



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina
Programação Front-End I

PROF. FERNANDO TAMBERLINI ALVES

1. Linguagem de programação multiparadigma usada para:
 - Atuar junto ao navegador (client-side);
 - Validação de formulários no lado do cliente;
 - Interação e alteração do comportamento da página WEB;
 - Atualmente também pode ser utilizada no lado do servidor (server-side) – NODE.JS
2. É uma linguagem interpretada com tipagem fraca e dinâmica;
 - Dinâmica: você não precisa declarar o tipo das variáveis. O tipo é determinado em tempo de execução;
 - Fraca: o JavaScript converte automaticamente tipos diferentes quando necessário, o que pode gerar comportamentos inesperados.

1. Tipos de Dados (boolean, number, string, undefined, arrays, objects etc)
2. Operadores Lógicos e Aritméticos (+, -, *, /, %, =, ==, ===, >, <, &&, ||)
3. Declaração de Variáveis (var, const e let)
4. Escopo de variáveis (bloco, função e global)
5. Declaração de Objetos, Classes e Funções
6. Estruturas de Controle e Repetição (if, for, while)
7. Classes Strings, Numbers, Math, Date, Arrays

1. Atributo onclick e outros manipuladores inline

```
<button onclick="alert('Você clicou!')">Clique aqui</button>
```

2. Script Interno – Tag `<script>` no head ou antes de fechar body

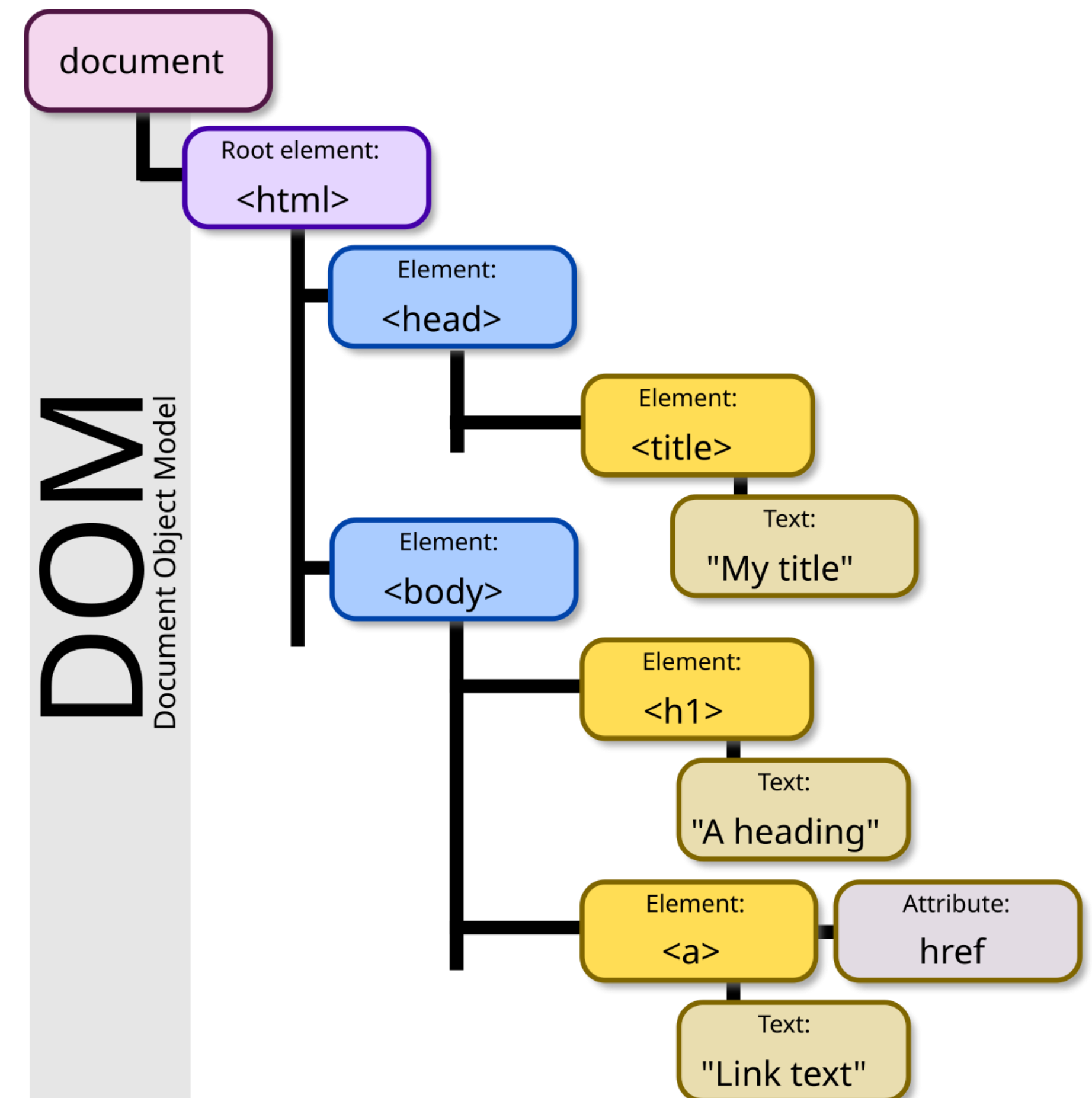
```
<script>  
    alert("Bem-vindo ao site!");  
</script>
```

