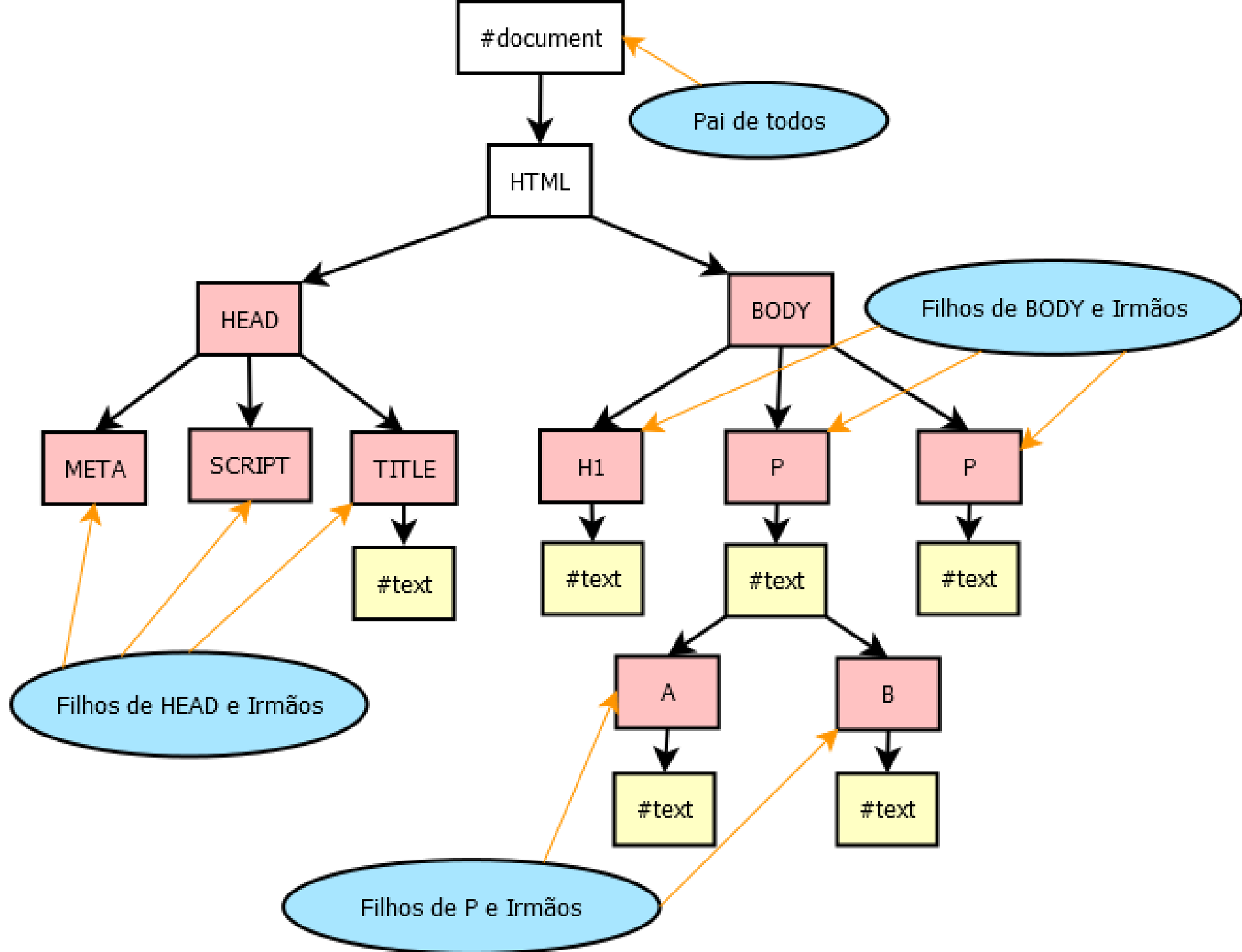


Aula 15

Manipulando Elementos do DOM



Slide 1: O Próximo Passo da nossa Missão 🚀

- **Revisão Rápida:** Na última aula, aprendemos a "capturar" elementos da página usando `getElementById` e `querySelector`.
- **A Missão de Hoje:** Agora que temos os elementos em nossas mãos (em variáveis), vamos aprender a **modificá-los**!
- **O Que Podemos Mudar?**
 - O **conteúdo** (textos e até mesmo outras tags HTML).
 - Os **atributos** (como o link de um `<a>` ou a imagem de um ``).
 - O **estilo** (alterando o CSS através de classes).

Slide 2: Alterando o Conteúdo (Texto e HTML) 🖋️

- `elemento.textContent`: A forma **simples e segura** de ler ou definir **apenas o texto** de um elemento. Qualquer tag HTML que você colocar aqui será tratada como texto puro.
- `elemento.innerHTML`: A ferramenta **poderosa**. Permite ler ou definir todo o conteúdo **HTML** dentro de um elemento. Com ele, você pode adicionar novas tags como ``, ``, etc.
- **Diferença Crucial:** `textContent` para texto, `innerHTML` para HTML. Use com cuidado!

Slide 3: Manipulando Atributos (`src` , `href` , `alt` ...)

