

# Programador Web

Algoritmos e JavaScript - Arrays

Adriel Sales



# JavaScript: Variáveis vs Arrays

JS

**Até agora as variáveis nos serviram muito bem, mas...**



# Arrays em JavaScript

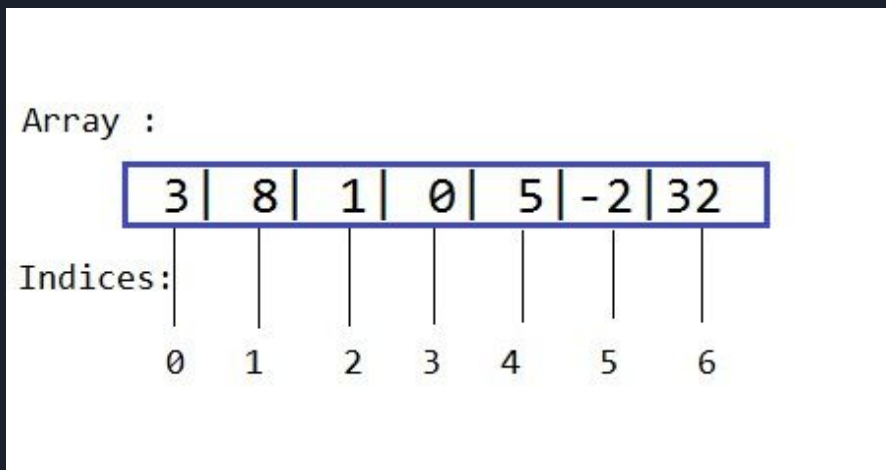


- Em programação de computadores, uma **matriz** (em inglês **array**) é uma estrutura de dados que armazena uma coleção de elementos de tal forma que cada um dos elementos possa ser identificado por, pelo menos, um índice ou uma chave. ([https://pt.wikipedia.org/wiki/Arranjo\\_\(computação\)](https://pt.wikipedia.org/wiki/Arranjo_(computação))).
- Em uma variável de tipo array, podemos armazenar diversos valores com variados tipos.
- Então, um array é uma variável especial, que pode conter mais de um valor de cada vez.

# Arrays em JavaScript



Geralmente, estão estruturados com índices e seus respectivos valores:



# Arrays em JavaScript



Declaração:

```
var array_name = [item1, item2, ...];
```

```
var meuArray = [  
  "João",  
  "Maria",  
  "José",  
  "Pedro",  
  "Roberto"  
];
```

```
var meuArray = ["João", "Maria", "José", "Pedro", "Roberto"];
```

# Arrays em JavaScript



Tipos diferentes também são permitidos em um mesmo array:

```
var meuArray = [true, "Maria", 123, "Pedro", 15.55];
```



Boolean | String | int | String | float

# Arrays em JavaScript



Seus valores serão acessados a partir de seus índices:

```
var meuArray = [true, "Maria", 123, "Pedro", 15.55];
```

```
alert(meuArray[0]); // true  
alert(meuArray[1]); // Maria  
alert(meuArray[2]); // 123  
alert(meuArray[3]); // Pedro  
alert(meuArray[4]); // 15.55
```

```
meuArray[2] = "novo valor"; // substituindo um valor
```

# Arrays em JavaScript



## Métodos importantes para arrays:

```
var meuArray = ["Maria", 123, "Pedro", 15.55, false];
```

```
console.log(meuArray);
```



```
alert(meuArray.toString()); // retorna: Maria,123,Pedro,15.55,false
```

```
var juntaArray = meuArray.join(" - ");
```

```
alert(juntaArray); // resulta em: Maria - 123 - Pedro - 15.55 - false
```

```
var meuArray = ["Maria", 123, "Pedro", 15.55, false, "Pedro"];
```

```
alert(meuArray.indexOf("Pedro")); // encontra o índice da primeira ocorrência
```

```
alert(meuArray.lastIndexOf("Pedro")); // encontra o índice da última ocorrência
```

```
alert(meuArray.reverse()); // inverte a posição dos elementos
```



