

# Curso de JavaScript 2023



## *Módulo 3 - Aula 07: DOM e Eventos*



Instrutor: *Prof. Me. Mário Carvalho*

E-mail para contato: [mario.carvalho@ufms.br](mailto:mario.carvalho@ufms.br)

Realização: *UFMS e Semadur*





# 1. Sobre o curso

- Instrutor: **Prof. Me. Mário de Araújo Carvalho**
- E-mail para contato: [mario.carvalho@ufms.br](mailto:mario.carvalho@ufms.br)
- Sala Google Meet: <https://meet.google.com/fcq-djzs-dzd>
- Repositório oficial do curso:
- <https://github.com/MarioCarvalhoBr/semadur-curso-javascript-2023>
- **Ava:** <https://www.eadfapec.com.br/course/view.php?id=245&section=4>
- Instituição: **UFMS e Semadur**
- Modalidade: **Online síncrono**
- Duração: 45 horas

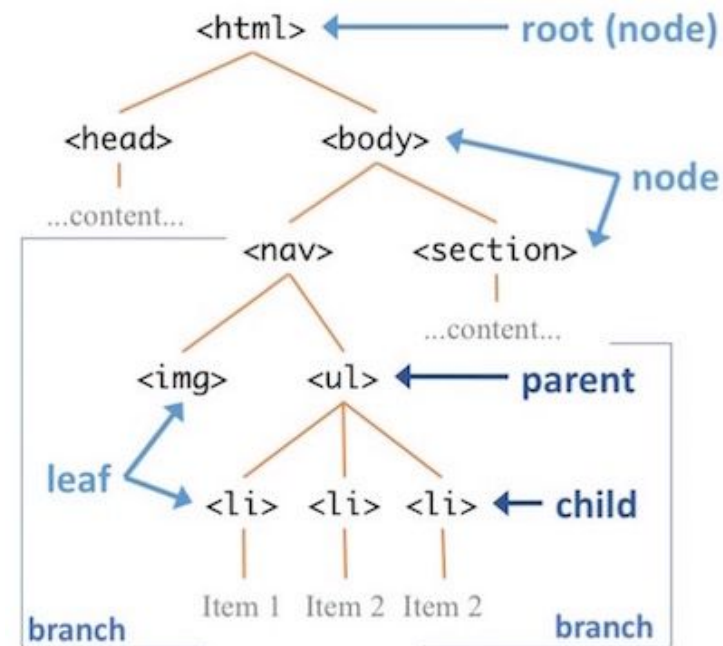
# Introdução

- O que é o DOM??

**Definição:** DOM (Document Object Model) é uma interface de programação para documentos HTML e XML.

**Função:** Representa a página de forma que programas possam alterar a estrutura do documento, estilo e conteúdo.

Apoio: [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)