3. Script Externo – Tag `<script>` no head com atributo src

```
<script src="script.js"></script>
```

JavaScript - DOM e Eventos

1. DOM – Document Object Model
2. Representação em árvore dos elementos HTML
3. Interface que permite ao Javascript interagir com o HTML
4. Cada elemento HTML se torna um objeto que pode ser manipulado
5. Permite adicionar, remover, modificar elementos dinamicamente



Selecionando Elementos

- **getElementById(id)**

Retorna um único elemento com o **id** especificado

```
const titulo = document.getElementById("meuTitulo");
```

- **getElementsByClassName(className)**

Retorna uma coleção (HTMLCollection) de todos os elementos com a **classe** especificada

```
const botoes = document.getElementsByClassName("btn");
```

Selecionando Elementos

- **getElementsByTagName(tagName)**

Retorna todos os elementos com a **tag** especificada.

```
const paragrafos = document.getElementsByTagName("p");
```

- **querySelector(selector)**

Retorna o primeiro elemento que corresponde ao **seletor** CSS fornecido.

```
const paragrafo = document.querySelector("p.exemplo");
```

- **querySelectorAll(selector)**

Retorna todos os elementos que correspondem ao **seletor** CSS, como uma NodeList.

```
const itens = document.querySelectorAll("ul li");
```

- **Object Collections**

Retorna pelo **id** do elemento

```
const x = document.forms["frm1"]  
const y = document.anchors.length;
```

Alterando os Elementos HTML

- **Alterando conteúdo do elemento HTML (.innerHTML)**

```
const item = document.getElementById(id)
item.innerHTML = "Novo Conteúdo HTML"
item.textContent = "Novo Texto"
```

- **Alterando valor do atributo elemento HTML (.attribute)**

```
const item = document.getElementById("myImage");
item.src = "landscape.jpg";
item.setAttribute('class', 'nova-classe');
item.style.color = 'red';
item.style.backgroundColor = 'yellow';
```

- Criando Novo elemento HTML – `appendChild()`

Cria um novo elemento `<p>`

```
const novo = document.createElement("p")
```

Adiciona um nó texto ao parágrafo

```
const no = document.createTextNode("Novo Parágrafo");
```

Acrescenta o no ao elemento novo

```
novo.appendChild(no)
```

Acrescenta o novo a um element já existente

```
const element = document.getElementById("div");  
element.appendChild(novo);
```

- Criando Novo elemento HTML – insertBefore()

```
<div id="div1">
  <p id="p1">Este é um parágrafo.</p>
  <p id="p2">Este é um outro parágrafo.</p>
</div>
<script>
  const para = document.createElement("p");
  const node = document.createTextNode("Novo Parágrafo");
  para.appendChild(node);

  const element = document.getElementById("div1");
  const child = document.getElementById("p1");
  element.insertBefore(para, child);
</script>
```