# Arrays em JavaScript



## Métodos importantes para arrays:

```
var meuArray = [true, "Maria", 123, "Pedro", 15.55];

var tamanhoDoArray = meuArray.length; // retorna 5
alert(tamanhoDoArray);
var ordenandoArray = meuArray.sort(); // retorna o array ordenado
alert(ordenandoArray); // imprime: 123,15.55,Maria,Pedro,true

meuArray.push("Programação"); //adiciona um elemento no final
meuArray.pop(); // remove e retorna o último elemento
meuArray.shift(); // remove e retorna o primeiro elemento
meuArray.unshift(); // adiciona um elemento no início
Array.isArray(meuArray); // retorna true se for tipo Array
```

# Arrays em JavaScript



Alterando conteúdo de um elemento no array:

```
var carros = ["Fusca", "Opala", "Kombi"];  
var indiceDoCarro = carros.indexOf("Opala");  
carros[indiceDoCarro] = "FIAT-147";  
console.log(carros);  
//imprime: "Fusca", "FIAT-147", "Kombi"
```

# Arrays em JavaScript



Alguns métodos retornam outros arrays:

**concat** (concatenação de arrays)

```
var carros = ["Fusca", "Opala", "Kombi"];  
var novoArray = carros.concat("Honda Civic", "HB20");  
console.log(novoArray);  
//imprime: "Fusca", "Opala", "Kombi", "Honda Civic", "HB20"
```

# Arrays em JavaScript



Alguns métodos retornam outros arrays:

**slice** (retorna uma parte do array, a partir de um ou dois índices)

```
var carros = ["Fusca", "Brasília", "Opala", "Kombi"];
var carrosAposUmIndice = carros.slice(2);
console.log(carrosAposUmIndice);
//imprime: ["Opala", "Kombi"]
var carrosAposUmIndice = carros.slice(1,3);
console.log(carrosAposUmIndice);
//imprime: ["Brasília", "Opala"]
```

# Arrays em JavaScript



Alguns métodos retornam outros arrays:

**splice** (retorna parte removida do array, iniciada a partir de um índice)

```
var carros = ["Fusca", "Brasília", "Opala", "Kombi"];  
var carrosRemovidos = carros.splice(1,2,"FIAT-147","Honda Civic", "HB20");  
console.log(carrosRemovidos);  
//imprime:"Brasília", "Opala"  
console.log(carros);  
//Array Original: ["Fusca", "FIAT-147", "Honda Civic", "HB20", "Kombi"]
```

(1) Início | (2) Quantidade de itens removidos | Itens que serão adicionados.

# Array Multidimensional



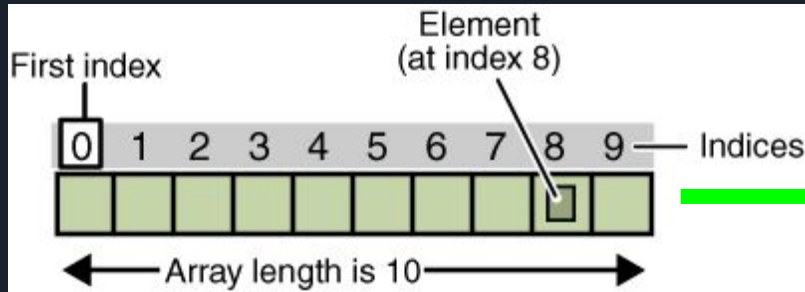
São arrays que recebem outros arrays como elementos.

