

AULA 2 – ARRAYS MULTIDIMENSIONAIS

OBJETIVO DA AULA

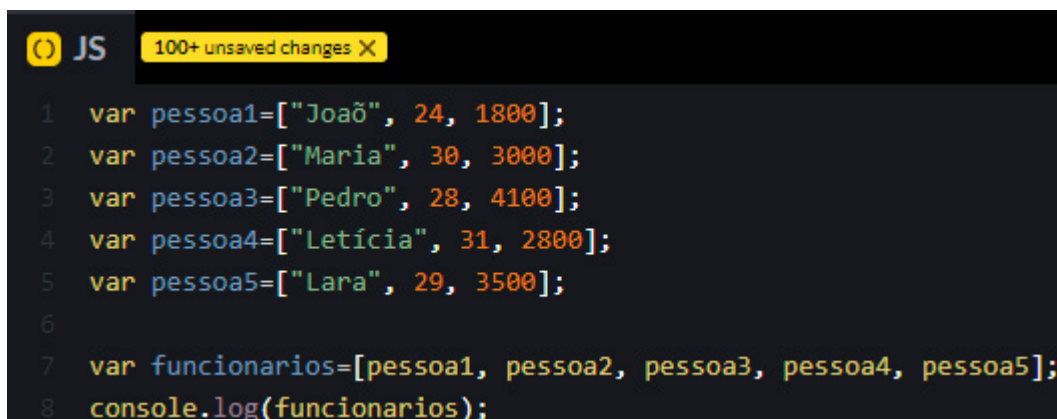
Entender o que é, para que serve e como manipular um array multidimensional em JavaScript.

APRESENTAÇÃO

Na aula passada aprendemos que um array nada mais é do que uma estrutura utilizada para armazenar e organizar valores em uma única variável. Nesta aula abordaremos o conceito de array multidimensional, ou seja, um array com múltiplas dimensões. Calma! Não se assuste! Um array multidimensional nada mais é do que um array que armazena outro array. É isso mesmo! Em outras linguagens de programação, um array de array é chamado de matriz. Não podemos esquecer que um array no *JavaScript* é um tipo especial de objeto que trabalha apenas com índices numéricos. Vamos juntos aprender como criar um array de array e como adicionar e remover elementos do mesmo.

1. ARRAYS MULTIDIMENSIONAIS

“*JavaScript* não suporta arrays multidimensionais de verdade, mas é possível ter algo parecido, como array de array” (FLANAGAN, 2013). Não é nada muito diferente do que você já sabe, por exemplo, para acessar um elemento de um array simples (unidimensional) você utiliza um único colchete [] (meuArray[x]) e para acessar um elemento de um array de array você deve utilizar o colchete [] duas vezes (meuArray[x][y]). Assim como no array simples, a forma mais fácil de criar um array de array é usar a notação literal de array. Vamos ver duas formas diferentes de criar um array com múltiplas dimensões. A primeira forma é:



```

JS 100+ unsaved changes X
1 var pessoa1=["João", 24, 1800];
2 var pessoa2=["Maria", 30, 3000];
3 var pessoa3=["Pedro", 28, 4100];
4 var pessoa4=["Letícia", 31, 2800];
5 var pessoa5=["Lara", 29, 3500];
6
7 var funcionarios=[pessoa1, pessoa2, pessoa3, pessoa4, pessoa5];
8 console.log(funcionarios);
  
```

```

Console

// [object Array] (5)
* [// [object Array] (3)
  ["Joaõ",24,1800],// [object Array] (3)
  ["Maria",30,3000],// [object Array] (3)
  ["Pedro",28,4100],// [object Array] (3)
  ["Letícia",31,2800],// [object Array] (3)
  ["Lara",29,3500]]

```

Criar um array (*linha 7*) que vai receber todos os arrays unidimensionais criados anteriormente (*linhas 1-5*), ou seja, o array funcionário agrupa todos os arrays pessoa. Agora vamos ver a segunda forma:

```

JS 100+ unsaved changes X
1 var funcionarios=[
2   ["Joaõ", 24, 1800],
3   ["Maria", 30, 3000],
4   ["Pedro", 28, 4100],
5   ["Letícia", 31, 2800],
6   ["Lara", 29, 3500],
7 ];
8 console.log(funcionarios);
9
Console
// [object Array] (5)
* [// [object Array] (3)
  ["Joaõ",24,1800],// [object Array] (3)
  ["Maria",30,3000],// [object Array] (3)
  ["Pedro",28,4100],// [object Array] (3)
  ["Letícia",31,2800],// [object Array] (3)
  ["Lara",29,3500]]

