

# Rotinas de Programação (Arrays)

APRESENTAR OS CONCEITOS DE ARRAY COM EXEMPLOS PRÁTICOS EM JAVASCRIPT

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

## Array

O elemento array, ou vetor, é uma variável sequencial de dados, enumerados por um índice. A praticidade de se utilizar um array se dá pela simplificação do código, o que torna o seu uso uma grande vantagem, em relação da utilização de diversas variáveis diferentes.

Ainda, além do array “tradicional” de uma dimensão, podemos ter arrays multidimensionais, representando elementos matemáticos como as matrizes.

## Array unidimensional

Um array unidimensional pode ser inicializado de maneiras diferentes. Vejamos:

```
1. <script type="text/javascript">
2.     var x [];
3.     var x1 = [1,2,3,4,5];
4.     var x2 = new Array(1,2,3,4,5);
5.     var x3 = Array = [1,2,3,4,5];
6. </script>
```

Neste fragmento de código podemos verificar três maneiras de se criar e inicializar os arrays. No primeiro exemplo é possível criar o array, sem inicializa-lo.

Para se utilizar o array ou exibir os seus valores armazenados, devemos informar o valor do índice corretamente. Vamos analisar o código a seguir:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 7)</div>
9.         <script type="text/javascript">
10.             var x1 = [1,2,3,4,5];
11.             var x2 = new Array(1,2,3,4,5);
12.             var x3 = Array = [1,2,3,4,5];
13.
14.             for (var i = 0; i < 5; i++){
15.                 document.write("<br>x1 = " + x1[i]);
16.                 document.write("&emsp; x2 = " + x2[i]);
17.                 document.write("&emsp; x3 = " + x3[i]);
18.             }
19.         </script>
20.     </body>
21. </html>
```

Neste exemplo, utilizamos uma estrutura de repetição for() para “varrer”, ou percorrer os valores armazenados, em cada uma das suas posições de índice.

## DICA:

O primeiro índice de um array é o valor zero, array[0]. Tenha muita atenção para não se esquecer deste padrão, pois caso você inicie de uma forma [0] e utilize os valores de um array de outra maneira [1], os dados não farão sentido, pois apontariam para valores diferentes. Muito cuidado!

Uma vantagem dos arrays em Javascript é que os tipos de dados podem ser alternados, em um mesmo vetor, ou array, por exemplo.

```
1. <script type="text/javascript">
2.     var valor = [1, "dois", 3.1, "4", 5.55];
3. </script>
```

Esta vantagem deve ser utilizada com cautela e muita atenção, afim de se evitar erros enquanto estes dados estiverem sendo manipulados.

Outra forma de inicializar os valores é utilizando uma estrutura de repetição, veja o exemplo:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 7)</div>
9.         <script type="text/javascript">
10.             var x = [];
11.             for (var i = 0; i < 10; i++){
12.                 x[i] = i+1;
13.             }
14.
15.             for (var i = 0; i < 10; i++){
16.                 document.write("<br>x = " + x[i]);
17.             }
18.         </script>
19.     </body>
20. </html>
```

Observe que no primeiro laço o array x[i] foi inicializado, enquanto que no segundo laço, os valores com os respectivos índices foram impressos no arquivo html.

Veremos um exemplo de utilização de array com tipos String:

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 7)</div>
9.         <script type="text/javascript">
10.             var x = ["DOMINGO", "SEGUNDA", "TERÇA", "QUARTA", "QUINTA", "SEXTA", "SÁBADO"];
11.             for (var i = 0; i < 7; i++){
12.                 document.write(x[i] + " é o " + (i+1) + "º dia da semana!<br>");
13.             }
14.         </script>
15.     </body>
16. </html>

```

## Funções

Em alguns casos, obter informações sobre o array pode ser muito trabalhoso. Para facilitar este trabalho, diversas funções ou propriedades estão disponíveis para nos auxiliar. Vamos verificar algumas delas:

Propriedade `length()`

Informa o tamanho de um array, de acordo com a quantidade de elementos existentes.

