

DESAFIO LÓGICA AFIADA

21
dias

by CódigoFonte



RESOLUÇÕES





DESAFIO 01

Bora então ordenar nosso array multidimensional. No desafio de ontem você podia escolher entre o **Bubble Sort** ou o **Selection Sort**. Nós escolhemos com o Selection Sort. Espero que tenha dado certo aí no seu código.

```
const produtos = [
  ["Arroz", 25.99],
  ["Feijão", 12.50],
  ["Leite", 6.49],
  ["Óleo", 8.99],
  ["Pão", 7.00],
  ["Café", 15.30],
  ["Açúcar", 4.89],
  ["Sal", 3.25],
  ["Macarrão", 5.79],
  ["Manteiga", 9.99]
];

function ordenarPorNome(arr) {
  let n = arr.length;

  for (let i = 0; i < n - 1; i++) {
    let minIndex = i;

    for (let j = i + 1; j < n; j++) {
      if (arr[j][0] < arr[minIndex][0]) { // Comparando
        pelo nome do produto
        minIndex = j;
      }
    }

    // Troca os elementos manualmente
    let temp = arr[i];
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }

  return arr;
}
```



```

function ordenarPorPreco(arr) {
  let n = arr.length;

  for (let i = 0; i < n - 1; i++) {
    let minIndex = i;

    for (let j = i + 1; j < n; j++) {
      if (arr[j][1] < arr[minIndex][1]) { // Comparando
        pelo preço do produto
        minIndex = j;
      }
    }

    // Troca os elementos manualmente
    let temp = arr[i];
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }

  return arr;
}

console.log("🛒 Lista Original:");
console.table(produtos);

console.log("📄 Produtos Ordenados por Nome:");
console.table(ordenarPorNome(produtos.slice())); // Criamos
uma cópia do array

console.log("$ Produtos Ordenados por Preço:");
console.table(ordenarPorPreco(produtos.slice())); // Criamos
uma cópia do array

```

Começamos com o array com a lista de produtos, mas esse você já tinha, pois nós que fornecemos. Veja que o primeiro elemento de cada elemento do array principal é o nome do produto e o segundo o preço.

Então com isso em mente vamos aplicar na função **ordenarPorNome** a ordenação, veja que passamos o array completão por parâmetro, então a função fará a mudança desse array e retornará ele já ordenado.



Na primeira linha, simplesmente obtemos o tamanho total desse array. Em seguida com um **for** vamos percorrendo todo array até o penúltimo item (veja que **$i < n - 1$**). Iteramos utilizando a variável **i** e armazenamos o menor índice encontrado na variável **minIndex**.

Para encontrarmos o menor elemento vamos precisar de um outro **for** de onde iniciamos a variável **j** a partir do valor de **i** que estamos iterando no **for** principal. Utilizando um **if** podemos verificar se o elemento atual (que está no índice **j**) é menor que o elemento que está no índice **minIndex**. Como estamos fazendo essa verificação com o nome do produto, então temos que utilizar sempre o índice **[0]**. Lembra que ele vem primeiro no array?

Caso seja realmente menor então temos um novo elemento menor. Essa comparação com texto acontece exatamente igual a números no JavaScript, então podemos definir um novo **minIndex** exatamente com o valor de **j**.

Completando esse **for** mais interno, podemos afirmar que temos um novo **minIndex**, então basta trocar o elemento de lugar. Pra isso criamos uma variável **temp** para armazenar o elemento atual na posição **i** e atribuímos o elemento da posição **i** utilizando o **minIndex**. E o que é menor e está na posição **minIndex** agora recebe o array que está em **temp**.

Se formos agora ver a função **ordenarPorPreco**, a única diferença é exatamente o índice que representa o preço no array, que nesse caso é **1**. Então podemos simplesmente copiar e alterar o código ou fazer ainda melhor.



Se você quiser, pode criar somente uma função e definir qual é o índice que será ordenado (0 ou 1) passando por parâmetro. Assim se você precisar adicionar mais elementos nesse array, você já terá uma função pronta para ordenar qualquer informação. Minha dica é fazer essa mudança.

Por fim para testar esse código, fizemos algumas chamadas mostrando o array original e depois ordenado por nome e por preço. Veja que ao passar o array por parâmetro nas funções, também usamos a função **slice()** do JavaScript, assim ele cria uma cópia do array original na memória.

Como estamos lidando com um array, ao invés de **console.log**, podemos utilizar o **console.table** para deixar os dados já formatados.

DESAFIO LÓGICA AFIADA



by **Código**Fonte

desafiologicaafiada.com.br