

MetodosArrays

Exercícios – Métodos de Arrays (JavaScript)

Esses exercícios foram criados para ajudar a entender como funcionam ferramentas importantes para manipular listas (arrays) no JavaScript.



Exercício 1 – Concat(): Juntando duas listas

Enunciado:

Crie dois arrays com três números cada e junte-os usando `concat()`.

Dicas:

- `concat()` junta duas listas em uma só.
 - Não altera os arrays originais.
 - Cria um novo array.
-



Exercício 2 – Splice(): Removendo um elemento

Enunciado:

Crie um array com cinco itens e remova um elemento usando `splice()`.

Dicas:

- Funciona como um "recorte".
- O primeiro número é onde começa.

- O segundo número é quantos itens remover.
 - Altera o array original.
-



Exercício 3 – Slice(): Pegando parte do array

Enunciado:

Crie um array com cinco itens e use `slice()` para pegar apenas alguns elementos.

Dicas:

- Funciona como cortar um pedaço de bolo.
 - Não altera o array original.
 - O último índice não entra na fatia.
-



Exercício 4 – Join(): Transformando em texto

Enunciado:

Crie um array com três palavras e transforme em uma frase usando `join()`.

Dicas:

- Junta tudo em um texto só.
 - Você escolhe o separador: vírgula, traço, espaço etc.
-



Exercício 5 – Reverse(): Invertendo a ordem

Enunciado:

Crie um array com cinco itens e inverta a ordem usando `reverse()`.

Dicas:

- Transforma o primeiro no último e o último no primeiro.
 - Altera o array original.
-



Exercício 6 – Concat() com três arrays

Enunciado:

Crie três arrays e junte todos usando `concat()`.

Dicas:

- Pode juntar 2, 3 ou mais arrays.
 - Cria uma nova lista sem mexer nas anteriores.
-



Exercício 7 – Splice() com índice negativo

Enunciado:

Use `splice()` para remover um item contando de trás para frente.

Dicas:

- `1` é o último elemento.
 - `2` é o penúltimo.
 - Exemplo: `splice(-2, 1)` remove o penúltimo item.
-



Exercício 8 – Slice() com índice negativo

Enunciado:

Use `slice()` para pegar apenas os dois últimos elementos do array.

Dicas:

- `slice(-2)` pega os últimos dois itens.
 - O array original continua igual.
-



Exercício 9 – Join() com separador personalizado

Enunciado:

Transforme um array em string usando `join()` com um separador como `-` ou `/`.

Dicas:

- Você define o que fica entre os itens.
 - Exemplo: `A-B-C`.
-



Exercício 10 – Reverse() com palavras

Enunciado:

Crie um array de palavras e inverta a ordem com `reverse()`.

Dicas:

- Funciona do mesmo jeito com textos.
 - Depois pode transformar em frase com `join()`.
-



Exercício 11 – Sort(): Organizando o array

Enunciado:

Crie um array com palavras ou números misturados e organize usando `sort()`.

Dicas:

- `sort()` coloca os itens em **ordem alfabética**.
 - Para números, ele pode organizar errado se você não usar uma função especial (mas aqui será simples).
 - Exemplo com palavras funciona muito bem.
-



Exercício 12 – Fill(): Preenchendo o array

Enunciado:

Crie um array com cinco elementos e use `fill()` para preencher o array com um único valor.

Dicas:

- `fill()` substitui todos os itens do array por um mesmo valor.
 - Exemplo: `fill(0)` → transforma tudo em zeros.
 - Altera o array original.
-



Exercício 13 – Push(): Adicionando no final

Enunciado:

Crie um array e adicione três novos elementos usando `push()`.

Dicas:

- `push()` coloca um item no **final** do array.

- Funciona como “colocar mais um na fila”.
-



Exercício 14 – Pop(): Removendo do final

Enunciado:

Crie um array e remova o último elemento usando `pop()`.

Dicas:

- `pop()` tira o **último item**.
 - É como retirar o último da fila.
 - Ele devolve o item removido.
-



Exercício 15 – Flat(): Deixando tudo em uma só camada

Enunciado:

Crie um array com subarrays dentro dele (ex.: `[1, [2, 3], [4]]`) e deixe tudo “reto” usando `flat()`.

Dicas:

- `flat()` “achata” o array, tirando os arrays de dentro.
 - Exemplo: `[1, [2, 3]]` vira `[1, 2, 3]`.
 - Pode usar níveis maiores como `flat(2)`.
-



Exercício 16 – indexOf(): Encontrando a posição

Enunciado:

Crie um array e use `indexOf()` para descobrir em qual posição está um item.

Dicas:

- Mostra **o número do índice** onde o item está.
- Se não existir, retorna **1**.
- Exemplo:

- `["a", "b", "c"].indexOf("b")` → 1