Estados do Nordeste	
Linha nº	UF
0	BA
1	CE
3	MA
4	PB
5	PE
6	PI
7	RN
8	SE

# Array Multidimensional

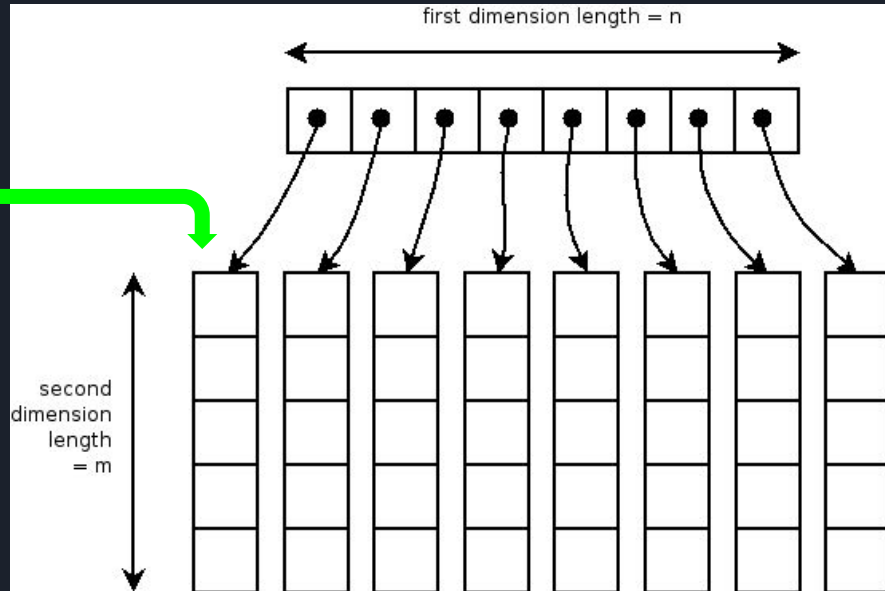
JS

São arrays que recebem outros arrays como elementos.



Array simples: índice e valor.

Array com outros arrays como elementos.



# Array Multidimensional



São arrays que recebem outros arrays como elementos.

Estados do Nordeste			
Linha nº	UF	Nome	Capital
0	BA	Bahia	Salvador
1	CE	Fortaleza	Ceará
3	MA	Maranhão	São Luís
4	PB	Paraíba	João Pessoa
5	PE	Pernambuco	Recife
6	PI	Piauí	Teresina
7	RN	Rio Grande do	Natal
8	SE	Sergipe	Aracajú



```
var estadosNE = [  
  ["AL", "Alagoas", "Maceió"],  
  ["BA", "Bahia", "Salvador"],  
  ["CE", "Ceará", "Fortaleza"],  
  ["MA", "Maranhão", "São Luís"],  
  ["PB", "Paraíba", "João Pessoa"],  
  ["PE", "Pernambuco", "Recife"],  
  ["PI", "Piauí", "Teresina"],  
  ["RN", "Rio Grande do Norte", "Natal"],  
  ["SE", "Sergipe", "Aracajú"]  
]
```



# Array Multidimensional

JS

```
var estadosNE = [
  ["AL", "Alagoas", "Maceió"],
  ["BA", "Bahia", "Salvador"],
  ["CE", "Ceará", "Fortaleza"],
  ["MA", "Maranhão", "São Luís"],
  ["PB", "Paraíba", "João Pessoa"],
  ["PE", "Pernambuco", "Recife"],
  ["PI", "Piauí", "Teresina"],
  ["RN", "Rio Grande do Norte", "Natal"],
  ["SE", "Sergipe", "Aracajú"]
]

for (var i = 0; i < estadosNE.length; i++) {
  for (var j = 0; j < estadosNE[i].length; j++) {
    if (estadosNE[i][j] == "PB") {
      imprime("<h1 class='alert alert-danger'>" + estadosNE[i][j] + " na linha: " + i + ", célula: " + j + "</h1>");
    }
    imprime("<h4 class='alert alert-info'> <b>" + estadosNE[i][j] + "</b> na linha: " + i + ", célula: <b>" + j + "</b></h4>");
  }
  imprime("<br>");
}
```



## REFERÊNCIAS:

LACERDA, Ivan Max Freire de, OLIVEIRA, Ana Liz Souto.

Programador Web: um guia para programação e manipulação de banco de dados.

Rio de Janeiro: Senac Nacional, 2013.

W3schools: <http://www.w3schools.com>

Slide: Algoritmo e lógica de programação:

[https://pt.slideshare.net/engenhariadecomputacao/algoritmo-e-logica-de-programao-aula-1?qid=4b7e1bbc-c2c4-4e0e-b267-e50d0867c402&v=&b=&from\\_search=2](https://pt.slideshare.net/engenhariadecomputacao/algoritmo-e-logica-de-programao-aula-1?qid=4b7e1bbc-c2c4-4e0e-b267-e50d0867c402&v=&b=&from_search=2)

Slide: Material de apoio de algoritmo e lógica de programação:

[https://pt.slideshare.net/rodfernandes/material-de-apoio-de-algoritmo-e-logica-de-programao?qid=d0e7d40-1d80-4a6b-a730-d5da859fe9f1&v=&b=&from\\_search=1](https://pt.slideshare.net/rodfernandes/material-de-apoio-de-algoritmo-e-logica-de-programao?qid=d0e7d40-1d80-4a6b-a730-d5da859fe9f1&v=&b=&from_search=1)

Apontamentos da Aula:

<https://github.com/adrielsales/senac/wiki/Aulas-Senac>