

Manipulação de DOM com JavaScript

Eae galera, hoje vamos falar sobre manipulação de DOM com JavaScript. Nessa aula, vamos explorar como interagir com páginas web utilizando o Document Object Model, permitindo que você manipule elementos HTML, eventos e inputs com JavaScript. Este conhecimento é fundamental para estruturar as interfaces do seu projeto web, especialmente quando trabalhamos com operações CRUD. Vamos descobrir como tornar suas páginas web verdadeiramente interativas!



by ALEX MUNIZ

Conteúdo Programático



O que é DOM?

Compreendendo a estrutura em árvore do documento HTML



Selecionando Elementos

Métodos para encontrar elementos na página



Manipulando Valores

Acessando e alterando conteúdo dos elementos



Eventos JavaScript

Respondendo a interações do usuário



Criando Interfaces

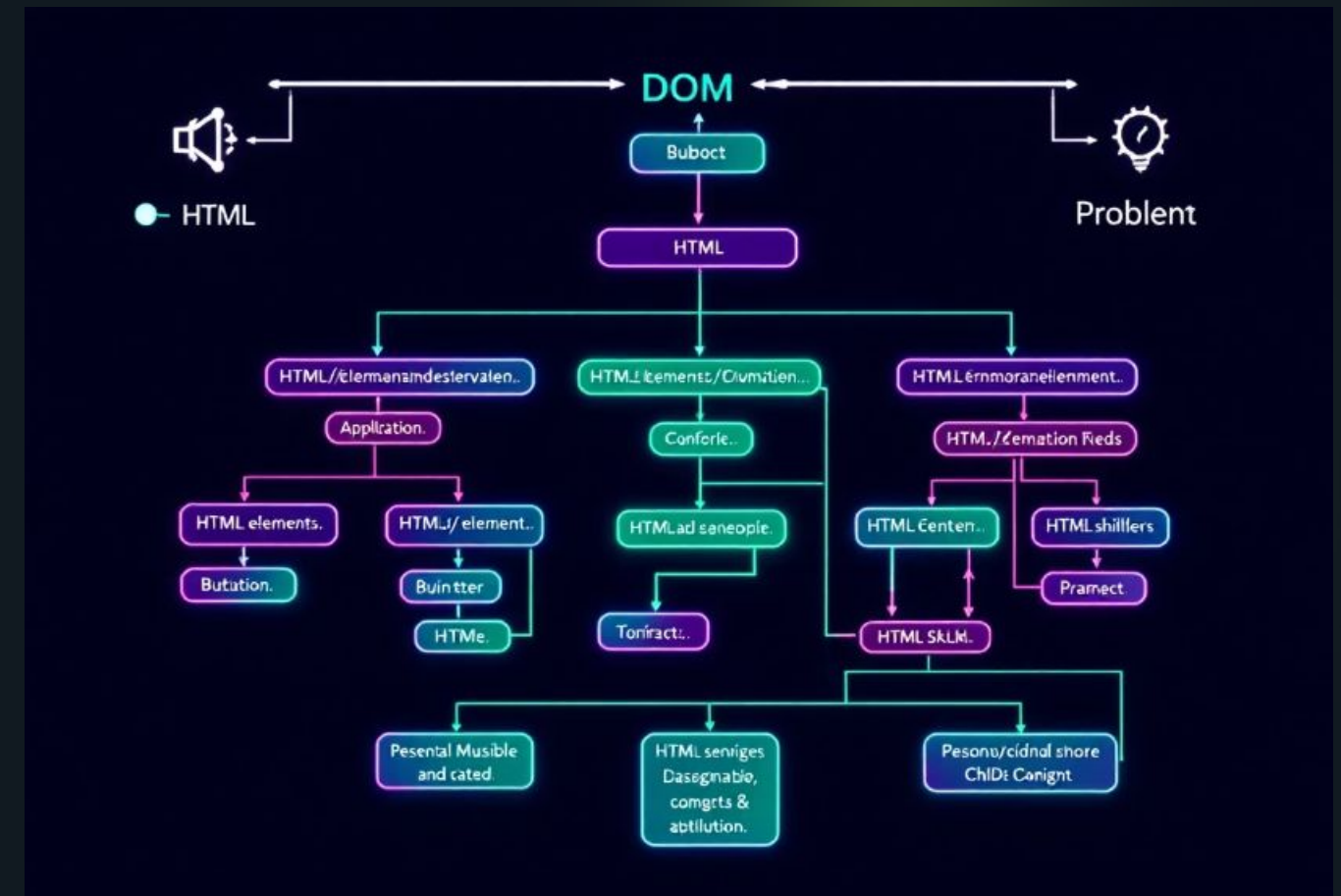
Combinando HTML e JavaScript para interfaces dinâmicas

O que é DOM?

