
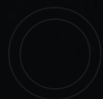




SHIFT

 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.
- Na FIAP é professor da FIAP School e FIAP Corporate, lecionando C#, Banco de Dados e Desenvolvimento Mobile.

✉ profalex.deus@fiap.com.br

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 7

DOM - DOCUMENT OBJECT MODEL

Este é um dos conceitos mais importantes no momento em que precisamos **interagir** com a página web que desenvolvemos.

Assim, para atualizar elementos, criar e até alterar o texto do HTML pelo JavaScript, nós utilizamos o **DOM**.

Então, é possível dizer que manipulamos o **Document Object Model** com o **JavaScript**.

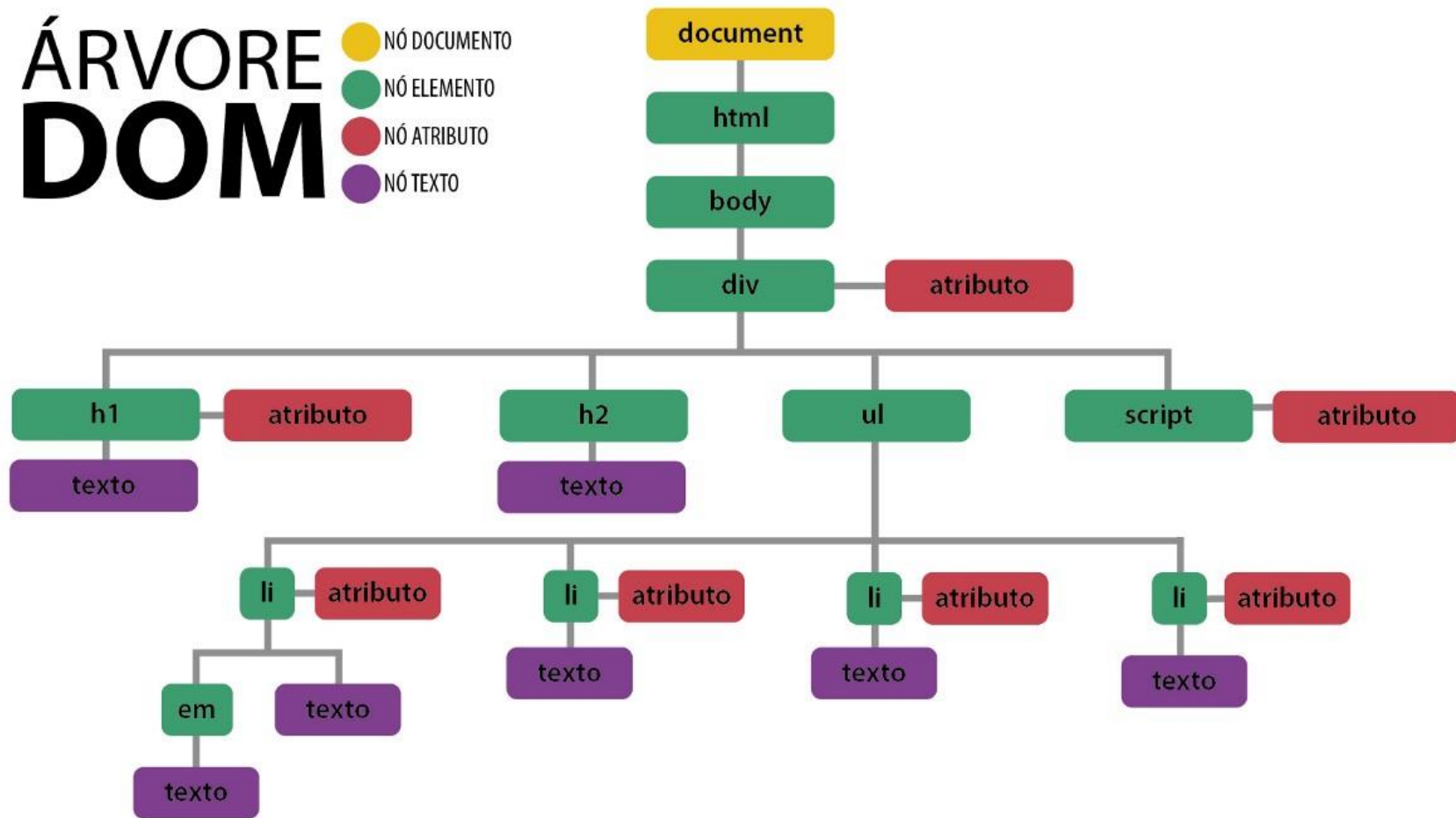
Porém, apesar da forte ligação com a linguagem JS, ele **não é uma parte dela**: podemos pensar no DOM como uma **API** que pode ser acessada pelo JavaScript.

