

DOM E MANIPULAÇÃO DE ELEMENTOS HTML

(AULA 14)

CURSO BÁSICO DE PROGRAMAÇÃO COM JAVASCRIPT

MAYARA MARQUES

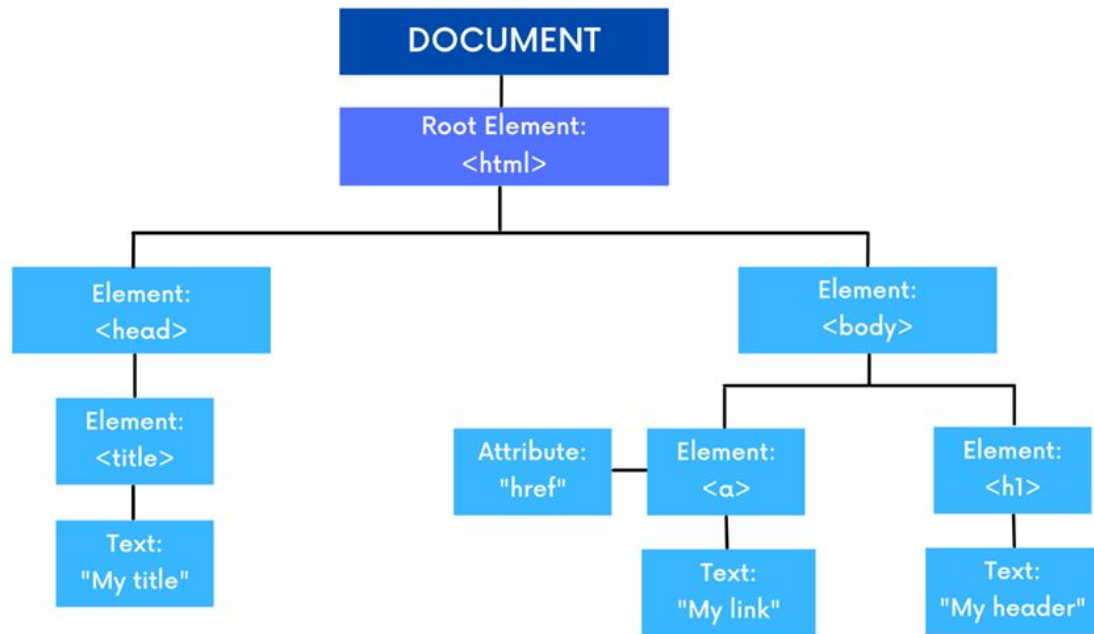
mmrosatab@gmail.com

SUMÁRIO

- DOM
- Manipulação de elementos HTML
- Mão na massa

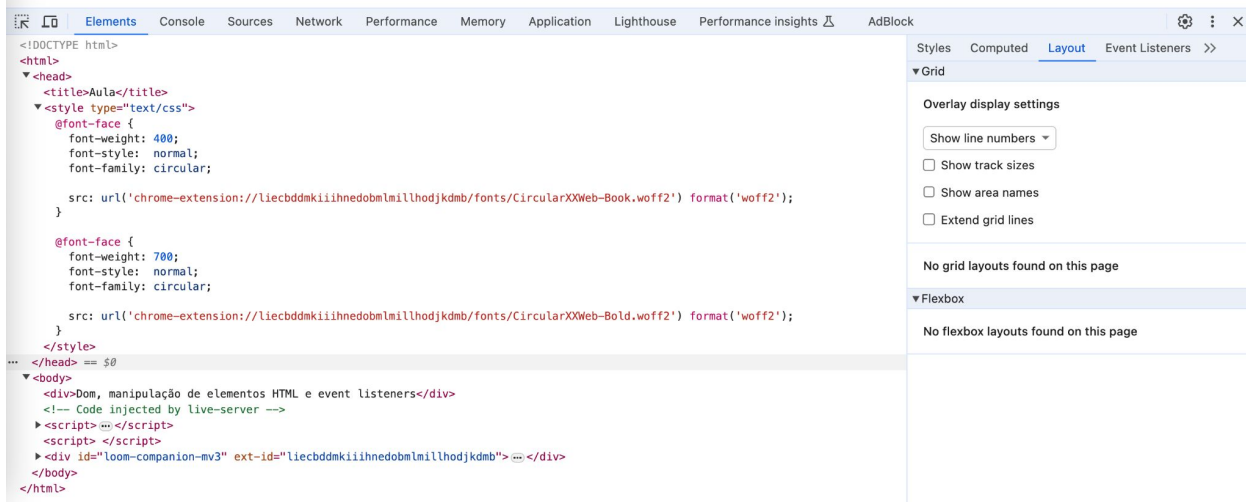
DOM

O DOM, ou Document Object Model (Modelo de Objeto de Documento), é uma *interface de programação de aplicativos (API)* que faz a **representação estruturada e hierárquica** de um **documento HTML** (ou XML) em **forma de uma árvore de objetos**.



Disponível em: [O que é o DOM?](#)

Dom, manipulação de elementos HTML e event listeners



Representação visual do DOM de uma página HTML

DOM

Com o DOM você pode escrever código JavaScript para:

- **Acessar elementos específicos** na página, como parágrafos ou botões.
- **Alterar o conteúdo de um elemento**, como modificar um texto ou imagem.
- **Adicionar ou remover elementos da página**, como criar novos parágrafos ou excluir imagens.
- **Adicionar ou remover atributos a elementos** da página.
- **Responder a eventos** do usuário, como **cliques** de mouse ou **envios** de formulários.
- **Adiciona ou remover eventos e estilos** a um elemento.

MANIPULAÇÃO DE ELEMENTOS HTML

A manipulação de elementos HTML refere-se à **adição, remoção e alteração de elementos** na página usando JavaScript. Isso pode incluir a modificação de estilos, classes, conteúdo textual e estrutura dos elementos.

MANIPULAÇÃO DE ELEMENTOS HTML

DOCUMENT

O document é um **objeto** fundamental no DOM (Document Object Model) em JavaScript, que representa o documento HTML carregado em um navegador. Ele fornece acesso a todos os elementos e recursos presentes na página, permitindo que você os manipule dinamicamente através de scripts.

MANIPULAÇÃO DE ELEMENTOS HTML

- Acesso a elementos:
 - `getElementById(id)`: Retorna o elemento com o ID especificado.
 - `getElementsByClassName(className)`: Retorna uma lista de elementos que possuem a classe especificada.
