Curso

Aula 01

Variáveis

Declarações de variáveis, onde quer que elas ocorram, são processadas antes que qualquer outro código seja executado.

let
Permite que você declare variáveis limitando seu escopo no bloco, instrução, ou em uma expressão na qual ela é usada. Isso é inverso da keyword var, que define uma variável globalmente ou no escopo inteiro de uma função, independentemente do escopo de bloco.

var
O **variable statement** declara uma variável, opcionalmente é possível atribuir à ela um valor em sua inicialização.

const
A declaração const cria uma variável cujo o valor é fixo, ou seja, uma constante somente leitura. Isso não significa que o valor é imutável, apenas que a variável constante não pode ser alterada ou retribuída.

Array

O objeto Array do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

```
var frutas = ['Maçã', 'Banana'];

console.log(frutas.length);
// 2
```

Array.prototype.map

O método **map()** invoca a função callback passada por argumento para cada elemento do Array e devolve um novo Array como resultado.

```
var numbers = [1, 4, 9];
var doubles = numbers.map(function(num) {
  return num * 2;
});
// doubles é agora [2, 8, 18]. numbers ainda é [1, 4, 9]
```

Array.prototype.find

O método **find()** retorna o **valor** do **primeiro elemento** do array que satisfizer a função de teste provida. Caso contrario, undefined é retornado.

```
const inventory = [
    {name: 'maças', quantity: 2},
    {name: 'bananas', quantity: 0},
    {name: 'cherries', quantity: 5}
];

const result = inventory.find( fruit => fruit.name === 'cherries' );

console.log(result) // { name: 'cherries', quantity: 5 }
```

Array.prototype.filter

O método filter() cria um novo array com todos os elementos que passaram no teste implementado pela função fornecida.

```
var filtered = [12, 5, 8, 130, 44].filter(value => value >= 10);
// filtered is [12, 130, 44]
```

Array.prototype.forEach

O método forEach() executa uma dada função em cada elemento de um array.

```
[2, 5, 9].forEach((value, index) => console.log("a[" + index + "] = " + value));
// logs:
// a[0] = 2
// a[1] = 5
// a[2] = 9
```

Array.prototype.reduce

O método **reduce()** executa uma função **reducer** (fornecida por você) para cada elemento do array, resultando num único valor de retorno.

```
[0, 1, 2, 3, 4].reduce(function(acumulador, valorAtual, index, array) {
  return acumulador + valorAtual;
});
// 10
```

Aula 02

Array.prototype.push()

O método push() adiciona um ou mais elementos ao final de um array e retorna o novo comprimento desse array.

```
var numeros = [1, 2, 3];
numeros.push(4);

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

numeros.push(5, 6, 7);

console.log(numeros); // [1, 2, 3, 4, 5, 6, 7]
```

Array.prototype.pop()

O método **pop()** remove o **último** elemento de um array e retorna aquele elemento.

```
var meuPeixe = ['acara-bandeira', 'palhaco', 'mandarim', 'esturjao'];

console.log(meuPeixe); // ['acara-bandeira', 'palhaco', 'mandarim', 'esturjao']

var meuPeixePop = meuPeixe.pop();

console.log(meuPeixe); // ['acara-bandeira', 'palhaco', 'mandarim' ]

console.log(meuPeixePop); // 'esturjao'
```

Array.prototype.sort()

O método **sort()** ordena os elementos do próprio array e retorna o array. A ordenação não é necessariamente estável. A ordenação padrão é de acordo com a pontuação de código unicode.

```
var fruit = ['cherries', 'apples', 'bananas'];
fruit.sort(); // ['apples', 'bananas', 'cherries']

var scores = [1, 10, 2, 21];
scores.sort(); // [1, 10, 2, 21]
// Observe que 10 vem antes do 2,
// porque '10' vem antes de '2' em ponto de código Unicode.

var things = ['word', 'Word', '1 Word', '2 Words'];
things.sort(); // ['1 Word', '2 Words', 'Word', 'word']
// Em Unicode, números vêem antes de letras maiúsculas,
// as quais vêem antes das minúsculas.
```

Array.prototype.indexOf()

O método **indexOf()** retorna o primeiro índice em que o elemento pode ser encontrado no array, retorna -1 caso o mesmo não esteja presente.

```
var array = [2, 5, 9];
array.indexOf(2);     // 0
array.indexOf(7);     // -1
array.indexOf(9, 2);  // 2
array.indexOf(2, -1); // -1
array.indexOf(2, -3); // 0
```

Array.prototype.join()

O método **join()** junta todos os elementos de um array (ou um array-like object) em uma string e retorna esta string.

```
arr.join([separador = ','])
```

Array.prototype.splice()

O método **splice()** altera o conteúdo de uma lista, adicionando novos elementos enquanto remove elementos antigos.

```
//remove 1 elemento do índice 3
removed = myFish.splice(3, 1);
//myFish é ["angel", "clown", "drum", "surgeon"]
//removed é ["mandarim"]
```