O Modelo de Objeto de Documento (DOM) é uma interface de programação para documentos HTML, XML e SVG . Ele fornece uma representação estruturada do documento como uma árvore, além de definir métodos que permitem acesso à árvore, a fim de que eles possam alterar a estrutura, o estilo e o conteúdo do documento, e fornecer uma representação do documento como um grupo estruturado de nós e objetos, possuindo várias propriedades e métodos.

Os nós também podem contar com manipuladores de eventos que lhe são inerentes, e, uma vez que um evento é acionado, os manipuladores de eventos são executados. Essencialmente, ele conecta páginas web a scripts ou linguagens de programação.

ÁRVORE DOM

- NÓ DOCUMENTO
- NÓ ELEMENTO
- NÓ ATRIBUTO
- NÓ TEXTO



JAVASCRIPT - **MÉTODOS DOM HTML**

Os métodos HTML DOM são **ações** que você pode executar em elementos HTML.

As propriedades DOM HTML são **valores** de elementos HTML que você pode definir ou alterar.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <title>DOM</title>
6   </head>
7   <body>
8     <p id="exemplo"></p>
9     <script type="text/javascript" src="js/script.js"></script>
10  </body>
11 </html>
12
```

O método
getElementById
acessa um elemento
pelo seu id

A propriedade
innerHTML acessa o
conteúdo de um
elemento, podendo
alterá-lo

```
<> index.html ●  Configurações  JS script.js ×
js > JS script.js
1 document.getElementById("exemplo").innerHTML = "Hello World!";
```


JAVASCRIPT - **MÉTODOS DOM HTML** **DOCUMENT OBJECT**

O **DOCUMENT OBJECT** representa sua página da web.

Caso você queira acessar qualquer elemento em uma página HTML, sempre comece acessando o DOCUMENT OBJECT.

Exemplos:

ENCONTRANDO ELEMENTOS HTML

MÉTODO	DESCRIÇÃO
<code>document.getElementById(id)</code>	Encontra um elemento pelo ID
<code>document.getElementsByTagName(name)</code>	Encontra um elemento por sua tag
<code>document.getElementsByClassName(name)</code>	Encontra um elemento pelo nome da classe

ALTERANDO ELEMENTOS HTML

PROPRIEDADE	DESCRIÇÃO
<code>element.innerHTML = new html content</code>	Altera o inner HTML de um elemento
<code>element.attribute = new value</code>	Altera o atributo value de um elemento
<code>element.style.property = new style</code>	Altera o style de um elemento HTML
MÉTODO	DESCRIÇÃO
<code>element.setAttribute(attribute, value)</code>	Altera o valor de um atributo de um elemento HTML

ADICIONANDO E EXCLUINDO ELEMENTOS HTML

MÉTODO	DESCRIÇÃO
<code>document.createElement(element)</code>	Cria um elemento HTML
<code>document.removeChild(element)</code>	Remove um elemento HTML
<code>document.appendChild(element)</code>	Adiciona um elemento HTML
<code>document.replaceChild(new, old)</code>	Altera um elemento HTML
<code>document.write(text)</code>	Escreve no documento

MANIPULADORES DE EVENTOS

MÉTODO	DESCRIÇÃO
<code>document.getElementById(id).onclick = function(){code}</code>	Adiciona um evento de clique a um elemento

ENCONTRANDO ELEMENTO PELO ID

A maneira mais fácil de encontrar um elemento HTML no DOM é utilizando o id do elemento.

Se o elemento for encontrado, o método o retornará como um objeto (em `myElement`).

Se, por sua vez, o elemento não for encontrado, `myElement` conterá `null`.

```
1 <!DOCTYPE html>
2 <html>
3 <meta charset="UTF-8">
4 <head>
5   <title>JavaScript DOM</title>
6 </head>
7
8 <body>
9
10   <h2>JavaScript HTML DOM</h2>
11
12   <p id="paragrafo1">Encontrando elementos HTML pelo ID</p>
13   <p>Demonstrando o método <b>getElementsById</b> .</p>
14
15   <p id="paragrafo2"></p>
16   <script type="text/JavaScript" src="js/script.js"></script>
17 </body>
18 </html>
```

```
2 const element = document.getElementById("paragrafo1");
3
4 document.getElementById("paragrafo2").innerHTML =
5   "O texto do primeiro parágrafo é: " + element.innerHTML;
```


ENCONTRANDO O ELEMENTO PELO NOME DE TAG

Este exemplo encontra todos os elementos <p>

```
1 <!DOCTYPE html>
2 <html>
3 <meta charset="UTF-8">
4 <head>
5   <title>JavaScript DOM</title>
6 </head>
7 <body>
8   <h2>JavaScript HTML DOM</h2>
9   <p>Encontrando elementos pelo tag.</p>
10  <p>Este exemplo demonstra o método <b>getElementsByName</b>.</p>
11  <p id="demo"></p>
12  <script type="text/JavaScript" src="js/script.js"></script>
13 </body>
14 </body>
15
16 </html>
```

```
1 const element = document.getElementsByTagName("p");
2
3 document.getElementById("demo").innerHTML =
4 'O texto no primeiro parágrafo (index 0) é: ' + element[0].innerHTML;
```

ALTERANDO O CONTEÚDO HTML

A maneira mais fácil de modificar o conteúdo de um elemento HTML é utilizando a **propriedade innerHTML**.