O **DOM (Document Object Model)** é a representação em forma de árvore do conteúdo HTML de uma página. Quando o navegador carrega um documento HTML, ele o transforma em um modelo de objetos que podem ser manipulados via JavaScript.

Cada tag HTML se torna um **nó** (node) no DOM. Existem quatro tipos principais de nós:

- Document: o objeto principal
- Element: qualquer tag HTML (div, input, etc.)
- Text: conteúdo textual dentro de um elemento
- Attribute: atributos HTML (id, class, type)



O DOM permite que você **leia, altere, adicione ou remova elementos HTML usando JavaScript**, tornando sua página verdadeiramente interativa. É a ponte entre o HTML estático e a interatividade dinâmica que os usuários esperam.

Selecionando Elementos no DOM

Método	Descrição	Exemplo
getElementById()	Seleciona pelo id	document.getElementById("meuInput")
getElementsByClassName()	Lista elementos por classe	document.getElementsByClassName("botao")
querySelector()	Seleciona o primeiro que bate com o seletor	document.querySelector("#formulario input")
querySelectorAll()	Lista todos os elementos que batem com o seletor	document.querySelectorAll("input[type='text']")

Boas práticas: Use id para identificar elementos únicos na página. Prefira querySelector para seleções mais complexas e compostas, como #form input[type='text'], que permitem uma navegação mais precisa pela estrutura do DOM.

Acessando e Manipulando Valores

Propriedade .value

Lê ou define o valor de campos de formulário

Exemplo: `input.value = "teste"`

Propriedade .textContent

Lê ou altera texto entre tags

Exemplo: `div.textContent = "Olá"`

Propriedade .innerHTML

Lê ou altera o HTML interno

Exemplo: `div.innerHTML = "Olá"`

Propriedade .classList

Adiciona ou remove classes CSS

Exemplo: `el.classList.add("erro")`

Demonstração: `const nome = document.getElementById("nome").value;`
`document.getElementById("mensagem").textContent = `Bem-vindo, ${nome}`;`

```

refortunate mtl):
    te and {
    onstill time { {
        "actant (mml):
            "abier in- mll garab):
            "ccade is leurer Hanc lava.tentl)");
            "unveicente):{
        };
    };
    "restating furfire "uistant font(/1):
        "oull ln and qond farrent);
    "raint ):
        "astation: {
            "etric.is mtl's doflins stap!";
        };
        "before hatent is lml);
        "ecting nettun";
    }om:
2,
er relcent:
practize tone in contiach() {
before ahten: desicunteon():
    "arten:
        "ecting is htl):
            "con: antdetecting relomatt ('):
        "con:
            "cone content is fohl);
        "fouy:
            "ecting is muls wantl);
            "ayts nol);
    };
-
rianl the/ietorendatiant mull, contents})
re lens {
    is bot's (som> {
;
    "upfren fnrls {
        "cione nettentl)"
    };

```

```
After
"castiad i leht {;
    "erabt;
        "action ie dectlat datl); {
            "cectucc is _ucttat; fanta";
        };
    "fAtl;
        "ccrine dohl);
};
}-.

"Astattinen ils fanls {(
    "ore, scition;
        cern ahctens dat "cheapgerfllenemt
        "nerton";
            "low allthe -fdwll");
            "(on);
                "rbetems"
                    fatter with ceratett));
            "(on);
        });
    );
});
}-.

After is "featts!;
"reactions {
    "implantant fonl);
        "actin);
            "autis are favarsloig(( arll);
        };
        "cere (ons avantsion, "ent-omil-LN
            "(on);
        );
            "foetic helng torfll);
        );
    }
}-.
"action jils ans(, Action close and {
    "efrere while ton @ tectel");
    "e ";
        "one stare fochlo);
            "love an hly);
        );
    );
}-.
}
```


Manipulando Eventos com JavaScript



Evento click

Disparado quando o usuário clica em um elemento

```
button.addEventListener("click", minhaFuncao)
```



Evento input

Ocorre quando um campo de texto é alterado

```
input.addEventListener("input", minhaFuncao)
```