```

Aqui temos o array funcionários funcionando como uma array multidimensional e produzindo o mesmo resultado do exemplo anterior. Este resultado pode ser mostrado em um formato de tabela, para isso, basta usar o método `console.table()` ao invés do `console.log()`. Veja:

```

1 var funcionarios=[
2   ["Joaõ", 24, 1800],
3   ["Maria", 30, 3000],
4   ["Pedro", 28, 4100],
5   ["Letícia", 31, 2800],
6   ["Lara", 29, 3500],
7 ];
8 console.table(funcionarios);

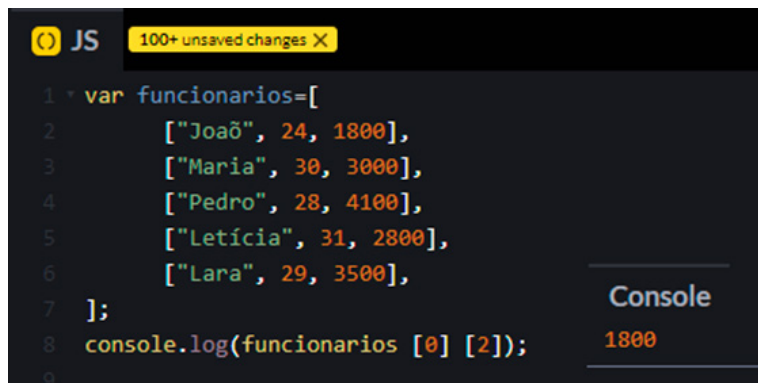
```

(index)	0	1	2
0	'Joaõ'	24	1800
1	'Maria'	30	3000
2	'Pedro'	28	4100
3	'Letícia'	31	2800
4	'Lara'	29	3500

Esta visualização conta os índices que representam cada um dos elementos presentes no array de arrays.

1.1. COMO ACESSAR UM ARRAY DE ARRAYS?

Podemos acessar os elementos de array de arrays usando a notação simples baseada em índice ou usando uma estrutura de repetição, assim como no array unidimensional. Veja:



```

1 * var funcionarios=[
2     ["João", 24, 1800],
3     ["Maria", 30, 3000],
4     ["Pedro", 28, 4100],
5     ["Letícia", 31, 2800],
6     ["Lara", 29, 3500],
7 ];
8 console.log(funcionarios [0] [2]);
9

```

Console
1800

Neste exemplo, estamos acessando o salário do João (funcionários [0] [2]), onde **[0]** seleciona a primeira linha e **[2]** seleciona o terceiro elemento dessa primeira linha que é 1800.

Veja agora como percorrer o vetor usando uma estrutura de repetição:



```

1 * var funcionarios=[
2     ["João", 24, 1800],
3     ["Maria", 30, 3000],
4     ["Pedro", 28, 4100],
5     ["Letícia", 31, 2800],
6     ["Lara", 29, 3500],
7 ];
8 * for(var i=0; i<funcionarios.length; i++){
9     for(var j=0; j<funcionarios[i].length; j++){
10         console.table(funcionarios[i][j]);
11     }
12 }

```

Console	
"João"	
24	
1800	"Letícia"
"Maria"	31
30	2800
3000	"Lara"
"Pedro"	29
28	3500
4100	

Esta versão é um pouco mais complexa, pois estamos fazendo uso de uma estrutura de repetição aninhada ou encadeada, isto significa que para cada rodada do primeiro *for*, o segundo é executado por completo. O primeiro *for* percorre os elementos do array externo e o segundo *for* percorre os elementos internos do array interno. Precisamos de duas estruturas de repetição aninhadas porque um array de arrays possui múltiplas linhas. Funciona mais ou menos assim: o primeiro *for* chega na primeira linha e o segundo passa por todos os elementos da linha 1, em seguida o primeiro *for* passa para a segunda linha e o segundo passa por todos os elementos da segunda linha e assim sucessivamente.



PRA PRATICAR

Os métodos: `push()`, `pop()`, `shift()`, `unshift()` e `splice()` também podem ser utilizados para manipular arrays multidimensionais. Sendo assim, deixo como desafio para você tentar incluir e excluir elementos em um array de arrays.

CONSIDERAÇÕES FINAIS

Nesta aula você aprendeu que o *JavaScript* não fornece o array multidimensional nativamente, para contornar esta situação temos que criar um array de arrays. Certamente, isso não é um problema para você, não é mesmo? Na aula anterior aprendemos como criar e manipular arrays unidimensionais. Então, aqui é quase a mesma coisa, a diferença está na quantidade de índices que cada elemento da nossa estrutura terá como identificado, por exemplo, `meuArray [5][5]` tem duas dimensões e cada dimensão possui 5 elementos. Logo, temos uma estrutura 5x5, capaz de armazenar 25 elementos distintos. Na próxima aula falaremos sobre funções. Te espero!

MATERIAIS COMPLEMENTARES

Curso JavaScript|Array Multidimensional – Matrizes|Coffee Tag:

<https://youtu.be/F1SIGxIQ-90>

Curso JavaScript|Adicionar um elemento a uma matriz Multidimensional|Coffee Tag:

<https://youtu.be/8Yz2csLjZAk>

Curso JavaScript|Deletar um valor da Array ou Matriz Multidimensional|Coffee Tag:

https://youtu.be/P8RIL_IlkOE

REFERÊNCIAS

FLANAGAN, David. *JavaScript: O guia definitivo*. Porto Alegre, RS: Bookman, 2013.