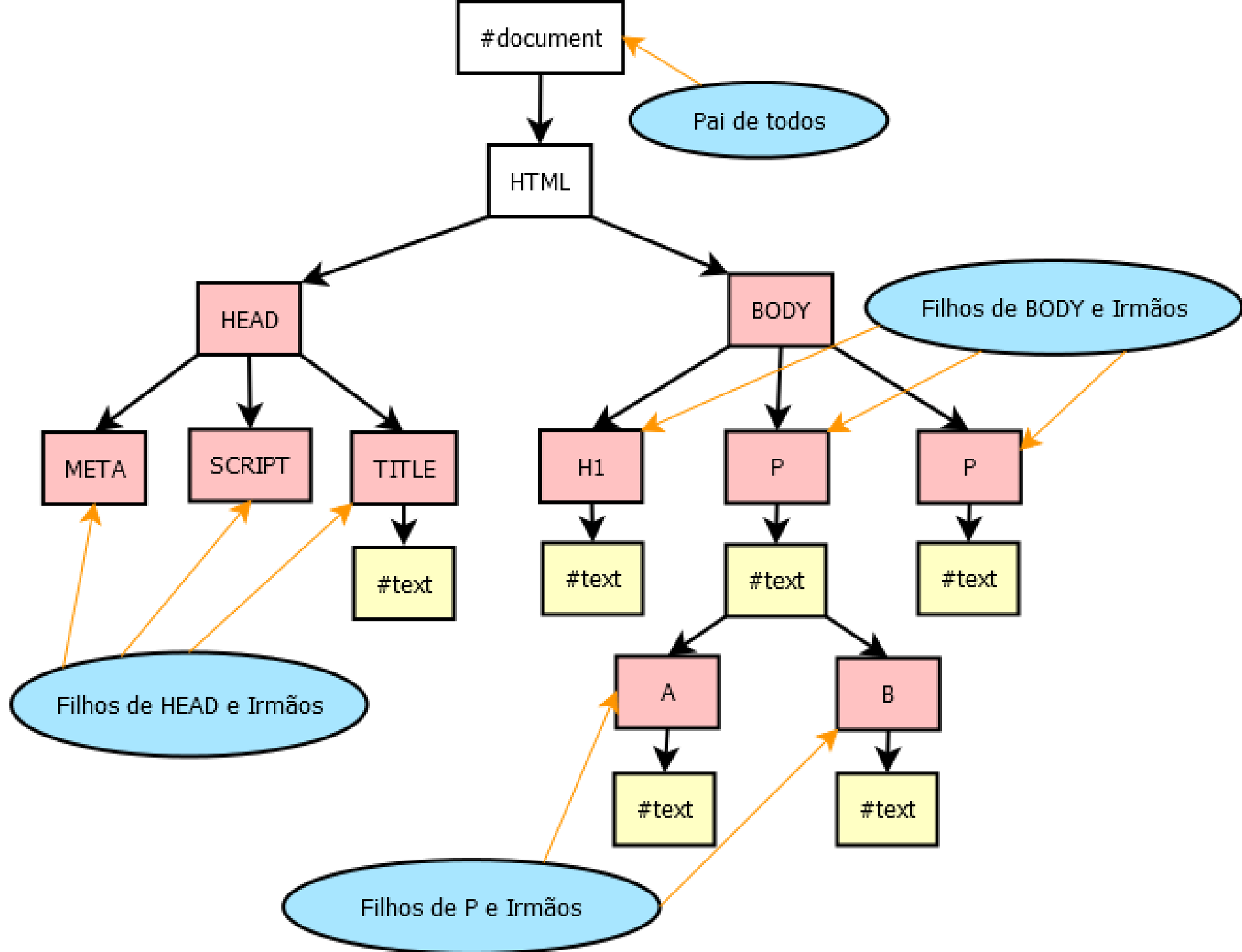


# Aula 18

## Eventos no DOM

**Parte 2 - O Objeto event e Ações.**



## Slide 1: Aprofundando nos Eventos 🕵️

- **Revisão Rápida:** Já sabemos "ouvir" eventos como `click` e `mouseover` com `addEventListener`.
- **Pergunta-chave:** Como podemos saber *mais detalhes* sobre o que aconteceu? Qual tecla foi pressionada? Onde exatamente o usuário clicou?
- **A Missão de Hoje:** Investigar o **objeto** `event`, o "detetive" que nos conta tudo sobre a interação do usuário.