Evento change

Disparado após alteração em select ou checkbox

```
select.addEventListener("change", minhaFuncao)
```

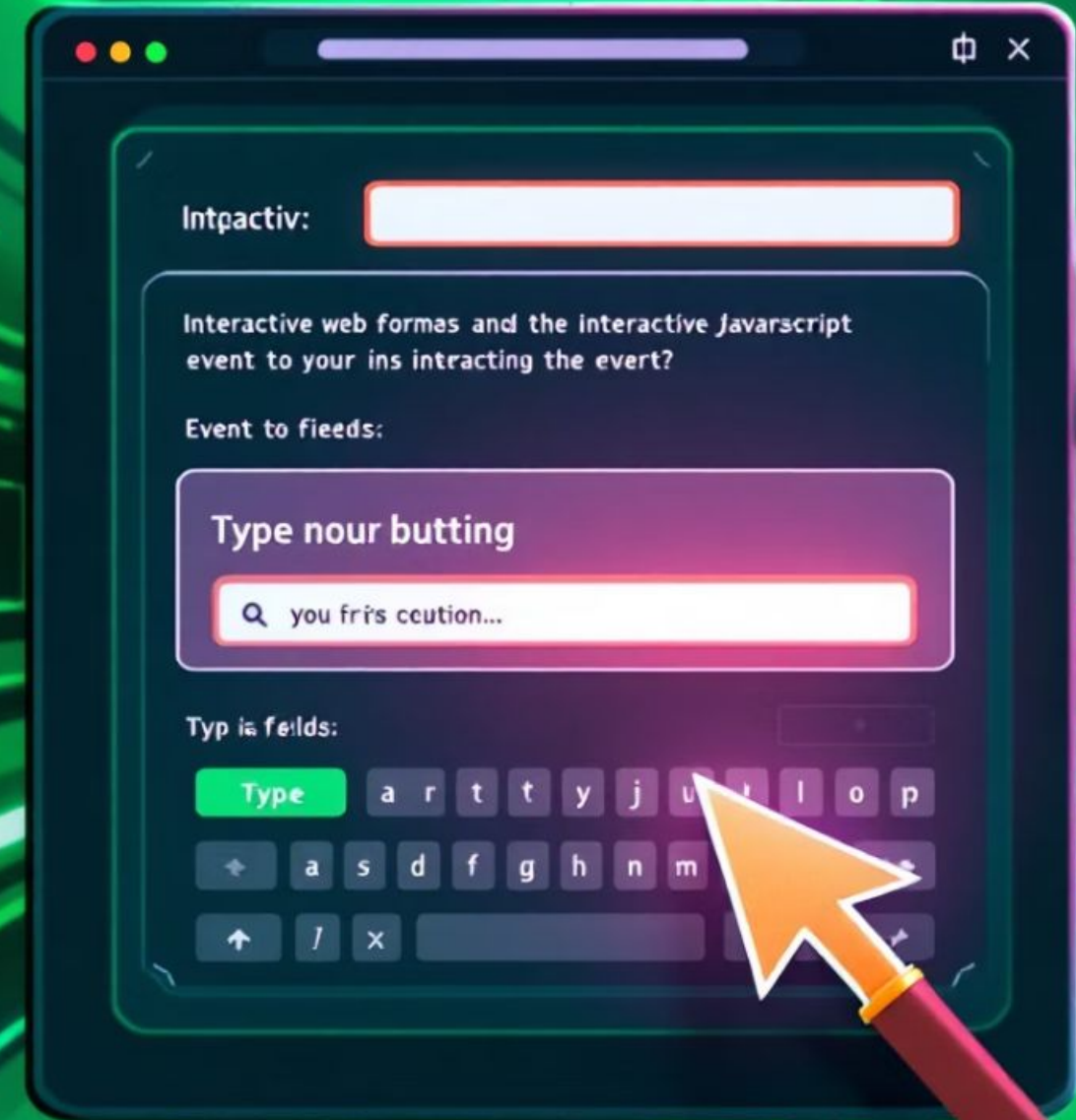


Evento submit

Ocorre quando um formulário é enviado

```
form.addEventListener("submit", minhaFuncao)
```

Exemplo Prático: `document.getElementById("botao").addEventListener("click", function () { const valor = document.getElementById("campo").value; alert("Você digitou: " + valor); });`



