



Programação Web

Professor: Euclides Paim
euclidespaim@gmail.com



Fundamentos de JavaScript

Arrays

Professor: Euclides Paim
euclidespaim@gmail.com



Programação Web

Sumário

- **Fundamentos de JavaScript**
 - Introdução *Arrays*
 - Criando *Arrays*
 - Acessando elementos do *Array*
 - Alterando elementos do *Array*
 - Convertendo elementos do *Array*
 - Adicionando elementos no *Array*
 - Diferença entre *Array* e Objetos
 - Métodos JS para *Arrays*
 - Método *.foreach()*
 - Método *.pop()*
 - Método *.remove()*
 - Método *.splice()*
- **Resumo**





Desenvolvimento Web

Introdução

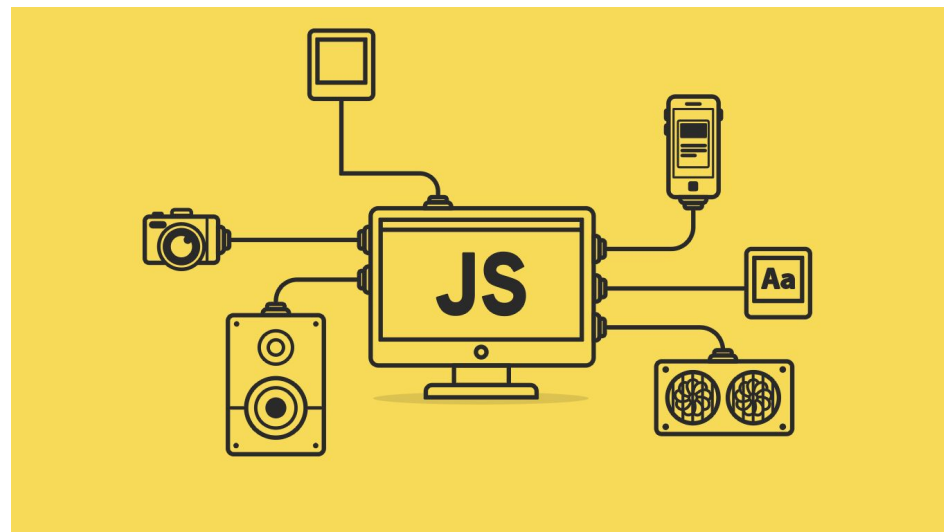
- O que estudamos até agora?
 - HTML
 - Elementos;
 - Atributos;
 - Imagens, tabelas, listas;
 - Classes;
 - Ids;
 - *Iframes*;
 - Elementos semânticos;
 - Formulários, elementos de formulários, tipos de *input*.
 - CSS
 - Sintaxe CSS;
 - Seletores;
 - Formas de inserir CSS, ordem de cascadeamento;
 - *Box model, backgrounds, margins e padding*,
 - Posicionamento, transformações 2D e 3D e transições.





Desenvolvimento Web I

JavaScript Arrays





Desenvolvimento Web

JavaScript Arrays

JavaScript

Arrays

- Um *array* é uma variável especial que pode armazenar mais de um valor.

Por

que

usar

arrays?

- Se tivermos uma lista de itens (nomes de carros, por exemplo), armazenar os carros em variáveis separadas seria improdutivo:

A solução é um array!

- Um array pode armazenar muitos valores sob um único nome, e você pode acessar esses valores usando um número de índice.



Desenvolvimento Web

JavaScript Arrays

Criando **um** **Array**

Usar um *array literal* é a maneira mais fácil de criar um Array em JavaScript.

Sintaxe:

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

Obs.: É uma prática comum declarar arrays com a palavra-chave **const**.





Desenvolvimento Web

JavaScript Arrays

Exemplo:

```
const cars = ["Saab", "Volvo", "BMW"];
```

Espaços e quebras de linha não são importantes. Uma declaração pode se estender por várias linhas:

Exemplo

```
const cars = [  
  "Saab",  
  "Volvo",  
  "BMW"  
];
```




Desenvolvimento Web

JavaScript Arrays

Acessando elementos do array

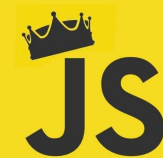
Você acessa um elemento do array referindo-se ao número do índice:

Exemplo

```
const cars = ["Saab", "Volvo", "BMW"];  
let car = cars[0];
```

Observação: Os índices de arrays começam em 0.

[0] é o primeiro elemento. [1] é o segundo elemento.





