OBJETOS

Array

Arrays

Arrays armazenam uma coleção de elementos. Estes podem ser strings, arrays, boolean, number, functions, objects e mais.

```
const instrumentos = ['Guitarra', 'Baixo', 'Violão'];
const precos = [49, 99, 69, 89];

const dados = [new String('Tipo 1'), ['Carro', 'Portas', {cor:
'Azul', preco: 2000}], function andar(nome) { console.log(nome)
}];

dados[2]('Ford');
dados[1][2].cor; // azul
```

Construção de Arrays

Toda array herda os métodos e propriedades do protótipo do construtor Array.

```
const instrumentos = ['Guitarra', 'Baixo', 'Violão'];
const carros = new Array('Corola', 'Mustang', 'Honda');

carros[1] // Mustang
carros[2] = 'Ferrari';
carros[10] = 'Parati';
carros.length; // 11
```

> *Os valores das array's não são estáticos*

Array.from()

Array.from() é um método utilizado para transformar array-like objects, em uma array.

```
let li = document.querySelectorAll('li'); // NodeList
li = Array.from(li); // Array

const carros = {
  0: 'Fiat',
  1: 'Honda',
  2: 'Ford',
  length: 4,
}

const carrosArray = Array.from(carros);
```

Array.isArray()

Verifica se o valor passado é uma array e retorna um valor booleano.

```
let li = document.querySelectorAll('li'); // NodeList
Array.isArray(li); // false

li = Array.from(li); // Array
Array.isArray(li); // true
```

Array.of(), Array() e new Array()

Verifica se o valor passado é uma array e retorna um valor booleano. A palavra chave new não é necessária para utilizar o construtor Array.

```
Array.of(10); // [10]
Array.of(1,2,3,4); // [1,2,3,4]
new Array(5); // [,,,,]
Array(5); // [,,,,]
Array(1,2,3,4); // [1,2,3,4]
```

Propriedades e Métodos do Prototype

`[].length` retorna o tamanho da array.

```
const frutas = ['Banana', 'Pêra', ['Uva Roxa', 'Uva Verde']];
frutas.length; // 3

frutas[0].length; // 6
frutas[1].length; // 5
frutas[2].length; // 2
```

Métodos Modificadores [].sort()

Os próximos métodos que vamos falar sobre, são métodos modificadores (mutator methods). Além de retornarem um valor, eles modificam a array original. `[].sort()` organiza a pelo unicode.

```
const instrumentos = ['Guitarra', 'Baixo', 'Violão'];
instrumentos.sort();
instrumentos; // ['Baixo', 'Guitarra', Violão]

const idades = [32,21,33,43,1,12,8];
idades.sort();
idades; // [1, 12, 21, 32, 33, 43, 8]
```

[].unshift() e [].push()

`[].unshift()` adiciona elementos ao início da array e retorna o length da mesma. `[].push()` adiciona elementos ao final da array e retorna o length da mesma.

```
const carros = ['Ford', 'Fiat', 'VW'];
carros.unshift('Honda', 'Kia'); // 5
carros; // ['Honda', 'Kia', 'Ford', 'Fiat', 'VW'];

carros.push('Ferrari'); // 6
carros; // ['Honda', 'Kia', 'Ford', 'Fiat', 'VW', 'Ferrari'];
```

[].shift() e [].pop()

`[].shift()` remove o primeiro elemento da array e retorna o mesmo. `[].pop()` remove o último elemento da array e retorna o mesmo.

```
const carros = ['Ford', 'Fiat', 'VW', 'Honda'];
const primeiroCarro = carros.shift(); // 'Ford'
carros; // ['Fiat', 'VW', 'Honda'];

const ultimoCarro = carros.pop(); // 'Honda'
carros; // ['Fiat', 'VW'];
```

[].reverse()

`[].reverse()` inverte os itens da array e retorna a nova array.

```
const carros = ['Ford', 'Fiat', 'VW', 'Honda'];
carros.reverse(); // ['Honda', 'VW', 'Fiat', 'Ford'];
```

[].splice()

`[].splice(index, remover, item1, item2, ...)` adiciona valores na array a partir do index. Remove a quantidade de itens que for passada no segundo parâmetro (retorna esses itens).

```
const carros = ['Ford', 'Fiat', 'VW', 'Honda'];
carros.splice(1, 0, 'Kia', 'Mustang'); // []
carros; // ['Ford', 'Kia', 'Mustang', 'Fiat', 'VW', 'Honda']

carros.splice(3, 2, 'Ferrari'); // ['Fiat', 'VW']
carros; // ['Ford', 'Kia', 'Mustang', 'Ferrari', 'Honda']
```