```

1. <script type="text/javascript">
2.     var x = ["DOMINGO", "SEGUNDA", "TERÇA", "QUARTA", "QUINTA", "SEXTA", "SÁBADO"];
3.     document.write("O array x possui " + x.length + " valores de índices");
4.     // Resultado: O array possui 7 valores de índices
5. </script>

```

Ou ser utilizado em um `for()`, como determinar o fim do elemento array:

```

1. <script type="text/javascript">
2.     var x = ["DOMINGO", "SEGUNDA", "TERÇA", "QUARTA", "QUINTA", "SEXTA"];
3.     for (var i = 0; i < x.length; i++){
4.         document.write(x[i] + ". ");
5.     }
6. </script>

```

#### Método push()

O método push adiciona um elemento, ou valor, ao final do array. Veja o exemplo:

```

1. <script type="text/javascript">
2.     var x = [1,2,3];
3.     x.push(4);
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " ");
6.         // Resultado: 1 2 3 4
7.     }
8. </script>

```

#### Método join()

O método Join adiciona uma String entre os elementos de um array, compondo um novo array com os elementos adicionais. Veja o exemplo:

```

1. <script type="text/javascript">
2.     var x = [1,2,3];
3.     var novaLista = x.join(' / ');
4.     document.write(novaLista);
5.     //Resultado: 1 / 2 / 3
6. </script>

```

#### Método pop()

O método pop retira o último elemento de um determinado array. Veja o exemplo:

```
1. <script type="text/javascript">
2.     var x = [1,2,3];
3.     x.pop();
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " ");
6.         // Resultado: 1 2
7.     }
8. </script>
```

Método shift()

O método shift retira o primeiro elemento de um array. Veja o exemplo:

```
1. <script type="text/javascript">
2.     var x = [1,2,3];
3.     x.shift();
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " ");
6.         // Resultado: 2 3
7.     }
8. </script>
```

Método unshift()

O método unshift() adiciona elementos ao início de um array. Veja o exemplo:

```
1. <script type="text/javascript">
2.     var x = [1,2,3];
3.     x.unshift(7,8);
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " ");
6.         // Resultado: 7 8 1 2 3
7.     }
8. </script>
```

Método splice()

O método splice() seleciona um intervalo onde os valores irão compor um novo array, indicando o índice inicial e final ao novo array. Observe que em um array o índice se inicia em zero. Veja o exemplo:

```

1. <script type="text/javascript">
2.     var x = ['a','b','c','d','e','f','g'];
3.     //índice: 0 , 1, 2, 3, 4, 5, 6
4.     x = x.slice(1,5);
5.     for (var i = 0; i < x.length; i++){
6.         document.write(x[i] + " ");
7.         // Resultado: b c d e
8.     }
9. </script>

```

#### Método sort()

O método sort() ordena os valores de um array. Tenha cuidado com os valores a serem ordenados, pois são analisados os primeiros caracteres ou dígitos de cada elemento. Veja o exemplo:

```

1. <script type="text/javascript">
2.     var x = ['3','7','4','10','6','2','5'];
3.     x=x.sort();
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " - ");
6.         // Resultado: 10 - 2 - 3 - 4 - 5 - 6 - 7 -
7.     }
8. </script>

```

Neste exemplo o valor 10 surge antes do valor 2, devido ao primeiro carácter de 10, valor 1, ser “menor” do que o primeiro carácter do próximo valor, valor 2.

#### Método concat()

O método concat() concatena, ou adiciona, uma nova sequência a um array. Veja o exemplo:

```

1. <script type="text/javascript">
2.     var x = [1,2,3,4];
3.     x = x.concat('um', 'dois', 'três', 'quatro');
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + ", ");
6.         // Resultado: 1, 2, 3, 4, um, dois, três, quatro,
7.     }
8. </script>

```

#### Método reverse()

O método `reverse()` inverte a ordem dos elementos de um array. Veja o exemplo:

