Faculdade de Tecnologia e Inovação SENAC

Curso de Análise e Desenvolvimento de Sistemas

Métodos de Arrays no JavaScript.

**Autores:** Isack Freitas Pinheiro Duarte, Luiz Henrique Maciel de Oliveira, Weslley Ferreira de Carvalho, Elisângela Silvestre Santana, Kaylon Gomes Rodrigues e Igor Cardoso Nunes.

**Professor Orientador:** Valdemir dos Santos Silva

Brasília, 2024.

2

**SUMÁRIO**

**1. INTRODUÇÃO............................................................................................3**

**2. DESENVOLVIMENTO..................................................................................4**

**3. CONCLUSÃO.............................................................................................8**

**4. FONTES BIBLIOGRÁFICAS..........................................................................9**

**3**

**INTRODUÇÃO**

**Arrays, também conhecidos como variáveis compostas, são fundamentais para a programação em qualquer linguagem que possamos imaginar. Nesse estudo abordaremos o que são arrays, o que são métodos e trabalharemos com alguns métodos para arrays específicos do Javascript que nos ajudam a manipular e tratar dados.**

**Como parte do projeto, também abordaremos alguns exemplos dos métodos .concat(), .copyWhiting e .flat para entendermos o nível de profundidade que a utilização desses métodos pode beneficiar o trabalho com arrays no JS.**

**4**

**DESENVOLVIMENTO**

1. **O que são arrays?**
   1. **- Definição**

Arrays são variáveis compostas. São formas de salvar mais de um item em uma variável. O vetor, como é chamado em português, também pode ser entendido como sendo uma lista de objetos, guardados na mesma variável.

Arrays são organizados por índices que se iniciam do 0 e podem ir até o infinito, sendo limitado somente pela capacidade da memória. Os índices são formas de identificar as “posições” dos itens do vetor e servem para trabalhar com a variável composta de diferentes formas. Uma das formas de manipulá-los é através de métodos, como veremos a seguir.

1. **Métodos**
   1. **- O que são e para que servem?**

Métodos são blocos de códigos pré-programados. No Javascript, métodos são funções associadas a um objeto que possibilita que eles executem ações.

* 1. **– Métodos que serão abordados**

Neste trabalho abordaremos três métodos associados a arrays. Serão eles: .concat() , .copyWhiting() e .flat() .

1. **Métodos em Arrays**
   1. **– Array.concat():**

Esse método é utilizado para concatenar dois ou mais arrays em apenas um. O .concat não altera os arrays principais, somente aglutina-os em um novo.

**5**

**Texto

Descrição gerada automaticamente**

Na foto acima vemos o seguinte, a constante “array1” que possui os valores: ['a', 'b', 'c'] e a constante “array2” que possui os valores: ['d', 'e', 'f'] estão sendo aglutinadas através do .concat em uma nova variável constante chamada “array3”. Os novos valores de “array3” passaram a ser: ["a", "b", "c", "d", "e", "f"].

**Interface gráfica do usuário, Texto

Descrição gerada automaticamente**

Nesse outro exemplo podemos ver que mais de um array pode ser aglutinado, seguindo a mesma lógica temos os arrays: “num1”, “num2” e “num3” sendo aglutinados na const “números”. Podemos perceber também a sintaxe padrão do método, que é:

**array0.concat(array1, array2, ... , arrayN);.**

* 1. **– Array.copyWhiting():**

O método .copyWhiting() é responsável por copiar parte de uma array para outra localização dentro dele, sem alterar o seu tamanho original.

A sintaxe padrão desse método é **array.copyWhiting(índice alvo, início, fim)**.

Índice alvo representando o índice do elemento que será substituído no array. Início se refere ao começo da leitura e o fim diz respeito ao final da leitura. Leitura por sua vez quer dizer quais elementos serão copiados. Por exemplo, se meus parâmetros forem (0, 2, 6), significa que o elemento no

**6**

índice 0 será substituído pelos elementos nos índices 2,3,4,5, que dizem respeito ao início da leitura “2” e ao final da leitura “6”, que não inclui o 6.

Quando o último parâmetro (“fim”) não é passado, por padrão a aplicação copia todos os elementos subsequentes aos que foram informados no parâmetro “início”.

Texto

Descrição gerada automaticamente

Podemos ver acima que o “array1” está sendo modificado pelo .copyWhiting da seguinte forma:

No primeiro exemplo, em que está sendo aplicado o método .copyWhiting(0, 3, 4), o elemento de índice “0” está sendo substituído pelo elemento de índice “3”.

Transformando o array de: [‘a’, ‘b’, ‘c’, ‘d’, ‘e’] para [‘d’, ‘b’, ‘c’, ‘d’, ‘e’]

Já no exemplo 2, onde o terceiro parâmetro não é informado, podemos perceber que o elemento 1 e 2 estão sendo substituídos pelos elementos do índice 3 até o último índice (4).

Transformando o array de: [‘d’, ‘b’, ‘c’, ‘d’, ‘e’] para [‘d’, ‘d’, ‘e’, ‘d’, ‘e’]

* 1. **– Array.flat:**

Por fim temos o método .flat que é responsável por aglutinar arrays multidimensionais baseado na “profundidade do aglutinamento” que foi passada como parâmetro. Ou seja, ele trata Arrays que possuam mais arrays dentro de si mesmo e ao tratar esse aninhamento de vetores, além de simplificar e facilitar a compreensão, ele pode também eliminar possíveis espaços vazios dentro desses vetores.

**Sintaxe padrão: Array.flat(profundidade)**.

**7**

No exemplo abaixo vemos o seguinte: constante arr1 recebendo os valores: [0, 1, 2, [3, 4]]. Valores esses que possuem duas dimensões de profundidade. Porém ao utilizarmos arr1.flat() removemos essa “profundidade” e fazemos com que o vetor se transforme em: [0, 1, 2, 3, 4].

**Texto

Descrição gerada automaticamente**

Outra constante que temos nesse exemplo é o arr2 = [0, 1, [2, [3, [4, 5]]]] que possui ainda mais níveis de profundidade. Aumentando o valor do parâmetro é possível perceber que vamos removendo essas “camadas” e se quisermos remover todas podemos utilizar o (Infinity), que as eliminará e deixará o vetor da seguinte forma: [0, 1, 2, 3, 4, 5].

**8**

**CONCLUSÃO**

Este estudo nos forneceu uma introdução aos arrays e a importância de manipulá-los em JavaScript utilizando três métodos: .concat(), .copyWithin() e .flat(). Permitindo-nos aprender como concatenar arrays sem alterar os dados originais, copiar partes de um array para diferentes locais e simplificar arrays multidimensionais. Cada método oferece uma maneira eficaz de trabalhar com arrays, otimizando o código e facilitando o tratamento de dados. A compreensão e o uso adequados desses recursos são essenciais para a manipulação eficaz e flexível de dados.

**9**

**FONTES BIBLIOGRÁFICAS**

[**https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/Arrays**](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays)

[**https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/methods**](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/methods)

[**https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/concat**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/concat)

[**https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/copyWithin**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/copyWithin)

[**https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/flat**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/flat)

**MATERIAL QUE SERÁ UTILIZADO COMO APOIO PARA O SEMINÁRIO:**

**Página:** [**https://luizhssb.github.io/trabalhoMetodosArray/**](https://luizhssb.github.io/trabalhoMetodosArray/)

**Código:** [**https://github.com/LuizHSSB/trabalhoMetodosArray**](https://github.com/LuizHSSB/trabalhoMetodosArray)