

List of Exercises: Array Methods in JavaScript

This list of exercises was developed to complement a class about the main array methods in JavaScript, with focus on manipulation, iteration, search and sorting.

Exercises

Methods of Addition and Removal (`push`, `pop`, `shift`, `unshift`)

1. **`push()`**: Create an array `listaDeCompras` with items "Pão" and "Leite". Add "Ovos" and "Café" at the end of the array using `push()`. Show the resulting array.
2. **`pop()`**: Given the array `pilhaDeLivros = ["Livro A", "Livro B", "Livro C"]`, remove the last book using `pop()` and show the removed book and the updated array.
3. **`shift()`**: Given the array `filaDeAtendimento = ["Cliente 1", "Cliente 2", "Cliente 3"]`, remove the first client using `shift()` and show the removed client and the updated queue.
4. **`unshift()`**: Create an array `tarefas = ["Estudar", "Trabalhar"]`. Add "Acordar" and "Meditar" at the beginning of the array using `unshift()`. Show the resulting array.
5. **Combination of `push()` and `pop()`**: Create an array `historico = []`. Add "Ação 1", "Ação 2" and "Ação 3" using `push()`. In turn, simulate the "undo" of the last action using `pop()`. Show the final history.

Iterative Methods (`forEach`, `map`, `filter`, `reduce`, `find`, `findIndex`)

6. **`forEach()`**: Given an array of numbers `valores = [10, 20, 30, 40]`, use `forEach()` to print each value multiplied by 2.
7. **`map()`**: Given an array of products `produtos = [{ nome: "Teclado", preco: 100 }, { nome: "Mouse", preco: 50 }, { nome: "Monitor", preco: 300 }]`, use `map()` to create a new array containing only product names.
8. **`filter()`**: Given an array of ages `idades = [15, 22, 17, 30, 19, 16]`, use `filter()` to create a new array containing only ages greater than or equal to 18.
9. **`reduce()` (Sum)**: Given an array of numbers `carrinho = [10, 25, 5, 15, 30]`, use `reduce()` to calculate the total value of the cart.
10. **`reduce()` (Concatenation)**: Given an array of words `palavras = ["JavaScript", "é", "incrível", "e", "poderoso"]`, use `reduce()` to form a complete sentence, separating words by spaces.

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.