 - `getElementsByTagName(tagName)`: Retorna uma lista de elementos com o nome da tag especificada.
 - `querySelector(selector)`: Retorna o primeiro elemento que corresponde ao seletor CSS especificado.
 - `querySelectorAll(selector)`: Retorna uma lista de todos os elementos que correspondem ao seletor CSS especificado.

MANIPULAÇÃO DE ELEMENTOS HTML

```
const elementById = document.getElementById('myElementId')
const elementsByClass = document.getElementsByClassName('myClass')
const elementsByTag = document.getElementsByTagName('div')
const elementBySelector = document.querySelector('.myClass')
const elementsBySelectorAll = document.querySelectorAll('p')
```

MANIPULAÇÃO DE ELEMENTOS HTML

- Manipulação de Conteúdo e Estilo:
 - `textContent`: Propriedade que define ou retorna o texto do conteúdo de um elemento.
 - `innerHTML`: Propriedade que define ou retorna o conteúdo HTML de um elemento.
 - `style`: Propriedade que permite acessar e modificar estilos CSS de um elemento.

```
const paragraph = document.getElementById('myParagraph')  
paragraph.textContent = 'Novo texto'
```

MANIPULAÇÃO DE ELEMENTOS HTML

- Manipulação de Conteúdo e Estilo:
 - `textContent`: Propriedade que define ou retorna o texto do conteúdo de um elemento.
 - `innerHTML`: Propriedade que define ou retorna o conteúdo HTML de um elemento.
 - `style`: Propriedade que permite acessar e modificar estilos CSS de um elemento.

```
const paragraph = document.getElementById('myParagraph')  
paragraph.innerHTML = '<strong>Texto em negrito</strong>'
```

MANIPULAÇÃO DE ELEMENTOS HTML

- Manipulação de Conteúdo e Estilo:
 - `textContent`: Propriedade que define ou retorna o texto do conteúdo de um elemento.
 - `innerHTML`: Propriedade que define ou retorna o conteúdo HTML de um elemento.
 - `style`: Propriedade que permite acessar e modificar estilos CSS de um elemento.

```
const paragraph = document.getElementById('myParagraph')  
paragraph.style.color = 'red'
```

