

IOS – Instituto de  
Oportunidade Social

## Aula JS 06 - Array



- > O que é Array
- > Array unidimensional
- > Arrays bidimensionais
- > Métodos de arrays
- > Convertendo Array em String
- > Propriedade: Tamanho do array
- > Inserir e remover elemento no array
- > Apagando um elemento em uma posição específica

IOS – Instituto de  
Oportunidade Social

O que é Array



**Arrays** são usados para **armazenar múltiplos valores** em **uma única variável**. **Arrays** podem ter mais de **uma ou mais dimensões** e, na literatura em português, é normal você encontrar o nome **vetor** para **referenciar array de uma dimensão** e **matrizes para array de duas dimensões**.

Resumindo, você pode chamar tudo de **array** que qualquer pessoa que programa irá entender.

# O que é Array

É importante saber que primeiro valor de um array possui índice zero ( 0 ), portanto para array unidimensional, temos os seguintes índices:



# O que é Array

Para **arrays bidimensionais**, temos os seguintes índices:

(0.0)				(0.4)
(1.0)				
(3.0)				(3.4)

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

## Array unidimensional



# Array unidimensional

Vamos ver como acessar elementos armazenados em um array unidimensional. No arquivo, main.js digite o seguinte código

Linha	[-2, 40, 16, 111, 33, 64]					
posição	0	1	2	3	4	5

> Vamos Praticar



IOS – Instituto de  
Oportunidade Social

## Arrays bidimensionais



# Arrays bidimensionais



Agora vamos ver como trabalhar com arrays bidimensionais. Nesse caso, o primeiro elemento vai estar na posição (0, 0). Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

```
Variável const matrix = [  
  Linha 1 ['Banana', 'Maçã', 'Pêra'],  
  posição    0          1          2  
  Linha 2 ['Laranja', true, 1],  
  Linha 3 [null, 'Uva', -350],  
];
```

> Vamos Praticar

# Arrays bidimensionais

Para alterar algum valor é bem simples, basta indicarmos o índice do elemento que queremos alterar e igualar ao valor desejado. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

```
let moedas = ['Real', 'Dólar', 'Coroa', 'Peso'];  
  
console.log(`Antes: ${moedas}`);  
moedas[2] = 'Yen';  
console.log(`Depois: ${moedas}`);
```

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

## Métodos de arrays



## > Convertendo Array em String

O método **toString()** converte um array em uma string, onde os elementos do array são separados por vírgulas. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
//Métodos em Arrays  
console.clear();  
let numArray = [1, 2, 3, 4];  
console.log(typeof numArray);  
  
let numArray2 = numArray.toString();  
console.log(typeof numArray2);
```

> Vamos Praticar

Existe também o método **join()** que converte o array em string, mas você pode colocar qualquer caractere como separador dos elementos. Vamos ver na prática:

```
// join()
console.clear();
let teste1 = [0, 1, 2, 3, 4, 5];
let teste2 = teste1.join(' *** ');
console.log(teste2);
console.log(typeof teste2);
```

> Vamos Praticar

## > Propriedade: Tamanho do array

A propriedade **length** retorna tamanho do array, ou seja, a quantidade de elementos que esse array possui. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
//Tamanho de Arrays  
console.clear();  
let array = [0, 1, 2, 3, 4];  
console.log(array.length);
```

> Vamos Praticar

## > Inserir e remover elemento no array

O método **pop()** remove o último elemento do array e retorna esse elemento retirado, que pode ser armazenado em uma variável. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
// Remover e inserir elementos
// pop()
console.clear();
const frutas1 = ['Banana', 'Laranja', 'Maçã', 'Manga'];
let x = frutas1.pop(); // x = "Manga"

console.log(frutas1);
console.log(x);
```

> Vamos Praticar



## > Inserir e remover elemento no array

Já o método **push()** insere um elemento no final do array e retorna o tamanho do novo array, que pode ser armazenado em uma variável.

```
// push()
console.clear();
const frutas2 = ['Banana', 'Laranja', 'Maçã', 'Manga'];
let y = frutas2.push('Kiwi'); // y = 5

console.log(frutas2);
console.log(y);
```

> Vamos Praticar

## > Inserir e remover elemento no array

Por sua vez, o método **shift()** remove o elemento no início do array deslocando todos os elementos em uma posição e retorna esse elemento retirado, que pode ser armazenado em uma variável.

```
// shift()
console.clear();
const frutas3 = ['Banana', 'Laranja'];
let z = frutas3.shift(); // z = 'Banana'
console.log(frutas3);
console.log(z);
```

> Vamos Praticar

## > Inserir e remover elemento no array

Por fim, o método **unshift()** insere um elemento no início do array e retorna o tamanho do novo array, que pode ser armazenado em uma variável.

```
// unshift()
console.clear();
const frutas4 = ['Banana', 'Laranja'];
let w = frutas4.unshift('Kiwi'); // w = 3

console.log(frutas4);
console.log(w);
```

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Apagando um elemento em uma  
posição específica



# Apagando um elemento em uma posição específica



O comando delete apaga um elemento na posição desejada e altera para slot vazio, que é o mesmo de undefined.

Vamos ver isso na prática, siga os passos para continuar a implementação:

```
// delete
console.clear();
const frutas5 = ['Banana', 'Laranja', 'Maçã', 'Manga'];
delete frutas5[2];

console.log(frutas5);
```

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Conclusão



Existem muitos outros métodos para manipular arrays e gastaríamos muito tempo se quiséssemos abordar todos esses métodos. À medida que outros métodos forem aparecendo, vamos explicando cada uma dessas. O importante é que você sempre deve usar a internet para procurar soluções em programação, mas não apenas copie e cole uma solução, um bom programador procura entender a solução proposta e adaptá-la no seu projeto. Para saber mais: Você pode consultar a lista de métodos de strings nos links:

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array) [https://www.w3schools.com/js/js\\_array\\_methods.asp](https://www.w3schools.com/js/js_array_methods.asp)

IOS – Instituto de  
Oportunidade Social

Exercício





Criar uma página array.html com seu respectivo arquivo JS com os exercícios abaixo, mostrando cada passo no console do navegador:

1 - Crie um Array Unidimensional com as seguintes informações.

**HTML, 1993, CSS, 1996, Bootstrap, 2011, JS, 1995, React, 2013, Java, 1995**

- Retorne todo o array;
- O tamanho do Array;
- Retorne apenas a informação **JS**;
- transforme o array em uma String e retorne o tipo no console;
- Troque o separador (,) pelo separador / e retorne a mudança no console;
- Adicione a informação **IOS** no **início do Array** e faça o retorno do novo array com a quantidade de posições atualizadas;

**2 - Crie um Array bidimensional** com as seguintes informações.

**HTML, 1993, CSS, 1996, Bootstrap, 2011, JS, 1995, React, 2013, Java, 1995**

Esse array deverá conter **três dimensões**, sendo assim deve conter **3 linhas** com **4 posições** cada linha.

- Retorne todo o array;
- O tamanho do Array;
- Retorne apenas o ano **2013**;
- Troque a informação **JS** por **JavaScript**;
- Remova a informação **Java**;
- Insere a informação **pipoca** no **final do array**;