```
1. <script type="text/javascript">
2.     var x = [1,2,3,4];
3.     x.reverse();
4.     for (var i = 0; i < x.length; i++){
5.         document.write(x[i] + " ");
6.         // Resultado: 4 3 2 1
7.     }
8. </script>
```

## Array multidimensional

A utilização de array multidimensionais se dá com a criação de um array “dentro” de outro array. A ideia é a mesma da utilização de uma matriz, onde possuímos linhas e colunas.

Veja este exemplo:

```
1. <script type="text/javascript">
2.
3.     var m = new Array(3);           // Criar um array de 3 elementos
4.     for (var L = 0; L < 3; L++){
5.
6.         m[L] = new Array(3);       // Cria um novo array de 3 elementos
7.         for (var C = 0; C < 3; C++){
8.             m[L][C] = (L + C + 1); // Atribui valores a cada posição do a
9.         }
10.    }
11.    //Exibi os valores da matriz
12.    for (var L = 0; L < 3; L++){
13.        for (var C = 0; C < 3; C++){
14.            document.write "[" + m[L][C] + " ] ";
15.        }
16.        document.write("<br>");
17.    }
18. </script>
```



A imagem a seguir apresenta o resultado do array multidimensional:

```
Programação de Interfaces (aula 7)
[1] [2] [3]
[2] [3] [4]
[3] [4] [5]
```

No exemplo a seguir, temos um array já inicializado, e a exibição de apenas os elementos da coluna 0. Veja o código de exemplo:

```
1. <script type="text/javascript">
2.     var m = [
3.         [1,2,3],
4.         [4,5,6],
5.         [7,8,9]
6.     ];
7.     for (L = 0; L < 3; L++){
8.         document.write(m[L][0] + " ");
9.         // Resultado: 1 4 7
10.    }
11. </script>
```

## DICA:

Para saber mais sobre arrays, veja em:

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp) ([https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp))

[https://www.w3schools.com/js/js\\_array\\_sort.asp](https://www.w3schools.com/js/js_array_sort.asp) ([https://www.w3schools.com/js/js\\_array\\_sort.asp](https://www.w3schools.com/js/js_array_sort.asp))

[https://www.w3schools.com/js/js\\_array\\_methods.asp](https://www.w3schools.com/js/js_array_methods.asp)

([https://www.w3schools.com/js/js\\_array\\_methods.asp](https://www.w3schools.com/js/js_array_methods.asp))

## IMPORTANTE:

Neste tópico apresentamos:

- Criação e uso de arrays.
- Utilização de propriedades e métodos aos arrays.

- Criação e uso de array multidimensionais.

## Conclusão

Nesta aula abordamos o uso de arrays e suas propriedades e métodos. Utilizar array pode ser muito útil e em dados momentos, será a única opção de se trabalhar com determinados dados.

## ATIVIDADE

Escolha a alternativa que apresente o resultado do seguinte código:

```
<script type="text/javascript">  
    var x = [1,2,3];  
    var novaLista = x.join(' - ');  
    document.write(novaLista);  
</script>
```

- A. 1 - 2 - 3 -
- B. 1 - 2 - 3
- C. 1 / 2 / 3
- D. 1 - 2   3

## ATIVIDADE

Escolha a opção correta ao seguinte código:

```
<script type="text/javascript">  
    var x = [1,2,3];  
    x.unshift(4,5);  
    for (var i = 0; i < x.length; i++){  
        document.write(x[i] + " ");
```

```
}
```

```
</script>
```

A. 5 1 2 3 4

B. 5 4 3 2 1

C. 1 2 3 4 5

D. 4 5 1 2 3

## ATIVIDADE

Escolha a alternativa correta ao seguinte código:

```
<script type="text/javascript">
```

```
var x = [1,2,3];
```

```
x.push(4);
```

```
for (var i = 0; i < x.length; i++){
```

```
    document.write(x[i] + " ");
```

```
}
```

```
</script>
```

A. 4 1 2 3

B. 1 2 3 4

C. 4 3 2 1

D. 4 3 1 2

## REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO, C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.





