

Módulo 7 – Funções Avançadas e Manipulação de Arrays em JavaScript

Objetivo do módulo

Aprender métodos avançados de arrays e funções que tornam o código mais eficiente e moderno.

1. Revisão rápida de funções

Funções permitem **reusar código**. Exemplo básico:

```
function soma(a, b) {  
    return a + b;  
}  
  
console.log(soma(3, 5)); // 8
```

2. Arrow Functions

Sintaxe mais curta para funções:

```
const soma = (a, b) => a + b;  
console.log(soma(3, 5)); // 8
```

3. Métodos avançados de arrays

3.1 map()

Cria um novo array transformando cada elemento.

```
let numeros = [1, 2, 3, 4];  
let dobrados = numeros.map(num => num * 2);  
console.log(dobrados); // [2, 4, 6, 8]
```

3.2 filter()

Cria um novo array com elementos que passam em um teste.

```
let pares = numeros.filter(num => num % 2 === 0);  
console.log(pares); // [2, 4]
```

3.3 reduce()

Reduz o array a um único valor.

```
let somaTotal = numeros.reduce((acumulador, num) => acumulador + num,  
0);  
console.log(somaTotal); // 10
```

3.4 find()

Encontra o primeiro elemento que satisfaça uma condição.

```
let maiorQueDois = numeros.find(num => num > 2);  
console.log(maiorQueDois); // 3
```

3.5 some() e every()

- `some()` verifica se **algum** elemento atende à condição.
- `every()` verifica se **todos** os elementos atendem à condição.

```
console.log(numeros.some(num => num > 3)); // true
console.log(numeros.every(num => num > 0)); // true
```

4. Funções como argumentos

Em JavaScript, funções podem ser passadas como **parâmetros**:

```
function saudacao(fn, nome) {
    console.log(fn(nome));
}

function ola(nome) {
    return "Olá, " + nome + "!";
}

saudacao(ola, "Maria"); // Olá, Maria!
```

5. Boas práticas

- Use **arrow functions** quando possível, mas prefira funções normais se precisar de `this`.
 - Prefira métodos como `map`, `filter` e `reduce` ao invés de loops tradicionais, pois deixam o código mais legível.
 - Evite alterar o array original dentro de métodos como `map`; crie sempre novos arrays.
-

6. Próximos passos

Agora você já tem um **conhecimento básico completo de JavaScript**.

Se quiser avançar, pode estudar:

- **Manipulação do DOM** (interagir com páginas web)
- **Eventos em JavaScript** (responder a cliques, teclas e movimentos do mouse)
- **Fetch API** (trabalhar com dados de servidores e APIs externas)

Pratique criando pequenos exercícios usando `map`, `filter` e `reduce`. Por exemplo, uma lista de compras ou de filmes e aplique os métodos para organizar e filtrar os dados.