```
1 <!DOCTYPE html>
2 <html>
3   <meta charset="UTF-8">
4
5   <head>
6     <title>JavaScript DOM</title>
7   </head>
8
9   <body>
10    <h2>JavaScript alterando HTML</h2>
11
12    <p id="p1">Hello World!</p>
13
14    <p>0 parágrafo acima foi alterado por um script.</p>
15
16    <script type="text/JavaScript" src="js/script.js"></script>
17  </body>
18 </body>
19
20 </html>
```

js > JS script.js

```
1 document.getElementById("p1").innerHTML = "Novo conteúdo";
```

ALTERANDO O ATRIBUTO HTML

```
1 <!DOCTYPE html>
2 <html>
3 <meta charset="UTF-8">
4
5 <head>
6   <title>JavaScript DOM</title>
7 </head>
8
9 <body>
10  <h2>JavaScript HTML DOM</h2>
11  
14  <p>A imagem original era o Homem de Ferro,
15    mas via script alteramos para o homem aranha
16  </p>
17  <script type="text/JavaScript" src="js/script.js"></script>
18 </body>
19 </body>
20
21 </html>
```

Neste exemplo,
estamos alterando
uma imagem pelo
script

js > JS script.js

```
1 document.getElementById("image").src = "https://t2.tudocdn.net/492362?w=5
2
```

CONTEÚDO HTML DINÂMICO

<> index.html > html > head

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <meta charset="UTF-8">
4  <head>
5      <title>JavaScript DOM</title>
6  </head>
7
8  <body>
9      <p id="demo"></p>
10     <script type="text/JavaScript" src="js/script.js"></script>
11 </body>
12 </body>
13
14 </html>
```

js > JS script.js > ...

```
1  let data = new Date();
2  document.getElementById("demo").innerHTML = "Data : " +
3      data.toLocaleString();
```

MOMENTO **HANDS-ON**

01. Crie uma aplicação WEB na qual o visitante realiza um login e deve, então, inserir um texto para autenticar o usuário e outro para autenticar a senha. Se o usuário digitado for igual a “adm” e a senha for igual a “123”, mostre uma mensagem em um parágrafo com o texto “Seja bem-vindo” na cor verde. Caso contrário, mostre a mensagem “Login incorreto” na cor vermelha. Dica: Você pode usar operadores de conjunção &&.

02. Crie uma página para uma loja virtual. A página deve conter a descrição do produto “Baralho de Uno”, que custa R\$ 25,00. O usuário deve então informar em quantas parcelas ele deseja pagar e a página web deve calcular o valor de cada uma delas.

03. Crie uma aplicação web para um boletim escolar em que o professor deve digitar o nome do aluno e inserir três notas de 0 a 10 para ele. A aplicação web deve tirar a média das menções e exibir uma mensagem de acordo com os seguintes critérios:

- a) Média igual ou maior que 7: exibir a mensagem “APROVADO” em verde
- b) Média entre 5 e 6,9: exibir a mensagem “RECUPERAÇÃO” em amarelo
- c) Média menor que 5: exibir a mensagem “REPROVADO” em vermelho

LEMBRE-SE DE VERSIONAR
SEUS PROJETOS NO GIT


BIBLIOGRAFIA


- **LINDEN, M. G. V**, JavaScript: Eventos e Objetos Nativos. [s.d.]. Disponível em: <<http://docplayer.com.br/17393758-Javascript-eventos-e-objetos-nativos.html>>. Acesso em: 08 dez. 2021.
- **MDN WEB DOCS**, Array. 2012. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array>. Acesso em: 08 dez. 2021.
- **MDN WEB DOCS**, JavaScript. 2016. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript>>. Acesso em: 08 dez. 2021.
- **MDN WEB DOCS**, Math. 2014. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math>. Acesso em: 08 dez. 2021.

BIBLIOGRAFIA

- **MDN WEB DOCS**, Modelo de Objeto de Documento (DOM). 2013. Disponível em: <https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM>. Acesso em: 08 dez. 2021.
- **MDN WEB DOCS**, . String. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/String>. Acesso em: 08 dez. 2021.
- **W3SCHOOLS**, W3Schools. 2021. Disponível em: <<https://www.w3schools.com/>>. Acesso em: 08 dez. 2021.

OBRIGADO

 profalex.deus@fiap.com.br

 /alexsanderresende

FIAP

Copyright © 2020 | Alex Sander Resende de Deus

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