- **O Que São Atributos?** São as informações extras nas tags HTML, como `src` em uma imagem ou `href` em um link.
- **Lendo um Atributo:**
 - Acesso direto: `elemento.id` , `elemento.src`
- **Definindo ou Alterando um Atributo:**
 - `elemento.setAttribute('atributo', 'novo-valor')`
- **Exemplo Prático:** Trocar a imagem de um perfil ou o destino de um link de "Saiba Mais".

Slide 4: A Melhor Forma de Estilizar: Classes! ✨

- **Lembrete:** Podemos usar `elemento.style`, mas isso mistura CSS com JS.
- **Uma Forma Mais Organizada:** Manipular as **classes** do CSS no elemento.
- **Conheça o** `elemento.classList`: Uma "caixa de ferramentas" para gerenciar classes:
 - `.add('nova-classe')`: Adiciona uma classe.
 - `.remove('classe-antiga')`: Remove uma classe.
 - `.toggle('classe-liga-desliga')`: Adiciona se não tiver, remove se já tiver. Perfeito para botões de "modo noturno", por exemplo.
- **Vantagem:** Mantém seu estilo no arquivo CSS e seu comportamento no arquivo JS.

Prática

✓ CODIGO    

<> index.html

JS manipulacao.js

style.css

<> index.html X

JS manipulacao.js

style.css

<> index.html > html > body > button#botao

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Aula 13: Manipulando o DOM</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <h1 id="titulo">Título Original</h1>
11
12     <div id="container">
13         Este é um texto simples dentro da div.
14     </div>
15
16     <hr>
17
18     <a id="link" href="https://google.com">Link para o Google</a>
19     
20
21     <hr>
22
23     <p id="paragrafo">Passe o mouse aqui... ou clique no botão!</p>
24     <button id="botao">Ativar/Desativar Destaque</button>
25
26
27     <script src="manipulacao.js"></script>
28     </body>
29 </html>
```

```
# style.css > 🔗 .tema-escuro
1  body {
2      font-family: sans-serif;
3      padding: 20px;
4  }
5
6  /* Classe que vamos adicionar/remover com JavaScript */
7  .destaque {
8      background-color: ■yellow;
9      border: 2px solid □#333;
10     padding: 10px;
11     font-weight: bold;
12 }
13
14 /* Classe para o tema escuro da atividade */
15 .tema-escuro {
16     background-color: □#222;
17     color: ■#eee;
18 }
```

<> index.html

JS manipulacao.js X

style.css

JS manipulacao.js > [🔍] botao

```
1  // AULA 13: Manipulando o DOM - Demonstração dos Conceitos
2  console.log("--- JS: aula13-dom-manipulacao.js (Demonstração) ---");
3
4  // --- EXEMPLO 1: textContent vs. innerHTML ---
5  console.log("\n--- EXEMPLO 1: textContent vs. innerHTML ---");
6  const titulo = document.getElementById('titulo');
7  const container = document.getElementById('container');
8
9  // Lendo o conteúdo
10 console.log("Texto do título (textContent):", titulo.textContent);
11
12 // Alterando o conteúdo com textContent (seguro)
13 titulo.textContent = "Novo Título com textContent!";
14
15 // Alterando o conteúdo com innerHTML (permite injetar HTML)
16 container.innerHTML = "<strong>Texto em negrito</strong> e uma lista: <ul><li>Item 1</li></ul>";
17
18
19 // --- EXEMPLO 2: Manipulando Atributos ---
20 console.log("\n--- EXEMPLO 2: Manipulando Atributos ---");
21 const link = document.getElementById('link');
22 const imagem = document.getElementById('imagem');
```

```
23
24 // Lendo atributos
25 console.log("Destino do link (href):", link.href);
26 console.log("Fonte da imagem (src):", imagem.src);
27
28 // Alterando atributos com .setAttribute()
29 link.setAttribute('href', 'https://www.bing.com');
30 link.textContent = "Novo link para o Bing";
31 imagem.setAttribute('src', 'https://tse4.mm.bing.net/th/id/OIP.4JNcv6toSkGqAMVuOMtKHQHaHa?rs=1&pid=ImgDetMain&o=7&rm=3');
32 imagem.setAttribute('alt', 'Nova imagem de exemplo azul');
33
34
35 // --- EXEMPLO 3: Manipulando Classes com classList ---
36 console.log("\n--- EXEMPLO 3: Manipulando Classes com classList ---");
37 const paragrafo = document.getElementById('paragrafo');
38
39 // Adicionando uma classe
40 console.log("Adicionando a classe 'destaque'...");
41 paragrafo.classList.add('destaque');
42 console.log("Classes do parágrafo:", paragrafo.className);
43
44 // Removendo uma classe (vamos esperar 3 segundos para ver o efeito)
45 setTimeout(() => {
46     console.log("Removendo a classe 'destaque' após 3 segundos...");
47     paragrafo.classList.remove('destaque');
48     console.log("Classes do parágrafo:", paragrafo.className);
49 }, 3000);
50
51 // Usando .toggle() com um botão (uma prévia da aula de eventos!)
52 const botao = document.getElementById('botao');
53 // Este código adiciona um "ouvinte de evento". Quando o botão for clicado, a função será executada.
54 botao.addEventListener('click', () => {
55     console.log("Botão clicado! Usando toggle na classe 'destaque'.");
56     paragrafo.classList.toggle('destaque');
57 });
```

Atividade

- Crie uma nova classe que altera as configurações do texto e da imagem e vincule ao botão “Ativar/Desativar Destaque”