

# Apostila: Dominando a Função Reduce no JavaScript

## O que é a função reduce?

A função reduce é uma das mais poderosas do JavaScript. Ela permite transformar um array em um único valor, seja ele um número, string, array, objeto ou booleano. A estrutura básica do reduce é:

```
array.reduce((acumulador, valorAtual) => {  
    return novoValor;  
}, valorInicial);
```

O acumulador carrega o valor de retorno da função anterior. A cada iteração, o valor do acumulador pode ser modificado ou mantido, dependendo da lógica aplicada.

## 1. Usando reduce para somar números

Exemplo: somar todos os números de um array usando reduce.

```
const numeros = [1, 2, 3, 4];  
const soma = numeros.reduce((acc, n) => acc + n, 0);  
console.log(soma); // 10
```

Exercícios:

1. Some os números: [10, 20, 30]
2. Some os números: [5, 5, 5, 5, 5]
3. Some os números negativos e positivos: [-10, 20, -5, 15]

## 2. Contar elementos com reduce

Exemplo: contar quantas vezes cada letra aparece em um array.

```
const letras = ['a', 'b', 'a'];  
const contagem = letras.reduce((acc, letra) => {  
    if (acc[letra]) acc[letra]++;  
    else acc[letra] = 1;  
    return acc;  
}, {});  
console.log(contagem); // { a: 2, b: 1 }
```

Exercícios:

1. Contar cores: ['azul', 'verde', 'azul', 'vermelho']
2. Contar nomes: ['Ana', 'Carlos', 'Ana', 'João', 'Carlos']
3. Contar letras: ['a', 'b', 'c', 'a', 'a', 'b']

## 3. Encontrar o maior número com reduce

Exemplo: encontrar o maior número em um array.

# Apostila: Dominando a Função Reduce no JavaScript

```
const numeros = [10, 25, 7, 90];  
const maior = numeros.reduce((acc, n) => n > acc ? n : acc);  
console.log(maior); // 90
```

## Exercícios:

1. Maior número: [1, 9, 3]
2. Maior número: [100, 99, 101, 98]
3. Maior número: [-10, -20, -3, -15]

## 4. Verificar condições com reduce (true/false)

Exemplo: verificar se todos os números são positivos usando reduce.

```
const numeros = [1, 2, 3];  
const todosPositivos = numeros.reduce((acc, n) => acc && n > 0, true);  
console.log(todosPositivos); // true
```

## Exercícios:

1. Verificar se todos são maiores que 10: [11, 15, 12]
2. Verificar se nenhum é negativo: [5, -3, 7]
3. Verificar se todos são pares: [2, 4, 6, 8]