[].copyWithin()

`[].copyWithin(alvo, inicio, final)` a partir do alvo, ele irá copiar a array começando do inicio até o final e vai preencher a mesma com essa cópia. Caso omita os valores de início e final, ele irá utilizar como inicio o 0 e final o valor total da array.

```
['Item1', 'Item2', 'Item3', 'Item4'].copyWithin(2, 0, 3);
// ['Item1', 'Item2', 'Item1', 'Item2']

['Item1', 'Item2', 'Item3', 'Item4'].copyWithin(-1);
// ['Item1', 'Item2', 'Item3', 'Item1']
```

[].fill()

`[].fill(valor, inicio, final)` preenche a array com o valor, do início até o fim.

```
['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana');
// ['Banana', 'Banana', 'Banana', 'Banana']

['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana', 2);
// ['Item1', 'Item2', 'Banana', 'Banana']

['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana', 1, 3);
// ['Item1', 'Banana', 'Banana', 'Item4']
```

[].fill()

`[].fill(valor, inicio, final)` preenche a array com o valor, do início até o fim.

```
['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana');
// ['Banana', 'Banana', 'Banana', 'Banana']

['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana', 2);
// ['Item1', 'Item2', 'Banana', 'Banana']

['Item1', 'Item2', 'Item3', 'Item4'].fill('Banana', 1, 3);
// ['Item1', 'Banana', 'Banana', 'Item4']
```

Métodos de Acesso [].concat()

Os métodos abaixo não modificam a array original, apenas retornam uma array modificada. `[].concat()` irá concatenar a array com o valor passado.

```
const transporte1 = ['Barco', 'Aviao'];
const transporte2 = ['Carro', 'Moto'];
const transportes = transporte1.concat(transporte2);
// ['Barco', 'Aviao', 'Carro', 'Moto'];

const maisTransportes = [].concat(transporte1, transporte2,
'Van');
// ['Barco', 'Aviao', 'Carro', 'Moto', 'Van'];
```

[].includes(), [].indexOf() e [].lastIndexOf()

`[].includes(valor)` verifica se a array possui o valor e retorna true ou false. `[].indexOf(valor)` verifica se a array possui o valor e retorna o index do primeiro valor na array. Já o `[].lastIndexOf(valor)` retorna o index do último.

```
const linguagens = ['html', 'css', 'js', 'php', 'python',
'js'];

linguagens.includes('css'); // true
linguagens.includes('ruby'); // false
linguagens.indexOf('python'); // 4
linguagens.indexOf('js'); // 2
linguagens.lastIndexOf('js'); // 5
```

[].join()

`[].join(separador)` junta todos os valores da array e retorna uma string com eles. Se você passar um valor como parâmetro, este será utilizado durante a junção de cada item da array.

```
const linguagens = ['html', 'css', 'js', 'php', 'python'];
linguagens.join(); // 'html,css,js,php,python'
linguagens.join(' '); // 'html css js php python'
linguagens.join('-_-'); // 'html-_-css-_-js-_-php-_-python'

let htmlString = '<h2>Título Principal</h2>'
htmlString = htmlString.split('h2');
// ['<', '>Título Principal</', '>']
htmlString = htmlString.join('h1');
// <h1>Título Principal</h1>
```

[].slice()

`[].slice(inicio, final)` retorna os itens da array começando pelo início e indo até o valor de final.

```
const linguagens = ['html', 'css', 'js', 'php', 'python'];
linguagens.slice(3); // ['php', 'python']
linguagens.slice(1, 4); // ['css', 'js', 'php']

const cloneLinguagens = linguagens.slice();
```

Exercícios

```
const comidas = ['Pizza', 'Frango', 'Carne', 'Macarrão'];
// Remova o primeiro valor de comidas e coloque em uma variável
// Remova o último valor de comidas e coloque em uma variável
// Adicione 'Arroz' ao final da array
// Adicione 'Peixe' e 'Batata' ao início da array

const estudantes = ['Marcio', 'Brenda', 'Joana', 'Kleber',
'Julia'];
// Arrume os estudantes em ordem alfabética
// Inverta a ordem dos estudantes
// Verifique se Joana faz parte dos estudantes
// Verifique se Juliana faz parte dos estudantes

let html = `<section>
              <div>Sobre</div>
              <div>Produtos</div>
              <div>Contato</div>
            </section>`
// Substitua section por ul e div com li,
// utilizando split e join
```

```
// Remova o último carro, mas antes de remover
// salve a array original em outra variável
```