

Resumo Completo de Manipulação do DOM com JavaScript

1. Seleção de Elementos

`document.getElementById('id')` - Seleciona um elemento pelo ID

`document.getElementsByClassName('classe')` - Seleciona todos os elementos com a classe (HTMLCollection)

`document.getElementsByTagName('tag')` - Seleciona todos os elementos com a tag (HTMLCollection)

`document.querySelector('seletor')` - Seleciona o primeiro elemento que corresponde ao seletor CSS

`document.querySelectorAll('seletor')` - Seleciona todos os elementos que correspondem ao seletor CSS (NodeList)

`document.forms['nome']` - Seleciona um formulário pelo nome ou índice

`document.images` - Seleciona todas as imagens da página (HTMLCollection)

`document.links` - Seleciona todos os links (elementos `<a>` com href)

`document.anchors` - Seleciona todos os `<a>` com atributo name (obsoleto)

`document.body` - Seleciona o corpo da página

`document.head` - Seleciona o cabeçalho da página

`document.documentElement` - Seleciona o elemento raiz `<html>`

2. Alteração de Conteúdo e Estilo

`element.value` - Acessa ou define o valor de inputs, selects e textareas

`element.innerHTML` - Acessa ou define o conteúdo HTML interno

`element.innerText` - Acessa ou define o texto visível (sem tags HTML)

`element.textContent` - Acessa ou define todo o texto, visível ou não

`element.style.propriedade` - Altera estilos diretamente (ex: `.style.color = 'red'`)

`element.className` - Acessa ou define a(s) classe(s) do elemento

`element.classList.add('classe')` - Adiciona uma classe ao elemento

`element.classList.remove('classe')` - Remove uma classe do elemento

`element.classList.toggle('classe')` - Adiciona ou remove a classe se ela existir

`element.classList.contains('classe')` - Verifica se o elemento possui uma classe

`element.setAttribute('atributo', 'valor')` - Define ou altera um atributo (ex: src, href)

`element.getAttribute('atributo')` - Obtém o valor de um atributo

`element.removeAttribute('atributo')` - Remove um atributo do elemento

`element.disabled = true` - Desativa um campo (formulário)

`element.checked = true` - Marca um checkbox ou radio

`element.src = 'imagem.jpg'` - Altera o caminho da imagem

`element.href = 'https://exemplo.com'` - Altera o link de um `<a>`

`element.focus()` - Dá foco a um campo

`element.blur()` - Remove o foco de um campo

`element.appendChild(elemento)` - Adiciona um novo nó filho

`element.removeChild(elemento)` - Remove um nó filho

`element.replaceChild(novo, antigo)` - Substitui um filho por outro

3. Criação de Elementos

`document.createElement('tag')` - Cria um novo elemento (ex: 'div', 'p', 'img')

`document.createTextNode('texto')` - Cria um nó de texto puro

`document.createComment('comentário')` - Cria um nó de comentário

`document.body.appendChild(elemento)` - Adiciona um elemento ao final do body

`element.appendChild(elemento)` - Adiciona um elemento como filho de outro

`element.insertBefore(novo, existente)` - Insere um elemento antes de outro

`element.insertAdjacentHTML('posição', 'html')` - Insere HTML em posição específica

Resumo Completo de Manipulação do DOM com JavaScript

`element.cloneNode(true)` - Clona um elemento (`true` = clonar com filhos)

Posições para `insertAdjacentHTML`:

`'beforebegin'` - Antes do elemento

`'afterbegin'` - Dentro, antes do primeiro filho

`'beforeend'` - Dentro, depois do último filho

`'afterend'` - Depois do elemento

4. Manipulação de Nós

`element.parentNode` - Acessa o nó pai

`element.childNodes` - Lista todos os nós filhos (inclusive texto/comentário)

`element.children` - Lista apenas os elementos filhos

`element.firstChild` / `element.lastChild` - Primeiro/último filho (qualquer nó)

`element.firstElementChild` / `element.lastElementChild` - Primeiro/último filho elemento

`element.nextSibling` / `element.previousSibling` - Nó irmão (qualquer tipo)

`element.nextElementSibling` / `element.previousElementSibling` - Irmão elemento

5. Eventos

`element.addEventListener('evento', função)` - Adiciona um ouvinte de evento (ex: `click`, `input`)

`element.removeEventListener('evento', função)` - Remove o ouvinte de evento

6. Formulários e Campos

`input.value` - Valor do campo

`checkbox.checked` - Verifica se está marcado

`select.options` - Acessa todas as opções de um `select`

`select.selectedIndex` - Índice da opção selecionada

`form.submit()` - Submete um formulário via JS

`form.reset()` - Reseta os campos de um formulário

7. Remoção de Elementos

`element.remove()` - Remove o próprio elemento

`element.parentNode.removeChild(element)` - Remove com acesso ao pai (mais antigo)

8. Estilo com CSS Inline

`element.style.display = 'none'` - Oculta o elemento

`element.style.visibility = 'hidden'` - Torna invisível, mas mantém espaço

`element.style.backgroundColor = 'blue'` - Altera estilo direto