Excluindo Elementos HTML

- **Excluindo elemento HTML – remove()**

```
<div>  
    <p id="p1">Este é um parágrafo.</p>  
    <p id="p2">Este é um outro parágrafo.</p>  
</div>
```

```
<script>  
    const elmnt = document.getElementById("p1");  
    elmnt.remove();  
</script>
```

Excluindo Elementos HTML

- **Excluindo elemento HTML – removeChild()**

```
<div>  
    <p id="p1">Este é um parágrafo.</p>  
    <p id="p2">Este é um outro parágrafo.</p>  
</div>
```

```
<script>  
    const parent = document.getElementById("div1");  
    const child = document.getElementById("p1");  
    parent.removeChild(child)  
</script>
```

JavaScript - DOM e Eventos

1. Eventos são ações que acontecem na página WEB
2. Exemplos: clicks, teclas pressionadas, movimento do mouse
3. JavaScript pode escutar esse eventos
4. Quando um evento ocorre, uma função Javascript pode ser executada
5. Isso torna as páginas web interativas

■ Eventos de Mouse

<code>onclick</code>	o elemento é clicado
<code>ondblclick</code>	duplo clique no elemento
<code>onmousedown</code>	o botão do mouse é pressionado
<code>onmouseup</code>	o botão do mouse é solto
<code>onmousemove</code>	o mouse se move sobre o elemento
<code>onmouseover</code>	o mouse entra na área do elemento
<code>onmouseout</code>	o mouse sai da área do elemento
<code>oncontextmenu</code>	o botão direito do mouse é clicado

■ Eventos do Teclado

<code>onkeydown</code>	uma tecla é pressionada
<code>onkeyup</code>	uma tecla é solta

- Eventos de formulário

<code>onsubmit</code>	um formulário é enviado
<code>onreset</code>	um formulário é resetado
<code>onchange</code>	o valor de um campo muda (e perde o foco)
<code>oninput</code>	o valor de um campo muda (enquanto digita)
<code>onfocus</code>	o campo recebe foco
<code>onblur</code>	o campo perde o foco

- Eventos de Toque (Mobile)

<code>ontouchstart</code>	o dedo toca a tela
<code>ontouchmove</code>	o dedo se move sobre a tela
<code>ontouchend</code>	o dedo é levantado
<code>ontouchcancel</code>	toque é interrompido (ex: chamada recebida)

- Eventos de Janela/Document

<code>onload</code>	a página termina de carregar
<code>onresize</code>	o tamanho da janela muda
<code>onscroll</code>	a página é rolada
<code>onunload</code>	a página está prestes a ser fechada
<code>onbeforeunload</code>	antes de sair ou recarregar a página
<code>onerror</code>	ocorre um erro ao carregar um recurso

O evento no `addEventListener` não tem o prefixo "on"

Exemplo – Evento Inline

```
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>Evento Inline Simples</title>
7  </head>
8  <body>
9      <h1>Exemplo de Evento Inline</h1>
10     <button onclick="mudarTexto()">Clique aqui!</button>
11     <div id="mensagem">Aguardando clique...</div>
12     <script>
13         function mudarTexto() {
14             document.getElementById('mensagem').innerHTML = 'Botão foi clicado!';
15         }
16     </script>
17 </body>
18 </html>
```

Exemplo – addEventListener

```
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>addEventListener Simples</title>
7  </head>
8  <body>
9      <h1>Exemplo de addEventListener</h1>
10     <button id="meuBotao">Clique aqui!</button>
11     <div id="mensagem">Aguardando clique...</div>
12     <script>
13         // Selecionar o botão
14         const botao = document.getElementById('meuBotao');
15         // Adicionar event listener
16         botao.addEventListener('click', function() {
17             document.getElementById('mensagem').innerHTML = 'Evento capturado com addEventListener!';
18         });
19     </script>
20 </body>
21 </html>
```

Atividade Prática I

Utilize o código exemplo e faça as seguintes alterações:



- Coloque o código Javascript em um arquivo externo;
- Altere o tipo de evento;
- Altere outro atributo do elemento html (cor, fonte, etc)

Dúvidas?



Dúvidas?

 fernando.alves@ifrj.edu.br

 ftamberlini.dev.br