Vetores

Professor: Dr Renato William R. de Souza

Análise e Desenvolvimento de Sistemas

Lógica de Programação

Tamanho do Array

O comprimento de um array, ou o número de elementos que ele contém, pode ser obtido usando a propriedade length:

```
let numeros = [1, 2, 3, 4, 5];
console.log(numeros.length); // 5
```

Iterando sobre Arrays

Você pode percorrer todos os elementos de um array de várias maneiras:

Usando for:	**let numeros = [1, 2, 3, 4, 5];
	for (let i = 0; i < numeros.length; i++) {
	  console.log(numeros[i]);
	}**

Usando forEach():	numeros.forEach(function(numero) {
	  console.log(numero);
	});

Usando for...of:	for (let numero of numeros) {
	  console.log(numero);
	}

O método forEach

Esse método permite percorrer todos os itens do array de maneira simples e eficiente, sem a necessidade de criar loops manuais como o for ou o while.

Sintaxe:

```
array.forEach(function(elemento, indice, array)
{
    // código a ser executado para cada
elemento
});
```

- elemento: O item atual do array sendo processado.
- indice (opcional): O índice do item atual.
- array (opcional): O próprio array sendo percorrido.

Exemplo ForEach

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


frutas.forEach(function(fruta, indice) {
    console.log(indice + ": " + fruta);
});
```

saída

0: maçã
1: banana
2: laranja

ForEach - Principais Características

- **Não retorna valores:** O forEach executa a função em cada item, mas não retorna um novo array, ao contrário de outros métodos como map().
- **Não interrompe a execução:** Diferente de loops como for e while, não é possível usar break ou continue dentro de um forEach. Ele sempre percorre todos os itens do array.
- **Não altera o array original:** A função callback não modifica o array original a menos que explicitamente modifique os elementos dentro da função.

ForEach - Uso com funções arrow

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

frutas.forEach((fruta, indice) => {
    console.log(`${indice}: ${fruta}`);
});
```

Usando for...of

O código em JavaScript usa o loop for...of para iterar sobre um array, imprimindo cada valor dentro dele.

```
let numeros = [1, 2, 3, 4, 5];

for (let numero of numeros) {
  console.log(numero);
}
```

for ...of - Explicação

```
for (let numero of numeros) {
    console.log(numero);
}
```

- O loop for...of percorre cada valor do array numeros, atribuindo-o à variável numero em cada iteração.
- let numero: Declara uma variável numero que irá armazenar o valor atual do array em cada iteração.
- of numeros: Isso indica que o loop vai iterar sobre os elementos de numeros.
- Para cada item no array, o valor será passado para a variável numero, e o bloco de código dentro do loop será executado.

Inclusão e exclusão de itens

- push() – Adiciona um ou mais elementos ao final do array;
- pop() – Remove o último elemento do array;
- shift() – Remove o primeiro elemento do array;
- unshift() – Adiciona um ou mais elementos no início do array;
- Filter() - cria um novo array contendo apenas os elementos que atendem à condição fornecida.

push() - Adiciona um ou mais elementos ao final do array

```
let numeros = [1, 2, 3];
numeros.push(4, 5);
console.log(numeros); // [1, 2, 3, 4, 5]
```

Método push ()

```
let frutas = ["Maçã", "Banana"];
frutas.push("Laranja");
console.log(frutas); // ["Maçã", "Banana", "Laranja"]
```

Inserindo múltiplos dados

```
frutas.push("Morango", "Uva");
console.log(frutas); // ["Maçã", "Banana", "Laranja",
"Morango", "Uva"]
```

pop() - Remove o último elemento do array

```
let numeros = [1, 2, 3, 4, 5];
numeros.pop();
console.log(numeros); // [1, 2, 3, 4]
```

shift() – Remove o primeiro elemento do array

```
let numeros [1, 2, 3, 4]
numeros.shift();
console.log(numeros); // [2, 3, 4]
```

unshift() – Adiciona um ou mais elementos no início do array

```
let numeros [2, 3, 4]


numeros.unshift(0);
console.log(numeros); // [0, 2, 3, 4]
```

Remover Elementos Específicos: filter()

```
let numeros = [1, 2, 3, 4, 5];
let numerosFiltrados = numeros.filter(function(numero) {
  return numero > 3;
});
console.log(numerosFiltrados); // [4, 5]
```

- Nesse caso, o novo array contém apenas os números maiores que 3.

Localizar Conteúdo

Como o número de elementos de um vetor pode ser grande, as linguagens de programação dispõem de alguns métodos para nos auxiliar no controle de seu conteúdo.

