



SHIFT

 FIAP



000000



FRONT-END SPECIALIST

DESENVOLVIMENTO WEB COM ANGULAR & REACT

LÓGICA DE PROGRAMAÇÃO

JAVASCRIPT



ALEX SANDER RESENDE DE DEUS

PROFESSOR

- Há 25 anos ensinando programação, é um apaixonado por tecnologia.
Atualmente é coordenador de cursos na ETEC Albert Einstein do Centro Paula Souza.
- É professor da FIAP School e FIAP Corporate, lecionando C#, Banco de Dados e Desenvolvimento Mobile.

AGENDA

1

AULA 1

Definição e histórico

Fundamentos Básicos da Programação – Tipos de dados, variáveis e estruturas básicas; Usando o GIT

2

AULA 2

Estruturas condicionais, operadores lógicos e switch case

3

AULA 3

Laços de Repetição: For, while,

4

AULA 4

Objetos nativos JavaScript – Date, String, Math

AGENDA

5

AULA 5

Arrays, Filter, Map

Reduce, Split, forEach, for in, for of

6

AULA 6

Funções: tipos, declarações, escopo, retorno, closures, call-back, IIFE, factory

7

AULA 7

DOM – Document Object Model

8

AULA 8

DOM – Document Object Model

AGENDA

9

AULA 9

Eventos

Objetos: Criação, prototype, getters, setters

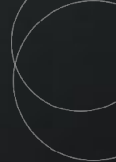

10

AULA 10

Classes JavaScript Assíncrono

AULA 5

ARRAYS JAVASCRIPT



Arrays indexados são arrays que ao serem adicionados um elemento, estes recebem uma posição dentro do array. Por padrão, a posição inicial do Array é 0 (zero). Todos os elementos dentro de um array estão armazenados dentro de um conjunto de chave e valor.

CRIANDO UM ARRAY

```
1  const carros = ["Renegade", "Toro", "Ranger"];
2
3  const carros = [
4      "Renegade",
5      "Toro",
6      "Ranger"
7  ];
8
9  const carros = [];
10 cars[0]= "Renegade";
11 cars[1]= "Toro";
12 cars[2]= "Ranger";
13
14 const carros = new Array("Renegade", "Toro", "Ranger");
```

Acessando um elemento

```
1  const carros = ["Renegade", "Toro", "Ranger"];
2  let x = carros[0];
3  alert (x); //Resultado: Renegade
```

Alterando um elemento

```
1  const carros = ["Renegade", "Toro", "Ranger"];
2  carros[0] = "Onix";
3  let x = carros[0];
4  alert (x); //Resultado: Onix
```

Acessando todo vetor

```
1  const carros = ["Renegade", "Toro", "Ranger"];
2  alert(carros); //Resultado: Renegade,Toro,Ranger
```

ARRAYS SÃO OBJETOS

Arrays são um **tipo especial de objetos**. O operador `typeof` em JavaScript retorna "objeto" para arrays.

Os arrays usam números para acessar seus "elementos".

Pode-se armazenar diferentes tipos de dados em um Array.

Exemplo:

```
1  const pessoas = ["José", "João", 46];  
2  alert (pessoas[0]); //resultado: José
```

ARRAYS SÃO OBJETOS

Os objetos usam nomes para acessar seus "membros":

```
1  const passageiro = {primeiroNome:"José", ultimoNome:"Silva", idade:46};  
2  alert (passageiro.primeiroNome); //Resultado: José
```

Elementos do Array podem ser **Objetos**

```
1  const meuArray=[];  
2  const carros=["Renegade","Toro","Ranger"];  
3  meuArray[0] = Date.now;  
4  meuArray[1] = carros;
```

PROPRIEDADES E MÉTODOS DOS ARRAYS



O grande diferencial dos Arrays são suas propriedades
e métodos integrados.
Vamos ver alguns deles.

COMPRIMENTO (LENGTH)

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  alert(frutas.length);    // Retorna 4
3
4  //Accessando o primeiro elemento:
5  alert (frutas[0]);
6
7  //Acessando o último elemento:
8  alert (frutas.length-1);
```


push()

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  alert(frutas);
3
4  //Adicionando um novo elemento
5  frutas.push("Morango");
6  alert(frutas);
```

pop()

Remove o último elemento de um array.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  alert(frutas);
3  //Retirando o último elemento:
4  frutas.pop();
5  alert(frutas);
```

shift()

Remove o primeiro elemento do vetor e traz todos os outros elementos para um índice antes.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];  
2  alert(frutas);  
3  frutas.shift();  
4  alert(frutas);
```

unshift()

Adiciona um novo elemento no início da fila empurrando os demais para baixo.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  alert(frutas);
3  frutas.unshift("Limão");
4  alert(frutas);
```

delete

Remove um elemento do Array.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];  
2  alert(frutas);  
3  delete frutas[1]; //removendo o item Laranja  
4  alert(frutas);
```



O delete deixa lacunas no Array.