Criando uma Interface com HTML + JS

Name

Name

Age

Age

Estrutura HTML

Crie a estrutura base com elementos como inputs, botões e áreas para exibir resultados:

```
<h2>Cadastro</h2>
```

```
<input type="text" id="nome" placeholder="Digite seu nome" />
```

```
<input type="number" id="idade" placeholder="Digite sua idade" />
```

```
<button id="enviar">Cadastrar</button>
```

```
<p id="resultado"></p>
```

Selecionar Elementos

Use os métodos de seleção para acessar os elementos HTML que precisam ser manipulados

Adicionar Eventos

Configure os listeners de eventos para responder às interações do usuário

Manipular o DOM

Atualize o conteúdo da página com base nas entradas do usuário

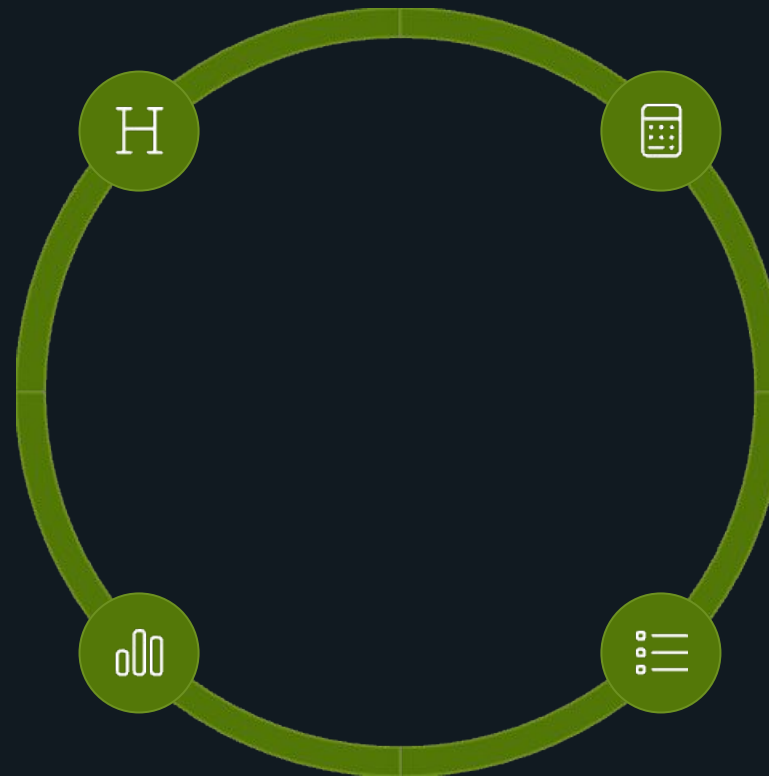
Atividades Práticas

Formulário Interativo

Criar um mini formulário com nome, e-mail e idade que exibe os dados formatados

Visualizador de Dados

Criar uma interface que permita ao usuário inserir dados e visualizá-los em diferentes formatos



Calculadora de Média

Desenvolver um sistema que calcule a média de 3 notas e exiba o resultado com formatação condicional

Lista de Tarefas

Implementar uma lista onde o usuário pode adicionar, marcar como concluído e remover itens

Estas atividades práticas ajudarão a consolidar os conceitos aprendidos, permitindo que você aplique as técnicas de manipulação do DOM em cenários realistas de desenvolvimento web.

Atividade Guiada: Formulário Interativo



Nesta atividade guiada, vamos criar um formulário interativo que coleta informações do usuário e as exibe de forma formatada. Começaremos construindo a estrutura HTML com os campos necessários, depois implementaremos o JavaScript para capturar o evento de envio e manipular os valores inseridos.

Este exercício prático demonstra como os conceitos de seleção de elementos, manipulação de valores e tratamento de eventos se combinam para criar uma experiência interativa para o usuário.

Desafio de Aula: Calculadora de Média

AB

Entrada de Dados

Criar campos para 3 notas

$\frac{-}{+}$
 $\frac{x}{=}$

Cálculo da Média

Implementar a lógica de cálculo



Exibição do Resultado

Mostrar status com formatação condicional

Objetivo: Criar um mini-sistema que calcule a média de 3 notas e exiba "Aprovado" (≥ 7), "Recuperação" (≥ 5 e < 7) ou "Reprovado" (< 5).

Extras: Implemente mudança de cor do texto conforme o resultado (verde para aprovado, laranja para recuperação, vermelho para reprovado) e adicione um botão "Limpar Campos" para resetar o formulário.