1. indexOf( )
2. lastindexOf( )
3. includes ( )

Localizar Conteúdo

indexOf( )

```
<script>

const idades = [5,6,8,3,6,9]

console.log(idades.indexOf(6)) //retorna 1
console.log(idades.lastIndexOf(6)) //retorna 4
console.log(idades.indexOf(7)) //retorna -1
console.log(idades.includes(3)) //retorna true


</script>
```

1. O método indexOf() procura pelo primeiro índice onde o valor especificado aparece no array. Neste caso, estamos procurando o valor 6.
2. O primeiro 6 no array idades aparece no índice 1 (os índices começam em 0). Então, o resultado será 1.

indexOf() com um valor que não existe

```
<script>

const idades = [5,6,8,3,6,9]

console.log(idades.indexOf(6)) //retorna 1
console.log(idades.lastIndexOf(6)) //retorna 4
console.log(idades.indexOf(7)) //retorna -1
console.log(idades.includes(3)) //retorna true


</script>
```

1. Aqui, o método indexOf() tenta encontrar o valor 7 no array.

   Como o valor 7 não está presente no array, o método retorna -1,

   o que indica que o valor não foi encontrado.

Localizar Conteúdo

lastIndexOf()

```
<script>

const idades = [5,6,8,3,6,9]

console.log(idades.indexOf(6)) //retorna 1
console.log(idades.lastIndexOf(6)) //retorna 4
console.log(idades.indexOf(7)) //retorna -1
console.log(idades.includes(3)) //retorna true


</script>
```

1. O método lastIndexOf() é semelhante a indexOf(), mas em vez de procurar a primeira ocorrência, ele retorna o último índice onde o valor aparece no array.
2. O último 6 no array idades aparece no índice 4. Portanto, o resultado será 4.

Localizar Conteúdo

includes( )

```
<script>

const idades = [5,6,8,3,6,9]

console.log(idades.indexOf(6)) //retorna 1
console.log(idades.lastIndexOf(6)) //retorna 4
console.log(idades.indexOf(7)) //retorna -1
console.log(idades.includes(3)) //retorna true


</script>
```

1. O método includes() verifica se o valor especificado está presente no array. Ele retorna um valor booleano (true ou false).
2. Como o valor 3 está presente no array idades (no índice 3), o método retorna true.

Atividades

Escreva uma função que receba um array de números e um número adicional. A função deve adicionar esse número ao final do array e retornar o novo array. Exemplo:

```
const numeros = [1, 2, 3];
adicionarNumero(numeros, 4);
// Resultado esperado: [1, 2, 3, 4]
```

Atividades

Dado o seguinte array de nomes, utilize o método forEach para imprimir no console uma mensagem como "Nome: [nome]" para cada nome do array.

const nomes = ['Ana', 'João', 'Pedro', 'Maria'];

Exemplo de saída esperada:

Nome: Ana

Nome: João

Nome: Pedro

Nome: Maria

Atividades

Crie um programa que utilize um loop for...of para percorrer o array de números e imprimir cada número multiplicado por 2.

const numeros = [2, 4, 6, 8];

Exemplo de saída esperada:

4

8

12

16

Atividades

Implemente uma função que receba um array e um valor. A função deve remover todas as ocorrências desse valor no array.

```
const numeros = [1, 2, 3, 4, 3, 5];
removerTodos(numeros, 3);
// Resultado esperado: [1, 2, 4, 5]
```

Atividades

Elaborar um programa que adicone números a um vetor. O programa deve impedir a inclusão de números repetidos. Exibir a lista de números a cada inclusão. Ao clicar no botão de Verificar ordem, o programa deve analisar o conteúdo do vetor e informar se os números estão ou não em ordem crescente.

Atividades

Elaborar um programa que lei nome e número de acertos de candidatos inscritos em um concurso. Listar os dados a cada inclusão. Ao clicar no botão Aprovados 2ª fase, ler o número de acertos para aprovação dos candidados para a 2ª fase do concurso. O programa deve então, exibir os candidatos aprovados, ou seja, apenas os que obtiveram nota maior ou igual a nota informada. exibir os candidatos aprovados em ordem Descrecente de número de acertos. Caso nenhum candidato tenha sido aprovado, exibir mensagem.

Atividades

Elaborar um programa para gerar uma tabela com os jogos de uma fase eliminatória de um campeonato. O programa deve conter 3 funções (Faça um menu de opções)
1- Validar o preenchimento, adiconar um clube e ao vetor e listar os clubes;

2- Listar os clubes

3 - Montar a tabela de jogos, no formato primeiro x último, segundo x penultimo e assim por diante.

Exibir mensagem e não listar a tabela de jogos, caso o numero de clubes informados seja impar.