

Manipulação do DOM com JavaScript - Apostila

Capítulo 1: Introdução à Manipulação do DOM

DOM significa Document Object Model (Modelo de Objeto de Documento). É a representação estruturada de um documento HTML, onde cada parte da página (como elementos, atributos e textos) é um objeto que pode ser manipulado com JavaScript. O DOM é organizado em forma de árvore, permitindo que os desenvolvedores acessem e alterem elementos individuais.

Visualização em ASCII da árvore DOM:

```
html
 /  \
head  body
      /  \
      h1  script
```

Capítulo 2: Selecionando Elementos do DOM

JavaScript oferece várias formas de selecionar elementos do HTML para manipulação:

- `document.querySelector('seletor')`: Retorna o primeiro elemento que corresponde ao seletor CSS.
- `document.querySelectorAll('seletor')`: Retorna todos os elementos que correspondem.
- `document.getElementById('id')`: Retorna o elemento com o ID específico.
- `document.getElementsByClassName('classe')`: Retorna uma coleção de elementos com a classe.
- `document.getElementsByTagName('tag')`: Retorna todos os elementos com a tag especificada.
- `document.forms`, `document.links`, etc.: Métodos específicos para certos elementos.

Capítulo 3: Modificando Conteúdo e Estilo

Depois de selecionar um elemento, você pode modificar seu conteúdo, atributos e estilo. Exemplos:

- `element.textContent`: altera o texto do elemento.
- `element.innerHTML`: altera o HTML interno.

Manipulação do DOM com JavaScript - Apostila

- `element.style.property`: altera o estilo (ex: `element.style.backgroundColor`).
- `element.setAttribute('atributo', valor)`: altera atributos (ex: `src`, `href`).
- `element.classList.add/remove/toggle()`: manipula classes CSS.

```
const titulo = document.querySelector('h1');
titulo.textContent = 'Novo Título';
titulo.style.color = 'blue';
titulo.setAttribute('id', 'principal');
titulo.classList.add('ativo');
```

Capítulo 4: Manipulação de Classes com classList

`classList` é uma propriedade útil que permite adicionar, remover e alternar classes CSS em elementos HTML.

- `element.classList.add('classe')`: adiciona uma classe.
- `element.classList.remove('classe')`: remove uma classe.
- `element.classList.toggle('classe')`: alterna a classe (adiciona se não tiver, remove se tiver).
- `element.classList.contains('classe')`: verifica se o elemento tem a classe.

```
const botao = document.querySelector('button');
botao.classList.toggle('ativo');
```

Capítulo 5: Eventos e addEventListener

Eventos permitem reagir a interações do usuário. O método `addEventListener` associa um evento a uma função.

Exemplo:

- `element.addEventListener('click', function)`: escuta o clique e executa uma função.

Eventos comuns: `click`, `input`, `change`, `mouseover`, `keydown`, `submit`, etc.

Manipulação do DOM com JavaScript - Apostila

```
document.querySelector('button').addEventListener('click', function () {  
    alert('Botão clicado!');  
});
```

Capítulo 6: Criando, Inserindo e Removendo Elementos

- `document.createElement('tag')`: cria um novo elemento.
- `element.appendChild(filho)`: adiciona o elemento ao final de outro.
- `element.prepend(filho)`: insere no início.
- `element.removeChild(filho)`: remove filho específico.
- `element.remove()`: remove o próprio elemento.

```
const novoParagrafo = document.createElement('p');  
novoParagrafo.textContent = 'Texto dinâmico';  
document.body.appendChild(novoParagrafo);
```

Capítulo 7: Lendo e Atualizando Valores de Inputs

Para obter o valor de inputs:

- `input.value`: retorna o valor do campo input ou textarea.
- `checkbox.checked`: retorna se está marcado.

Para atualizar:

- `input.value = 'novo valor'`.

```
const valor = document.querySelector('#entrada').value;  
document.querySelector('#saida').textContent = valor;
```

Capítulo 8: Projeto: Jogo do Número Secreto

Vamos analisar o jogo passo a passo:

Manipulação do DOM com JavaScript - Apostila

```
document.querySelector('.check').addEventListener('click', function () {  
    const guess = Number(document.querySelector('.guess').value);  
    compareNumber(guess);  
});
```

Este trecho adiciona um ouvinte de evento ao botão 'check'. Quando clicado, pega o valor do input '.guess', transforma em número e chama a função que compara.

Capítulo 9: Exercícios

1. Crie um botão que muda o texto de um parágrafo.
2. Altere a cor de fundo da página com um botão.
3. Crie um campo de input que, ao digitar, atualize um texto ao lado.
4. Crie um botão que cria um novo item de lista.
5. Crie um contador com botões de + e -.

Capítulo 10: Soluções

```
// 1  
document.querySelector('#btn').addEventListener('click', function () {  
    document.querySelector('p').textContent = 'Texto alterado!';  
});  
  
// 2  
document.querySelector('#cor').addEventListener('click', function () {  
    document.body.style.backgroundColor = 'lightgreen';  
});  
  
// 3  
document.querySelector('#input').addEventListener('input', function () {  
    document.querySelector('#espelho').textContent = this.value;  
});  
  
// 4  
document.querySelector('#add').addEventListener('click', function () {  
    const li = document.createElement('li');
```

Manipulação do DOM com JavaScript - Apostila

```
li.textContent = 'Novo item';
document.querySelector('ul').appendChild(li);
});

// 5
let count = 0;
document.querySelector('#mais').addEventListener('click', () => {
    count++;
    document.querySelector('#contador').textContent = count;
});
document.querySelector('#menos').addEventListener('click', () => {
    count--;
    document.querySelector('#contador').textContent = count;
});
```