Desenvolvimento Web

JavaScript Arrays

Alterando um elemento do array

Esta instrução altera o valor do primeiro elemento em `cars`:

```
cars[0] = "Opel";
```

Exemplo

```
const cars = ["Saab", "Volvo", "BMW"];  
cars[0] = "Opel";
```

Observação: Os índices de arrays começam em 0. [0] é o primeiro elemento. [1] é o segundo elemento.



Desenvolvimento Web

JavaScript Arrays

Convertendo um array para uma string

O método `toString()` do JavaScript converte um array em uma string com os valores separados por vírgula.

Exemplo.:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
document.getElementById("demo").innerHTML = fruits.toString();
```

Resultado:

Banana,Orange,Apple,Mango



Desenvolvimento Web

JavaScript Arrays

Adicionando

elementos

ao

array

A maneira mais fácil de adicionar um novo elemento a um array é usando o método `push()`:

```
const fruits = ["Banana", "Orange", "Apple"];  
fruits.push("Lemon"); // Adds a new element (Lemon) to fruits
```





Desenvolvimento Web

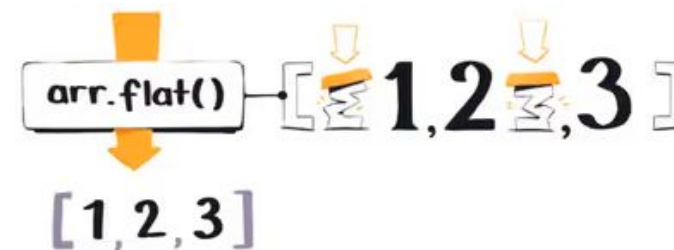
JavaScript Arrays

A diferença entre arrays e objetos

Em JavaScript, arrays usam índices numéricos.

Em JavaScript, objetos usam índices nomeados (chaves).

Dica: Arrays são um tipo especial de objeto, com índices numéricos.





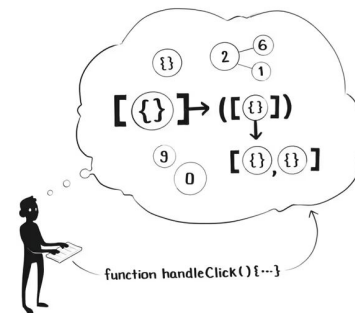
Desenvolvimento Web

JavaScript Arrays

Arrays e Objetos Aninhados

Valores em objetos podem ser arrays, e valores em arrays podem ser objetos.

```
const myObj = {  
  name: "John",  
  age: 30,  
  cars: [  
    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},  
    {name:"BMW", models:["320", "X3", "X5"]},  
    {name:"Fiat", models:["500", "Panda"]}  
  ]  
}
```





Desenvolvimento Web

JavaScript Arrays Methods





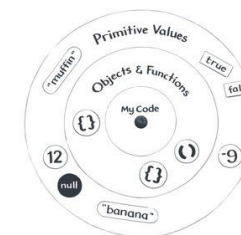
Desenvolvimento Web

JavaScript Arrays Methods

Comprimento de Array em JavaScript

A propriedade `length` retorna o comprimento (tamanho) de um array:"

```
const fruits = ["Banana", "Orange",  
"Apple", "Mango"];  
  
let size = fruits.length;
```





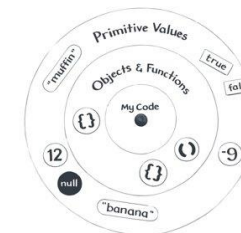
Desenvolvimento Web

JavaScript Arrays Methods

Comprimento de Array em JavaScript

A propriedade `length` também pode ser usada para definir o comprimento de um array.

```
const fruits = ["Banana", "Orange",  
"Apple", "Mango"];  
  
fruits.length = 2;
```





Desenvolvimento Web

JavaScript Arrays Methods

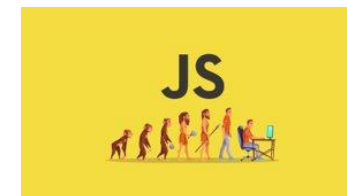
JavaScript

Array

forEach()

O método `forEach()` chama uma função (uma função de retorno de chamada) uma vez para cada elemento do array.

```
const numbers = [45, 4, 9, 16, 25];  
let txt = "";  
numbers.forEach(myFunction);  
  
function myFunction(value, index, array) {  
    txt += value + "<br>";  
}
```





Desenvolvimento Web

JavaScript Arrays Methods

Observe que a função recebe 3 argumentos:

- O valor do item
- O índice do item
- O próprio array





Desenvolvimento Web

JavaScript Arrays Methods

O exemplo anterior usava apenas o parâmetro de valor. O exemplo pode ser reescrito da seguinte forma:

```
const numbers = [45, 4, 9, 16, 25];  
let txt = "";  
numbers.forEach(myFunction);  
  
function myFunction(value) {  
    txt += value + "<br>";  
}
```





Desenvolvimento Web

JavaScript Arrays Methods

JavaScript Array pop()

O método **pop()** remove o **último elemento** de um array.

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.pop();
```





Desenvolvimento Web

JavaScript Arrays Methods

O método **pop()** retorna o valor que foi "removido".