## Slide 2: O Que é o Objeto `event` ?

- **O Parâmetro Secreto:** A função que passamos para o `addEventListener` recebe, automaticamente, um argumento: o objeto `event`.
- **O Que Ele Contém?** Uma coleção de informações sobre o evento que acabou de ocorrer.
- **Propriedades Essenciais:**
  - `event.target` : Nos diz **qual elemento HTML exato** originou o evento.
  - `event.key` : Em eventos de teclado, revela **qual tecla** foi pressionada (ex: 'Enter', 'a', 'Escape').
  - `event.type` : Mostra o tipo de evento que foi disparado (ex: 'click', 'keydown').

### Slide 3: Controlando o Comportamento Padrão 🚦

- **Ações Automáticas:** Alguns elementos HTML têm um comportamento padrão.
  - Clicar em um link `<a>` navega para outra página.
  - Enviar um formulário `<form>` recarrega a página.
  - Pressionar a barra de espaço em uma página rola o conteúdo.
- **Assumindo o Controle:** Podemos impedir essas ações padrão com o JavaScript.
- **O Método Mágico:** `event.preventDefault()`.

#### Slide 4: Por Que Usar `event.preventDefault()` ?

- **Validação de Formulários:** O uso mais comum! Podemos verificar os dados de um formulário *antes* de enviá-lo. Se houver um erro, usamos `preventDefault()` para impedir o recarregamento da página e mostrar uma mensagem de erro.
- **Links com Funções JS:** Criar links que, em vez de navegar, executam uma função JavaScript, como abrir uma janela pop-up (modal) ou mostrar/esconder um menu.
- **Controle Total:** Garante que o nosso código JavaScript tenha a palavra final sobre o que acontece na página.

Prática

```
8 <body>
9
10 <div id="container">
11     <p>Clique em um dos botões:</p>
12     <button>Botão A</button>
13     <button>Botão B</button>
14     <button>Botão C</button>
15 </div>
16 <p>Último botão clicado: <span id="botao-alvo">Nenhum</span></p>
17
18 <hr>
19
20 <label for="campo-texto">Digite algo e veja o console:</label>
21 <input type="text" id="campo-texto" placeholder="Veja a tecla no console...">
22
23 <hr>
24
25 <form id="formulario">
26     <p>Este formulário não será enviado da forma tradicional.</p>
27     <button type="submit">Enviar</button>
28 </form>
29
30 <script src="script.js"></script>
31 </body>
32 </html>
```



```
1 // --- EXEMPLO 1: Identificando o Alvo com 'event.target' ---
2 // Isso é chamado de "Delegação de Eventos". Em vez de adicionar um ouvinte
3 // para cada botão, adicionamos um único ouvinte ao container pai.
4 console.log("\n--- EXEMPLO 1: event.target ---");
5 const container = document.getElementById('container');
6 const span = document.getElementById('botao-alvo');
7
8 container.addEventListener('click', (event) => {
9     // 'event.target' nos diz qual elemento DENTRO do container foi clicado.
10    const elementoClicado = event.target;
11    span.textContent = elementoClicado.textContent;
12
13
14 });
```

```
16 // --- EXEMPLO 2: Capturando Teclas com 'event.key' ---
17 console.log("\n--- EXEMPLO 2: event.key ---");
18 const campo = document.getElementById('campo-texto');
19
20 campo.addEventListener('keydown', (event) => {
21     // 'event.key' contém o valor da tecla que foi pressionada.
22     console.log(`Tecla pressionada: ${event.key} (Código: ${event.code})`);
23
24 });
25
```

```
26 // --- EXEMPLO 3: Impedindo o Comportamento Padrão com 'event.preventDefault()' ---
27 console.log("\n--- EXEMPLO 3: event.preventDefault() ---");
28 const formulario = document.getElementById('formulario');
29
30 // O evento para formulários é o 'submit'.
31 formulario.addEventListener('submit', (event) => {
32     // A primeira coisa a se fazer é impedir o recarregamento da página.
33     event.preventDefault();
34
35     💡 alert("Formulário não foi enviado! O JavaScript assumiu o controle.");
36     // Aqui faríamos a validação dos campos e o envio dos dados com outras técnicas (ex: Fetch API).
37 });
```

Atividade

Criar um formulário de inscrição para uma rede social que usa `event.preventDefault()` para impedir o envio se os campos não forem preenchidos corretamente, demonstrando uma validação real.