

# Lista de Exercícios: Métodos de Arrays em JavaScript

Esta lista de exercícios foi elaborada para complementar a aula sobre os principais métodos de arrays em JavaScript, com foco nos métodos de manipulação, iteração, busca e ordenação.

## Exercícios

### Métodos de Adição e Remoção (`push`, `pop`, `shift`, `unshift`)

1. **`push()`**: Crie um array `listaDeCompras` com os itens "Pão" e "Leite". Adicione "Ovos" e "Café" ao final do array usando `push()`. Exiba o array resultante.
2. **`pop()`**: Dado o array `pilhaDeLivros = ["Livro A", "Livro B", "Livro C"]`, remova o último livro usando `pop()` e exiba o livro removido e o array atualizado.
3. **`shift()`**: Dado o array `filaDeAtendimento = ["Cliente 1", "Cliente 2", "Cliente 3"]`, remova o primeiro cliente usando `shift()` e exiba o cliente removido e a fila atualizada.
4. **`unshift()`**: Crie um array `tarefas = ["Estudar", "Trabalhar"]`. Adicione "Acordar" e "Meditar" ao início do array usando `unshift()`. Exiba o array resultante.
5. **Combinação de `push()` e `pop()`**: Crie um array `historico = []`. Adicione "Ação 1", "Ação 2" e "Ação 3" usando `push()`. Em seguida, simule o "desfazer" da última ação usando `pop()`. Exiba o histórico final.

### Métodos Iterativos (`forEach`, `map`, `filter`, `reduce`, `find`, `findIndex`)

6. **`forEach()`**: Dado um array de números `valores = [10, 20, 30, 40]`, use `forEach()` para imprimir cada valor multiplicado por 2.
7. **`map()`**: Dado um array de produtos `produtos = [{ nome: "Teclado", preco: 100 }, { nome: "Mouse", preco: 50 }, { nome: "Monitor", preco: 300 }]`, use `map()` para criar um novo array contendo apenas os nomes dos produtos.
8. **`filter()`**: Dado um array de idades `idades = [15, 22, 17, 30, 19, 16]`, use `filter()` para criar um novo array contendo apenas as idades maiores ou iguais a 18.
9. **`reduce()` (Soma)**: Dado um array de números `carrinho = [10, 25, 5, 15, 30]`, use `reduce()` para calcular o valor total do carrinho.
10. **`reduce()` (Concatenação)**: Dado um array de palavras `palavras = ["JavaScript", "é", "incrível", "e", "poderoso"]`, use `reduce()` para formar uma frase completa, separando as palavras por espaços.

11. **find()**: Dado um array de usuários `usuarios = [{ id: 1, nome: "Ana" }, { id: 2, nome: "Bruno" }, { id: 3, nome: "Carlos" }]`, use `find()` para encontrar o usuário com `id: 2`. Exiba o objeto encontrado.
12. **findIndex()**: Dado o mesmo array de usuários do exercício anterior, use `findIndex()` para encontrar o índice do usuário com `nome: "Ana"`. Exiba o índice.

## Métodos de Busca e Ordenação (`includes`, `sort`, `join`)

13. **includes()**: Dado um array `ingredientes = ["farinha", "açúcar", "ovos", "fermento"]`, verifique se o array inclui "ovos" e "chocolate" usando `includes()`. Exiba os resultados.
14. **sort() (Numérico)**: Dado um array de números `desordenados = [5, 2, 8, 1, 9, 3]`, ordene-o em ordem crescente e decrescente usando `sort()` com uma função de comparação. Exiba os arrays ordenados.
15. **sort() (Alfabético)**: Dado um array de nomes `nomes = ["Carlos", "Ana", "Bruno", "Daniela"]`, ordene-o em ordem alfabética. Exiba o array ordenado.
16. **join()**: Dado um array `partesDeEndereco = ["Rua A", "123", "Centro", "Cidade X", "Estado Y"]`, use `join()` para formar um endereço completo separado por vírgulas e espaços. Exiba o endereço formatado.
17. **join() com outro separador**: Dado um array `tags = ["javascript", "programacao", "web"]`, use `join()` para criar uma string de tags separadas por `#`. Exiba a string resultante.
18. **Combinação de `map()` e `join()`**: Dado um array de nomes `nomesComuns = ["joao", "maria", "pedro"]`, use `map()` para capitalizar a primeira letra de cada nome e, em seguida, use `join()` para uni-los em uma única string separada por `,`. Exiba a string final.