

Sessão 06 (28/01/2026) – Módulo 02 – Fundamentos JavaScript – Aula 02 – Lógica-e-controlo-de-fluxo

TEORIA – INSERIR E REMOVER ELEMENTOS NO MEIO DO ARRAY

Em JavaScript, arrays são estruturas indexadas.

Cada elemento possui uma posição (índice) começando em 0.

Para INSERIR um elemento no meio de um array:

- Precisamos deslocar manualmente os elementos à direita.
- Normalmente percorremos o array de trás para frente.

Para REMOVER um elemento do meio:

- Deslocamos os elementos à esquerda para "tapar o buraco".
- Depois diminuimos o tamanho do array.

Esses exercícios ajudam a entender como arrays funcionam internamente.

EXPLICAÇÕES:

`indexOf()`

O que faz?

O método `indexOf()` procura um valor dentro de um array e retorna a primeira posição (índice) onde esse valor aparece.

Como funciona?

- A busca é feita da esquerda para a direita
- Se o valor existir, retorna o índice
- Se não existir, retorna -1

Sintaxe

`array.indexOf(valorProcurado)`

Exemplo

```
const frutas = ["maçã", "banana", "uva"];
```

```
console.log(frutas.indexOf("banana")); // 1
```

```
console.log(frutas.indexOf("pera")); // -1
```

👉 Muito usado para verificar se um valor existe no array usando if.

lastIndexOf()

O que faz?

O método `lastIndexOf()` funciona de forma parecida com o `indexOf()`, mas procura o valor da direita para a esquerda, retornando a última ocorrência.

Como funciona?

- Começa a busca pelo final do array
- Retorna o índice da última vez que o valor aparece
- Se não encontrar, retorna -1

Sintaxe

`array.lastIndexOf(valorProcurado)`

Exemplo

```
const numeros = [1, 2, 3, 2, 4];
```

```
console.log(numeros.lastIndexOf(2)); // 3
```

👉 Útil quando o array possui valores repetidos.

join()

O que faz?

O método `join()` transforma todos os elementos de um array em uma única string.

Como funciona?

- Junta os elementos do array
- Entre os elementos, insere um separador definido por você
- O array original não é alterado

Sintaxe

`array.join(separador)`

Exemplo

```
const palavras = ["Olá", "mundo", "JavaScript"];
```

```
const frase = palavras.join(" ");
```

```
console.log(frase); // "Olá mundo JavaScript"
```

Outros exemplos:

```
array.join(",") // separa por vírgula
```

```
array.join("-") // separa por hífen
```

```
array.join("") // sem separador
```

👉 Muito usado para criar frases, listas ou strings formatadas.

`concat()`

O que faz?

O método `concat()` junta dois ou mais arrays, criando um novo array.

Como funciona?

- Não altera os arrays originais
- Retorna um novo array combinado

Sintaxe

```
novArray = array1.concat(array2)
```

Exemplo

```
const a = [1, 2];
```

```
const b = [3, 4];
```

```
const c = a.concat(b);
```

```
console.log(c); // [1, 2, 3, 4]
```

Também é possível concatenar mais de um array:

```
a.concat(b, [5, 6])
```

👉 Muito usado para unir listas, resultados ou dados de várias fontes.

-
1. Crie um array com 5 números e imprima todos usando `for`.
 2. Some todos os números de um array usando `while`.
 3. Conte quantos números pares existem em um array.
 4. Encontre o maior número de um array.
 5. Encontre o menor número de um array.
 6. Inverta um array manualmente (sem `reverse()`).
 7. Insira um número na posição 2 de um array.
 8. Remova o elemento da posição 3 de um array.
 9. Use `indexOf` para encontrar a posição de um valor.
 10. Use `lastIndexOf` para encontrar a última ocorrência.
 11. Verifique se um valor existe no array usando `if`.
 12. Use `join` para transformar um array em string.

13. Use concat para juntar dois arrays.
14. Crie um menu com switch para manipular um array.
15. Conte quantas vezes um valor aparece no array.