Splice()

Adiciona novos elementos no Array.

- O primeiro parâmetro define a posição onde novos elementos devem ser adicionados (emendados).
- O segundo parâmetro define quantos elementos devem ser removidos .
- O resto dos parâmetros ("Limão", "Kiwi") definem os novos elementos a serem adicionados .

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  alert(frutas);
3  frutas.splice(2, 0, "Limão", "Kiwi");
4  alert(frutas);
```

join()

O `join()` método também une todos os elementos do array em uma string. Ele se comporta exatamente como `toString()`, mas, além disso, você pode especificar o separador.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];  
2  alert(frutas.join(" - ")); //Retorno: Banana - Laranja - Maçã - Manga
```

concat()

O `concat()` cria uma nova matriz mesclando (concatenando) matrizes existentes.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  const verduras=["Couve","Brócolis"]
3
4  const comidas=frutas.concat(verduras);
5  alert(comidas);
```


slice()

O slice() divide uma parte de uma matriz em uma nova matriz, a partir do valor do parâmetro.

```
1  const comidas = ["Banana", "Laranja", "Maçã", "Manga", "Couve", "Brócolis"];
2  const verduras=comidas.slice(4);
3  alert(verduras);
```

sort e reverse()

O sort() ordena um array, enquanto o reverse coloca os elementos em ordem inversa.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2  frutas.sort();//Ordena o Array
3  alert(frutas);
4  frutas.reverse();//Coloca os elementos em ordem reversa
5  alert(frutas);
```

PERCORRENDO UM ARRAY

PERCORRENDO UM ARRAY

As duas melhores maneiras de se percorrer um Array são utilizando um FOR, um FOR EACH ou um MAP.

USANDO UM FOR

HTML

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3      <head>
4          <title>Array</title>
5      </head>
6      <body>
7          <p id="saida"></p>
8          <script type="text/javascript" src="js/script.js"></script>
9      </body>
10 </html>
```

Vamos fazer a saída
nesta tag p.

JAVASCRIPT

Este for percorre do primeiro ao último elemento.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2
3  let text = "<ul>";
4  for (let i = 0; i < frutas.length; i++) {
5    |  text += "<li>" + frutas[i] + "</li>";
6  }
7  text += "</ul>";
8  document.getElementById("saida").innerHTML = text;
```

A variável text está criando uma lista para exibição no arquivo HTML.

USANDO UM FOREACH

JAVASCRIPT

Para cada elemento no vetor, o `foreach` vai chamar a função nos parênteses.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Manga"];
2
3  let text = "<ul>";
4  frutas.forEach(minhaFuncao);
5  text += "</ul>";
6  document.getElementById("saida").innerHTML = text;
7
8  function minhaFuncao(value) {
9      text += "<li>" + value + "</li>";
10 }
```

A função cria o texto a ser exibido no arquivo html.

USANDO UM MAP

JAVASCRIPT

Para cada elemento no vetor, o map vai chamar a função nos parênteses.

```
1  const frutas = ["Banana", "Laranja", "Maçã", "Mangas"];
2
3  let text = "<ul>";
4  frutas.map(minhaFuncao);
5  text += "</ul>";
6  document.getElementById("saida").innerHTML = text;
7
8  function minhaFuncao(value) {
9      text += "<li>" + value + "</li>";
10 }
```

A função cria o texto a ser exibido no arquivo html.



MOMENTO HANDS ON

Elabore um programa em JavaScript que contenha o nome de 5 pessoas em um vetor A, outras 5 pessoas em um vetor B. Crie o vetor C, com o dobro da capacidade, contendo os elementos do vetor A e do Vetor B.

Faça um programa com os nomes de 20 alunos em um vetor A. Divida este vetor ao meio guardando a segunda metade de nomes em um novo vetor.

Insira 10 números inteiros em um vetor A. Crie o vetor B, de mesma dimensão, contendo os elementos de A em ordem inversa.

LEMBRE-SE DE VERSIONAR
SEUS PROJETOS NO GIT.

REFERÊNCIAS

- <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>
- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/String
- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math
- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array
- https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM
- <https://www.w3schools.com/>
- <http://docplayer.com.br/17393758-Javascript-eventos-e-objetos-nativos.html>
- <https://www.adalgisa-souza.appspot.com/javascript/>

OBRIGADO



/profalex.deus@fiap.com.br



/alexsanderresende


FIAP

Copyright © 2021 | Professor Alex Sander Resende

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



SHIFT

 FIAP



000000