**MAP**

Apresentação do curso

Oque é, Sintaxe e Map vs forEach

**MAP**

Objetivo = Conhecer a implementação do método map;

Y=f(x)

2

4

6

1

2

3

Mapping Function  
y=2x

Quando um conjunto passa por uma operação e se transforma em outro conjunto;

* Cria um novo array;
* Não modifica o array original;
* Realiza as operações em ordem;

**SINTAXE**

Escreve: “array.map(call-back, thisArg) onde, “callback(item, index, array)

Callback: Função a ser executada em cada elemento

thisArg(Opcional): valor de ‘this’ dentro da função de callback

**MAP vs forEach**

//Usando map

Const array = [1, 2, 3, 4, 5];

Array.map((item) => item \* 2); // retorno: [2, 4, 6, 8, 10];

//Usando forEach

Const array = [1, 2, 3, 4, 5];

Array.forEach((item) => item \* 2); // retorno: undefined

Diferenças:

* Valor de retorno;
* Considere se o array auxiliar será necessário;

Slide de apresentação

Slide da aula

**FILTER**

Filter e Sintaxe

**FILTER**

Array

Objetivo = Conhecer a implementação do método Filter;

Basicamente ele pega um array e passa pelo filter deixando apenas

Elementos selecionados:

Filter

* Cria um novo array;
* Não modifica o array original;

**SINTAXE**

Array.filter(call-back, thisArg)

Callback: Função a ser executada em cada elemento

thisArg(Opcional): valor de ‘this’ dentro da função de callback

**EXEMPLO**

Const frutas = [‘maçã fuji’ , ‘maçã verde’ , ‘laranja’ , ‘abacaxi’];

Frutas.filter((fruta) => fruta.includes(‘maçã’))

// retorno: [‘maçã fuji’ , ‘maçã verde’];

Slide da aula

**REDUCE**

Reduce e Sintaxe

**REDUCE**

Objetivo = Conhecer a implementação do método Reduce;

.reduce()

* Exececuta uma função em todos os elementos do array, retornando um valor único

**SINTAXE**

Array.reduce(callbackFn, initialValue)

Callback: Função a ser executada a partir do acumulador

initialValue(Opcional): valor sobre o qual o retorno final irá atuar

**EXEMPLO**

Const callbackFn = function (accumulator , currentValue , index , array) {

// do something

}

Array.reduce(callbackFn, initialValue)

Accumulator/prevValue(Opcional): acumulador de todas as chamadas de callbackFn

currentValue: elemento atual sendo acessado pela função

Slide da aula

**COLOCANDO EM PRÁTICA**

Aula prática

**Atividades**

**Map**

* Pratique a sintaxe de multiplicação de números, uma vez utilizando o parâmetro this de um objeto criado por você, e depois sem ele.

**Filter**

* Filtre e retorne todos os números pares de um array.

**Reduce**

1. Some todos os números de um array
2. Crie uma função que recebe uma lista de preços e um número representando o saldo disponível. Calcule qual será o saldo final após subtrair todos os preços da lista enviada.

##### Sobre o método “filter”, é correto afirmar que:

Ele sempre retorna um novo array

##### Sobre o método reduce, é correto afirmar que:

Ele sempre retorna apenas um valor, que pode ou não ser um array

##### Dado o código abaixo: let array = [1, 2, 3, 4]; array.reduce((current, acc) => current + acc); Qual será o retorno?

10

##### Sobre o método “map”, é correto afirmar que:

Ele sempre retorna um novo array

##### Qual a diferença entre os métodos map e forEach?

O retorno: o método map retorna um array, enquanto forEach retorna “undefined”

##### Sobre o método filter, é correto afirmar que:

Percorre um array e retorna outro contendo apenas os itens que passaram por determinada validação

##### Sobre o método map, é correto afirmar que:

Percorre o array e executa uma função em cada um de seus itens

##### Dado o código abaixo let array = [1, 2, 3, 4]; array.filter((item) => item % 2 === 0); Qual será o retorno?

[2, 4]

##### Sobre o método reduce, é correto afirmar que:

Percorre um array e acumula o valor de cada operação para retornar um valor único no final

##### Dado o código abaixo: let array = [1, 2, 3, 4]; array.map((item) => item \* 2); console.log(array); O que será logado no console?

[1, 2, 3, 4]