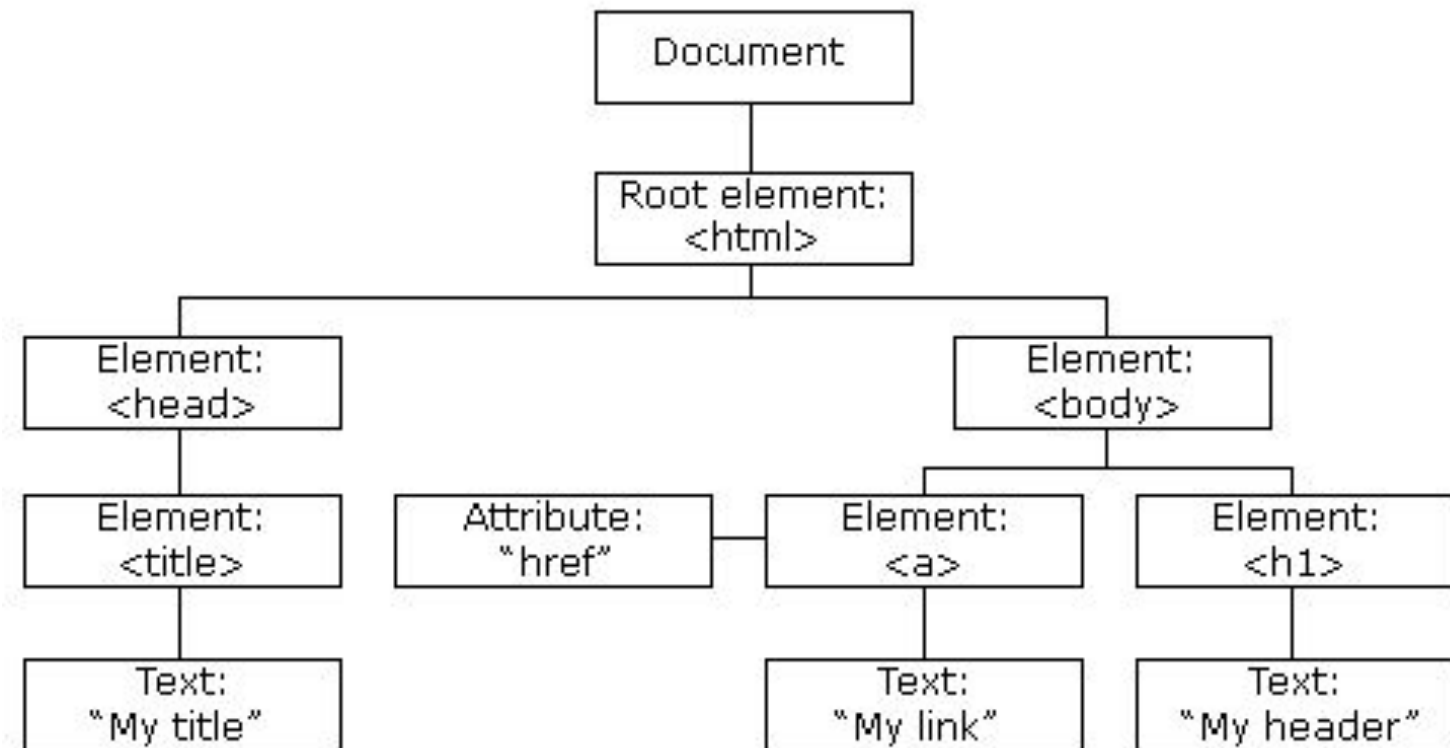


# Introdução

- Componentes do DOM

**Árvore de Nós:** Estrutura que representa todos os elementos da página como nós.

**Elementos e Atributos:** Cada parte do documento, como tags, textos e atributos, é um nó no DOM





# Introdução

- Porque aprender sobre DOM

Com o modelo de objeto, o JavaScript obtém todo o poder necessário para criar HTML dinâmico:

1. JavaScript pode **alterar** todos os elementos HTML da página
2. JavaScript pode **alterar** todos os atributos HTML da página
3. JavaScript pode **alterar** todos os estilos CSS da página
4. JavaScript pode **remover** elementos e atributos HTML existentes
5. JavaScript pode **adicionar** novos elementos e atributos HTML
6. JavaScript pode **reagir** a todos os eventos HTML existentes na página
7. JavaScript pode criar novos **eventos** HTML na página





# DOM

- Manipulação do DOM

## Interagindo com o DOM

**Acesso e Modificação:** Scripts podem usar o DOM para alterar elementos, adicionar ou remover elementos e responder a eventos na página.

**Exemplo prático:**

// 1. Crie no arquivo **index.html** dentro da tag **<body>** uma tag **<h1>Título</h1>** com **id=título**

// Arquivo index.html

**<body>**

**<h1 id="título">Título</h1>**

...

// Arquivo main.js

*// Acessando um elemento*

```
let titulo = document.getElementById("titulo");
```

*// Modificando conteúdo*

```
titulo.textContent = "Novo Título";
```



# Eventos

## Eventos no DOM

**Definição:** Eventos são ações ou ocorrências que acontecem no sistema que você está programando, que o sistema lhe informa para que você possa responder a eles de alguma forma se desejar.

**Exemplos de Eventos:** Cliques de mouse, pressionamentos de teclas, movimentos do mouse, etc.

# Eventos

## Trabalhando com Eventos

**Interação do Usuário:** O DOM permite que scripts respondam a eventos do usuário, como cliques, movimentos do mouse e pressionamento de teclas.