MANIPULAÇÃO DE ELEMENTOS HTML

- Manipulação de Conteúdo e Estilo:
 - `textContent`: Propriedade que define ou retorna o texto do conteúdo de um elemento.
 - `innerHTML`: Propriedade que define ou retorna o conteúdo HTML de um elemento.
 - `style`: Propriedade que permite acessar e modificar estilos CSS de um elemento.

```
<div id="example">Este é um exemplo</div>
```

```
// Acessando o elemento pelo ID e alterando seu conteúdo  
const element = document.getElementById('example')  
element.textContent = 'Texto alterado'
```

MANIPULAÇÃO DE ELEMENTOS HTML

- Manipulação de Estrutura:
 - `createElement(tagName)`: Cria um novo elemento com a tag especificada.
 - `appendChild(node)`: Adiciona um novo nó como filho de um elemento.
 - `removeChild(node)`: Remove um nó filho de um elemento.

```
const newElement = document.createElement('div')
newElement.textContent = 'Novo elemento'
document.body.appendChild(newElement)
```

```
const parentElement = document.getElementById('parentElement')
const childElement = document.getElementById('childElement')
parentElement.removeChild(childElement)
```

MANIPULAÇÃO DE ELEMENTOS HTML

```
<div id="container"></div>
```

```
// Criando um novo elemento e adicionando-o ao DOM
const container = document.getElementById('container')
const newElement = document.createElement('p')
newElement.textContent = 'Novo parágrafo'
container.appendChild(newElement)
```




MÃO NA MASSA

MÃO NA MASSA



Baixe os arquivos script.js, index.html e styles.css contidos no link a seguir para realizar os exercícios:

<https://drive.google.com/drive/folders/1Ho8mRFRACM9S1blavFrjE2-U1AcgQCkw?usp=sharing>

MÃO NA MASSA



No JavaScript

1. Altere o conteúdo do elemento com o ID "exemplo" para "Texto alterado!".
2. Altere o texto de todos os elementos com a classe "item" para "Alterado".
3. Altere o conteúdo do primeiro parágrafo (com a classe "paragrafo") para "Primeiro parágrafo alterado".
4. Altere o estilo do elemento com o ID "caixa" para ter cor de fundo azul e largura de 400px.
5. Altere o valor do campo de entrada com o ID "campo" para "Novo valor".
6. Crie um novo item com a classe "item" e adicione-o à lista com o ID "lista".
7. Crie um novo item e insira-o antes do primeiro item da lista "lista".
8. Altere o texto do parágrafo com o ID "textoSubstituir" para "Texto substituído com sucesso!".
9. Altere o estilo do elemento com o ID "caixa" para que sua cor de fundo seja verde e sua altura seja 200px.
10. Adicione uma nova classe "destacado" ao elemento com o ID "caixa".

MÃO NA MASSA



No JavaScript

11. Remova o elemento com o ID "caixa2" do DOM.
12. Substitua o conteúdo do parágrafo com o ID "textoSubstituir" por "Texto completamente novo".
13. Crie e adicione 5 itens à lista "lista".
14. Altere o valor do campo de entrada com o ID "campo" para "Campo alterado".
15. Adicione um estilo inline ao elemento com o ID "caixa" que defina sua altura como 100px e largura como 250px.
16. Crie um novo elemento , adicione alguns itens a ele, e insira o dentro do parágrafo com a classe "paragrafo".
17. Altere a cor do texto de todos os parágrafos (com a classe "paragrafo") para vermelho.
18. Altere o tamanho da fonte do parágrafo com o ID "textoSubstituir" para 20px.
19. Substitua o elemento com o ID "caixa" pelo elemento com o ID "caixa2" usando o método replaceChild.
20. Crie um novo parágrafo e insira-o dentro do elemento com o ID "caixa2".

REFERÊNCIAS

- https://www.w3schools.com/js/js_htmlDOM_methods.asp
- https://www.w3schools.com/js/js_htmlDOM_document.asp
- https://www.w3schools.com/js/js_htmlDOM_elements.asp
- https://www.w3schools.com/js/js_htmlDOM_html.asp