescreva usando *switch*.
3. Receba três notas, calcule a média e exiba: Aprovado ( $\geq 7$ ), Recuperação ( $\geq 5$  e  $< 7$ ) ou Reprovado ( $< 5$ ).
4. Imprima números de 10 até 1. Faça uma vez com *for* e uma vez com *while*.
5. Receba um número e gere a tabuada até 10 usando um *loop*.
6. Calcule a soma total do *array* a seguir usando um *loop for*.  

```
let numeros = [5, 10, 3, 8];
```
7. Sem usar funções prontas, encontre o maior número de um *array*.
8. Remova todas as ocorrências de "banana" do objeto a seguir.  

```
let frutas = ["maçã", "banana", "uva", "banana"];
```
9. Crie um *array* de nomes. Exiba o primeiro elemento, o último elemento e o tamanho da lista.
10. Dado um *array* de números, crie um novo *array* com cada valor multiplicado por 3.
11. Crie um objeto representando um usuário com nome, idade e e-mail. Exiba o nome, idade e e-mail.
12. Atualize a idade do usuário anteriormente criado e adicione uma nova propriedade: cidade.
13. Crie um *array* com 3 produtos (formato { nome, preco, categoria }). No fim, liste apenas os nomes.
14. No *array* de produtos acima, encontre o produto com categoria "eletronico".
15. Some todos os preços dos produtos.

16. Dado o objeto abaixo, filtre apenas os produtos com preço maior que 30.

```
let produtos = [
  { nome: "Mouse", preco: 25 },
  { nome: "Teclado", preco: 40 },
  { nome: "Headset", preco: 80 }
];
```

17. Use `map()` em produtos para gerar o seguinte *array*:

```
["Mouse", "Teclado", "Headset"]
```

18. Usando `reduce()` Mostre o preço total da compra representada no objeto abaixo.

```
let carrinho = [
  { nome: "Camiseta", preco: 25 },
  { nome: "Calça", preco: 80 },
  { nome: "Boné", preco: 30 }
];
```

19. Dado o objeto abaixo, use programação funcional para encontrar o de maior idade.

```
let usuarios = [
  { nome: "Ana", idade: 18 },
  { nome: "João", idade: 22 },
  { nome: "Pedro", idade: 16 }
];
```

20. Use `filter()` para gerar duas listas: uma com maiores de idade e outra com menores de idade.

21. Use `sort()` para ordenar o *array* de produtos.

22. Com o *array* de alunos abaixo, crie três listas (aprovados, reprovados e nomes dos aprovados). Use `map` + `filter`.

Desafios:

1. De acordo com o objeto abaixo, faça:

- Usando `filter`, crie uma função que retorne apenas produtos abaixo de R\$ 100.
- Usando `map`, crie uma nova lista contendo apenas os nomes dos produtos.
- Usando `reduce`, calcule o preço total dos produtos da categoria "periférico".

```
const produtos = [
  { nome: "Mouse", preco: 50, categoria: "periférico" },
  { nome: "Teclado", preco: 200, categoria: "periférico" },
```

```
{ nome: "Notebook", preco: 3500, categoria: "computador" },  
{ nome: "Cadeira Gamer", preco: 900, categoria: "mobiliário" }  
];
```

2. Dada a *string* (const texto = "JavaScript é incrível e JavaScript é divertido");

- a. Transforme a *string* em um *array* de palavras.
- b. Conte quantas vezes cada palavra aparece usando um objeto acumulador.
- c. Retorne algo como o exemplo abaixo (Dica: utilize o `reduce`).

```
{  
  JavaScript: 2,  
  é: 2,  
  incrível: 1,  
  e: 1,  
  divertido: 1  
}
```