**Exemplo de um evento:**

```
let botao = document.getElementById("meuBotao");  
botao.addEventListener("click", function() {  
    alert("Botão clicado!");  
});
```



# DOM

- Seleccionando Elementos

## Métodos de Seleção

**getElementById:** Selecciona um elemento pelo seu ID.

**querySelector:** Usa seletores CSS para encontrar o primeiro elemento correspondente.

**querySelectorAll:** Retorna todos os elementos que correspondem a um seletor CSS.

Exemplo:

```
let titulo = document.getElementById("titulo");  
let itensMenu =  
document.querySelectorAll(".item-menu");
```

# DOM

- Alterando Elementos

## Modificando Conteúdo e Atributos

**textContent**: Altera o texto de um elemento.

**innerHTML**: Modifica o HTML interno de um elemento.

**setAttribute**: Altera ou adiciona um novo atributo a um elemento.

Exemplo:

```
let descricao = document.getElementById("descricao");  
descricao.textContent = "Novo texto de descrição";
```

```
let link = document.querySelector("a");  
link.setAttribute("href", "https://novo-link.com");
```

# DOM e Eventos

- Tipos de Eventos

## Exemplos Comuns de Eventos

**Eventos de Mouse:** click, mouseover, mouseout.

**Eventos de Teclado:** keypress, keydown, keyup.

**Eventos de Formulário:** submit, change, focus.

## Exemplo:

```
let botao = document.getElementById("botaoEnviar");  
botao.addEventListener("click", function() {  
    alert("Botão clicado!");  
});
```



# DOM e Eventos

- Eventos no DOM

## Adicionando Eventos a Elementos

**addEventListener:** Associa um manipulador de eventos a um elemento.

**Tipos de Eventos:** click, mouseover, mouseout, keydown, load, etc.

Exemplo:

```
let botao = document.getElementById("botaoEnviar");  
botao.addEventListener("click", function() {  
    alert("Botão clicado!");  
});
```

# DOM e Eventos

- Adicionando Eventos a Elementos

## Como Adicionar Eventos

**addEventListener():** Método para vincular um evento a um elemento.

**Sintaxe:** elemento.addEventListener(evento, função)

**Exemplo:**

... Prática em sala

# DOM

- Manipulação de Estilos

## Alterando Estilos via JavaScript

**style:** Modifica o estilo CSS de um elemento.

**classList:** Adiciona, remove ou alterna classes CSS de um elemento.

Exemplo:

```
let mensagem = document.getElementById("mensagem");  
mensagem.style.color = "blue";
```

```
let elemento = document.querySelector(".item");  
elemento.classList.add("classe-ativa");
```

# DOM e Eventos

## Exercícios e Discussão

1. **Alterar Texto:** Use o DOM para mudar o texto de um parágrafo na sua página.
2. **Criar Elementos:** Crie e adicione um novo elemento à página usando o DOM.
3. **Manipular Estilos:** Altere o estilo de um elemento (cor, tamanho) através do DOM.
4. **Clique do Botão:** Adicione um evento de clique a um botão que muda a cor de um texto.
5. **Evento de Teclado:** Vincule um evento de teclado a um campo de entrada e exiba o valor digitado em tempo real.
- 6.
7. **Lâmpada:** Simule o funcionamento de uma lâmpada.
8. **Projeto de loja produtos:** Adicionar, listar produtos, detalhar e pesquisar.

# DOM e Eventos

## Conclusões

1. **Poder do JavaScript no DOM:** A manipulação do DOM é fundamental para criar páginas **web** interativas e dinâmicas.
2. **Prática e Experimentação:** Aprofunde seu conhecimento praticando com diferentes métodos e eventos.
3. **Interface Poderosa:** O DOM é uma ferramenta essencial para a interação dinâmica e a manipulação de páginas **web**.
4. **Fundamental para o Desenvolvimento Web:** Compreender o DOM é crucial para qualquer desenvolvedor **web**.

# Encerramento

- - Resumo dos tópicos abordados e orientações sobre recursos para continuar aprendendo JavaScript:
  - MDN Web Docs:  
<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- - Próximos passos e recursos adicionais para aprendizado autônomo:
  - HTML: <https://www.w3schools.com/html/default.asp>
  - JavaScript: <https://www.w3schools.com/js/default.asp>

Dúvidas?



Obrigado!

Mário Carvalho

[mario.carvalho@ufms.br](mailto:mario.carvalho@ufms.br)