

Estruturas de dados homogêneas

Vetores - Parte 2

O que é um vetor

- Agora que já vimos o que são vetores, iremos compreender alguns métodos comuns da linguagem JavaScript voltados para manipulação dessa importante estrutura de dados.
- Veremos formas de inserir novos dados, remover dados, subdividir os vetores, dentre outros.
- Isso facilitará a manipulação desse tipo de estrutura, tornando-a mais simples de se utilizar e mais simples de ser aplicada a diferentes contextos.

Inserindo elementos a um vetor

- Conforme mencionado anteriormente, é possível criar um vetor vazio, ou mesmo com alguns elementos, e posteriormente ir adicionando elementos a ele conforme novos dados forem surgindo.
- Para inserir novos dados no fim de um vetor podemos utilizar o método “push”:
- `let vetor = [1, 2, 3];`
- `vetor.push(4, 5);`
- `// Resultado: vetor = [1, 2, 3, 4, 5]`

Inserindo elementos a um vetor

- Para inserir novos dados no início de um vetor podemos utilizar o método “unshift”:
- `let vetor = [1, 2, 3];`
- `vetor.unshift(-1, 0);`
- `// Resultado: vetor = [-1, 0, 1, 2, 3]`

Remoção de elementos

- Remover elementos de um vetor também é importante, para isso temos dois métodos muito utilizados:
- pop: Remove elementos do fim do vetor
 - **let vetor = [1, 2, 3, 4, 5];**
 - **vetor.pop();**
 - **// Resultado: vetor = [1, 2, 3, 4]**
- shift: Remove elementos do início do vetor
 - **let vetor = [1, 2, 3, 4, 5];**
 - **vetor.shift();**
 - **// Resultado: vetor = [2, 3, 4, 5]**

Inserir ou remover em posição específica

- Para inserir ou remover elementos a partir de um posição específica do nosso vetor podemos utilizar o método “splice”:
- Removendo com splice:
 - **let vetor = [1, 2, 3, 4, 5];**
 - **vetor.splice(2, 1);** //a partir da posição 2 remova 1 elemento
 - **// Resultado: vetor = [1, 2, 4, 5]**
- Inserindo com splice:
 - **let vetor = [1, 2, 3, 4, 5];**
 - **vetor.splice(2, 0, 'a', 'b');** //a partir da posição 2 remova 0 elementos e insira a e b
 - **// Resultado: vetor = [1, 2, 3, 'a', 'b', 4, 5]**

Criar subvetor (slice)

- O método slice cria uma cópia superficial (shallow copy) de uma parte de um vetor, sem modificar o vetor original.
- Utilização do método slice:
 - **let vetor = [1, 2, 3, 4, 5];**
 - **let subvetor = vetor.slice(1, 4);**//Onde irá iniciar (posição 1 - inclusive) e onde termina (posição 4 - exclusive)
 - **// Resultado: subvetor = [2, 3, 4]**

Concatenar dois vetores

- O método `concat` é utilizado quando temos dois ou mais vetores e queremos juntá-los, formando apenas um.
- Utilização do método `concat`:
 - **`let vetor = [1, 2, 3, 4, 5];`**
 - **`let vetor2 = [6, 7, 8, 9, 10];`**
 - **`let concatenado = vetor.concat(vetor2);`**
 - **`// Resultado: concatenado = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`**

Procurando um elemento no vetor

- Para retornar o índice de um elemento que EXISTE no vetor podemos utilizar dois métodos:
- `indexOf` (retorna o índice do primeiro elemento que atende aos critérios da busca)
 - **`let vetor = [1, 2, 3, 4, 2];`**
 - **`let indice = vetor.indexOf(2);`**
 - **// Resultado: indice = 1**
- `lastIndexOf` (retorna o índice do último elemento que atende aos critérios da busca)
 - **`let vetor = [1, 2, 3, 4, 2];`**
 - **`let indice = vetor.lastIndexOf(2);`**
 - **// Resultado: indice = 4**

Ordenando elementos de um elemento no vetor

- Para ordenar elementos de um vetor podemos utilizar o método “sort”
- Vetor de palavras
 - **let frutas = ['banana', 'maçã', 'uva', 'abacaxi', 'laranja'];**
 - **frutas.sort();**
- Vetor de inteiros (ordem crescente)
 - **let numeros = [10, 5, 8, 2, 1];**
 - **numeros.sort((a, b) => a - b);**
- Vetor de inteiros (ordem decrescente)
 - **let numeros = [10, 5, 8, 2, 1];**
 - **numeros.sort((a, b) => b - a);**

Conclusão

- Estes métodos são poderosos para manipular vetores em JavaScript.
- Estes não são os únicos métodos existentes, há muitos outros que podem auxiliá-lo na manipulação de vetores, entretanto aqui foram apresentados os mais comuns que serão utilizados por você ao longo de todo curso.
- Escolha o método adequado com base nas necessidades específicas do seu código.

Exercícios

1. Dado um vetor de números e um segundo vetor de números, crie uma função que retorne um novo vetor contendo os elementos do primeiro vetor seguidos pelos elementos do segundo vetor.
2. Dado um vetor de números, crie uma função que divida o vetor em duas partes iguais e retorne um novo vetor contendo as duas partes concatenadas, com a segunda parte aparecendo primeiro.
3. Dados dois vetores de números, ordene o primeiro em ordem crescente, o segundo de forma decrescente e posteriormente faça a junção dos dois vetores.