

Introdução a Interfaces Pessoa-Máquina

TPC-7 (Parte 1-Pesquisa sobre Arrays e Iterables)

Liedson Delgado - LEIT 2º Ano

Arrays(Vetores)

Em JavaScript, um array é uma estrutura de dados que armazena uma coleção ordenada de elementos, registrados numericamente a partir de zero.

Arrays oferecem funcionalidade para adicionar, remover e manipular seus elementos desse array.

Eles funcionam como um vetor infinito, com posições que vão de 0 até n-1, onde cada elemento fica armazenado em uma posição específica que pode ser acessada diretamente.

Iterables(Iteradores)

Um **iteradores** em JavaScript é um objeto que permite percorrer uma lista de elementos, como um array, de maneira ordenada e controlada.


Ele fornece uma interface para acessar os elementos dessa lista um de cada vez, sem que seja necessário conhecer ou manipular diretamente a estrutura interna da lista.

❖ ***Funções que permitem manipular um array(arrays e iterables)***

Em JS existe varias funcoes que permitem a manipular arrays de diversas formas, como algumas citadas e exemplificadas abaixo pois existem varias.

- **PUSH()**- Permite a adiç o de um elemento no final do array e atualiza e retorna o seu comprimento.


EXEMPLO PUSH():



```
1 let array=[1,2,3];
2 array.push(4,5); //adiciona 4 e 5 ao array
3 console.log(array); //array atualizado para [1,2,3,4,5]
4
```

- **POP()**- Permite a remo o do ultimo elemento do array, caso o array estiver vazio retorna undefined ou simplesmente retorna o elemento removido.

EXEMPLO POP():



```
1 let array=[1,2,3,4,5];
2 array.pop; //remove o  ltimo do array nesse caso sera o 5
3 console.log(array); //array atualizado para [1,2,3,4]
```

- **SHIFT()**- Permite remover o primeiro elemento contido no inicio do array, retorna o elemento removido ou undefined se o array estiver vazio.

EXEMPLO SHIFT():



```
1 let array=[1,2,3];
2 array.shift(); // remove o 1 do array
3 console.log(array); //array atualizado para [2,3]
```

- **UNSHIFT()-** Permite adicionar um ou varios elementos no inicio do array, retorna o novo comprimento do array.

EXEMPLO UNSHIFT():



```
1 let array=[1,2,3];
2 array.unshift(0); //adiciona 0 ao inicio do vetor
3 console.log(array); //atualiza o vetor para [0,1,2,3]
```

- **CONCAT()-** Permite “fundir” varios arrays em apenas um unico retornando um novo array contendo os elementos que cada array continha inicialmente.

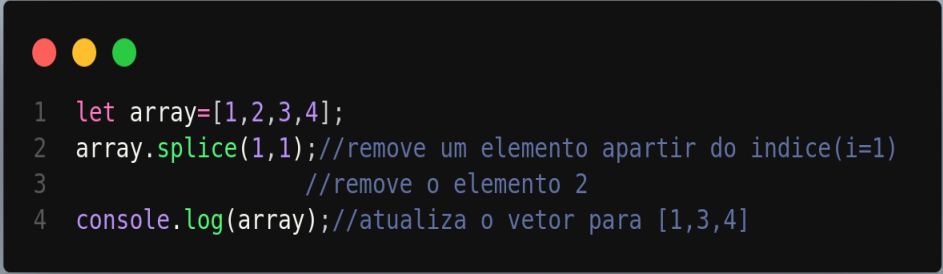
EXEMPLO CONCAT():



```
1 let array_1=[1,3,5];
2 let array_2=[2,4,6];
3 let new_array=array_1.concat(array_2); //a combinacao dos dois arrays
4 console.log(new_array); //novo vetor resultado da combinacao de
5                          // array_1 e array_2
```

- **SPLICE()**- Permite a alterar um array, adicionando,removendo ou substituindo elementos existentes por novos elementos retornando o array com os elementos removidos.


EXEMPLO SPLICE():



```
1 let array=[1,2,3,4];
2 array.splice(1,1);//remove um elemento apartir do indice(i=1)
3           //remove o elemento 2
4 console.log(array);//atualiza o vetor para [1,3,4]
```

- **INDEXOF()**- Permite fazer a procura de um elemento do array, que caso for encontrado retorna o seu indice caso contrario não encontrar retorna -1.

EXEMPLO INDEXOF:



```
1 let array=[1,2,3];
2 let return_indexOf=array.indexOf(3); //faz a procura do elemento 3
3 console.log(return_indexOf); //retorna o indice de 3(i=2)
```

- **SLICE()**- Permite criar uma copia de um array de um de determinado elemento indicando onde começa e termina que é no final.

EXEMPLO SLICE:



```
1 let array=[1,2,3,4,5,6];
2 new_array=array.slice(1,4);//extrai elementos do indice 1 ao 4(i=[1,4])
3 console.log(new_array); //retorna um novo vetor para com os elementos [1,6]
4                          //extraídos apartir dos indices indicados
```

- **MAP()-** Permite a criar um novo array com o resultado de uma funcao contida de cada elemento do array inicialmente criado.

EXEMPLO MAP():



```
1
2 let array=[1,2,3,4,5,6];
3 let quadrado=array.map(val=>val*val); //faz o quadrado cada elemento ou seja [1*1,2*2,3*3,4*4,5*5,6*6]
4 console.log(quadrado); //retorna o novo array apartir da operacao com os elementos do array inicial
5                          //[1,4,9,16,25,36]
```

- **FILTER()-** Permite retornar um vetor que cujo seus elementos satisfazem uma determinada condição.

EXEMPLO FILTER():



```
1 let notas=[3,6,9,14,15,17,19.2,13];
2 let estado_aluno=notas.filter(val =>{ //compara os valores do array para ver qual satisfaz a condicao
3     if(val>=10){
4         console.log("Aprovado");
5     }else{
6         console.log("Reprovado");
7     }
8 });
9 console.log(estado_aluno); // retorna um novo vetor com os elementos que satisfizeram as condicoes
10                          //[13,14,15,17,19.2]
```

Referências bibliográficas da pesquisa:

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://javascript.info/array>

https://devdocs.io/javascript/global_objects/array/map