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
let fruit = fruits.pop();
```





Desenvolvimento Web

JavaScript Arrays Methods

JavaScript Array push()

O método **push()** adiciona um **novo elemento no final** de um array.

Exemplo:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Kiwi")
```





Desenvolvimento Web

JavaScript Arrays Methods

O método **push()** retorna o **novo comprimento (tamanho)** do array.

Exemplo:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
let length = fruits.push("Kiwi");
```





Desenvolvimento Web

JavaScript Arrays Methods

JavaScript Array splice()

O método **splice()** pode ser usado para **adicionar novos itens em um array**.

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.splice(2, 0, "Lemon", "Kiwi");
```

- O **primeiro parâmetro (2)** define a **posição onde os novos elementos devem ser adicionados** (inseridos no array).
- O **segundo parâmetro (0)** define **quantos elementos devem ser removidos**.
- O restante dos parâmetros ("**Lemon**", "**Kiwi**") define os **novos elementos que serão adicionados**.





Desenvolvimento Web

JavaScript Arrays Methods

O método **splice()** retorna um **array com os itens que foram removidos**.

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
let removed = fruits.splice(2, 2, "Lemon", "Kiwi");
```





Desenvolvimento Web

JavaScript Arrays Methods

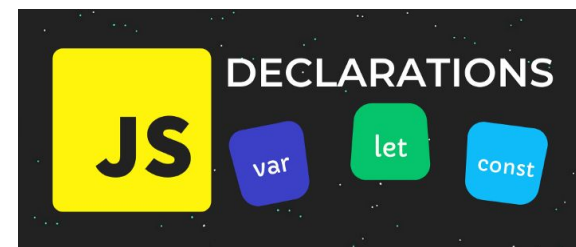




Desenvolvimento Web

JavaScript Arrays Methods

.....



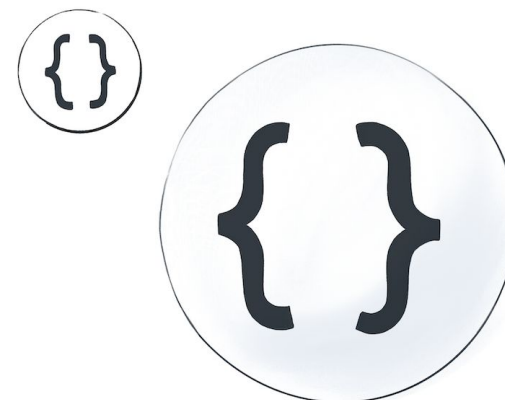


Desenvolvimento Web

JavaScript Arrays Methods

.....

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```





Desenvolvimento Web

JavaScript Arrays Methods

.....

```
typeof "John Doe"    // Retorna "string"  
typeof 3.14           // Retorna "number"
```





Desenvolvimento Web I

Resumo

- **Fundamentos de JavaScript**
 - Introdução *Arrays*
 - Criando *Arrays*
 - Acessando elementos do *Array*
 - Alterando elementos do *Array*
 - Convertendo elementos do *Array*
 - Adicionando elementos no *Array*
 - Diferença entre *Array* e Objetos
 - Métodos JS para *Arrays*
 - Método *.foreach()*
 - Método *.pop()*
 - Método *.remove()*
 - Método *.splice()*





Referências



Referências Básicas

RESIG, John; BIBEAULT, Bear; MARZ, Josip. ***Secrets of the JavaScript Ninja***. 2. ed. Shelter Island: Manning Publications, 2016.

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

GRONER, Loiane. ***Estrutura de dados e algoritmos com JavaScript: escreva aplicações de JS modernas e performáticas utilizando estruturas de dados e algoritmos***. São Paulo: Novatec Editora, 2019.

Referências Complementares

FLANAGAN, David. **o guia definitivo**. . O Really. 2012

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. . Novatec. 2010

IEPSEN, Edécio Fernandes. **Lógica de Programação e Algoritmos com JavaScript**. Novatec. 2018.

Referências na Internet

<https://www.w3schools.com>

<https://developer.mozilla.org/pt-BR/docs/Web>

<https://illustrated.dev/advancedjs>