

IOS – Instituto de
Oportunidade Social

Aula JS 09 - Métodos de alto nível para manipular arrays



- > Método `forEach()`
- > Método `map()`
- > Método `filter()`
- > Método `find()`

IOS – Instituto de
Oportunidade Social

Método forEach()



> Método forEach()

O método **forEach()** executa uma determinada função para cada um dos elementos de um array. Ele não é executado em elementos vazios do array. A sintaxe do método é:

```
nome_do_array.forEach(callback(currentValue [, index [, array]]), thisArg);
```

Os parâmetros entre colchetes são opcionais e cada um significa:

- **callback**: é a função que será chamada para ser executada em cada elemento do array.
- **currentValue**: é o valor do elemento que está sendo processado no momento.
- **index (opcional)**: O índice do elemento atual sendo processado no array.
- **array (opcional)**: O array que forEach() está sendo aplicado.
- **thisArg (opcional)**: Valor a ser usado como this quando executar callback.

Método forEach()

- A instrução mostra como chamar o método forEach():

Nome do array	Ponto final	forEach	(função chamada)
frutas	.	forEach	(minhaFuncao)

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Método map()



> Método map()

O método **map()** cria um novo array com o resultado gerado pela chamada de uma função para cada elemento do array. Ele não é executado em elementos vazios do array. A sintaxe do método é:

```
let novo_array = nome_do_array.map(callback[, thisArg]);
```

Os parâmetros entre colchetes são opcionais e cada um significa:

- **callback**: é a função que o retorno produz o elemento do novo Array.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Método filter()



> Método filter()

O método **filter()** cria um novo array com o resultado do filtro aplicado pelo método, ou seja, os elementos que “passaram” na condição configurada no filtro. A sintaxe do método é:

```
let novo_array = nome_do_array.filter(callback[, thisArg]);
```

Os parâmetros entre colchetes são opcionais e cada um significa:

- **callback**: é a função com a condição para testar cada elemento do array.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Método find()



> Método find()

O método **find()** retorna o valor dos elementos do array que passarem na condição configurada. Semelhante ao método `filter()`, porém o `find()` não cria um novo array. A sintaxe do método é:

```
nome_do_array.find(callback(element[, index[, array]]), thisArg);
```

Os parâmetros entre colchetes são opcionais e cada um significa:

- **callback**: é a função com a iteração de cada elemento do array.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Para Aprender mais



Procure sempre aprender e estudar mais. Seguem alguns links para você estudar e aprender mais:

`forEach()`

https://developer.mozilla.org/ptBR/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

https://www.w3schools.com/jsref/jsref_foreach.asp

`map()`

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/map

https://www.w3schools.com/jsref/jsref_map.asp

`filter()`

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

https://www.w3schools.com/jsref/jsref_filter.asp

`find()`

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/find

https://www.w3schools.com/jsref/jsref_find.asp

Existem muito outros métodos para manipular arrays, vale a pena você conferir a lista nos links:

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array#

https://www.w3schools.com/jsref/jsref_obj_array.asp

IOS – Instituto de
Oportunidade Social

Exercício



Exercício 1

Crie um Array unidimensional com 8 posições contendo os seguintes valores 17, 43, 8, 4, 97, 56, 29 em cada posição irá conter um valor do tipo number.

- utilizando o método `forEach` crie uma função que receba cada valor do array e verifique se o número é par ou ímpar e exiba o resultado no `console.log`

Exercício 2

Crie um Array unidimensional com 14 posições contendo os seguintes valores 17, 43, 8, 4, 97, 56, 29, 3, 75, 34, 88, 82, 61 em cada posição irá conter um valor do tipo number.

- Utilizando Arrow Function com o método filter crie uma verificação para retornar somente os valores que estão entre o número 20 e 80.