

JavaScript

Prof. Daniel Di Domenico



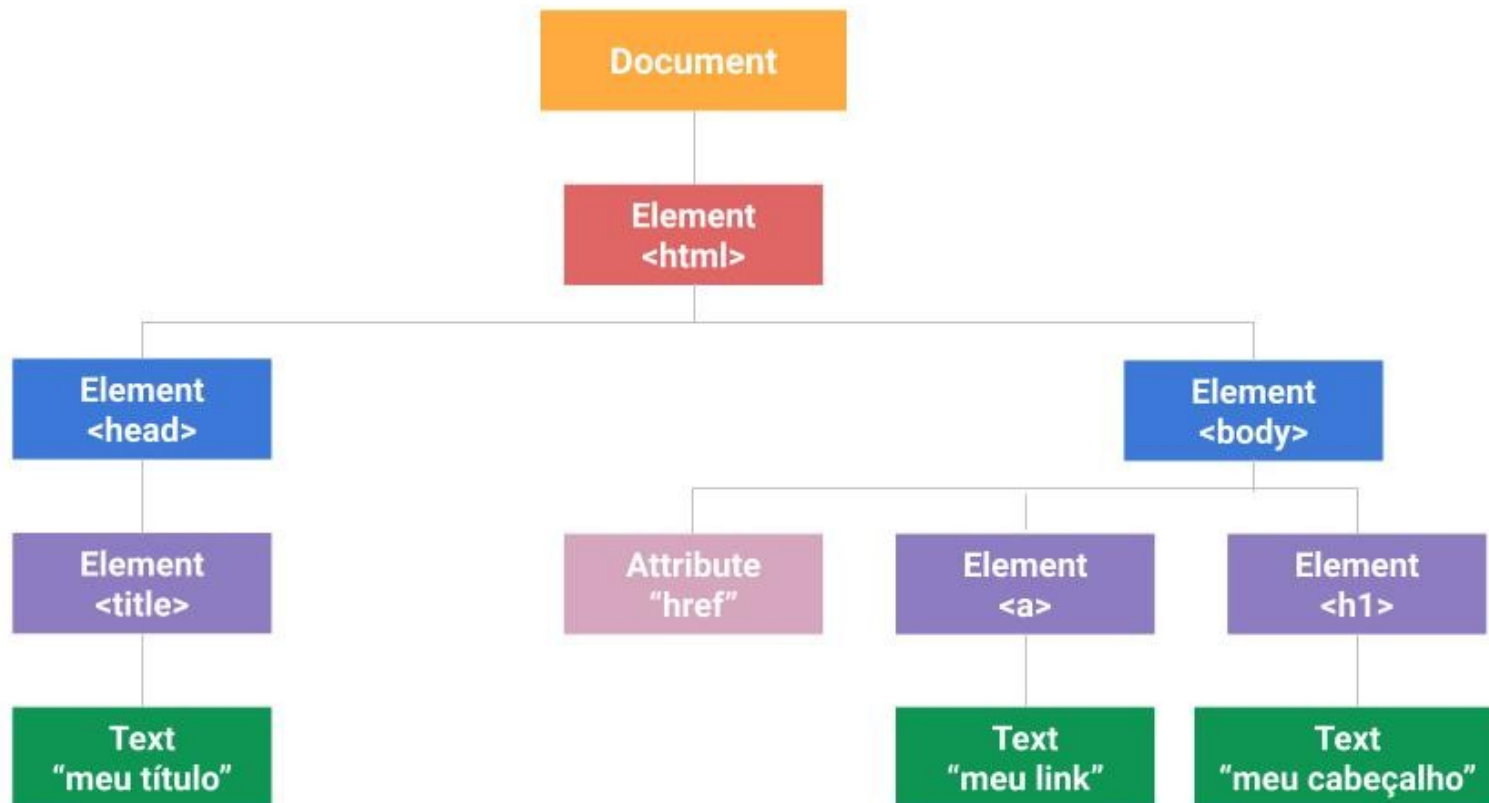
DOM

DOM

- **Document Object Model**
 - Representação, na forma de objetos, de todos os elementos que existem em uma página HTML
 - **Árvore de objetos** que representa a página HTML
 - O JavaScript permite acesso aos objetos do **DOM**, sendo possível:
 - Alterar os elementos da página HTML
 - Alterar os estilos CSS da página HTML
 - Adicionar ou remover elementos HTML e seus atributos
 - Criar eventos ou reagir a eventos já existentes na página HTML

DOM

- Exemplo de árvore **DOM**



DOM

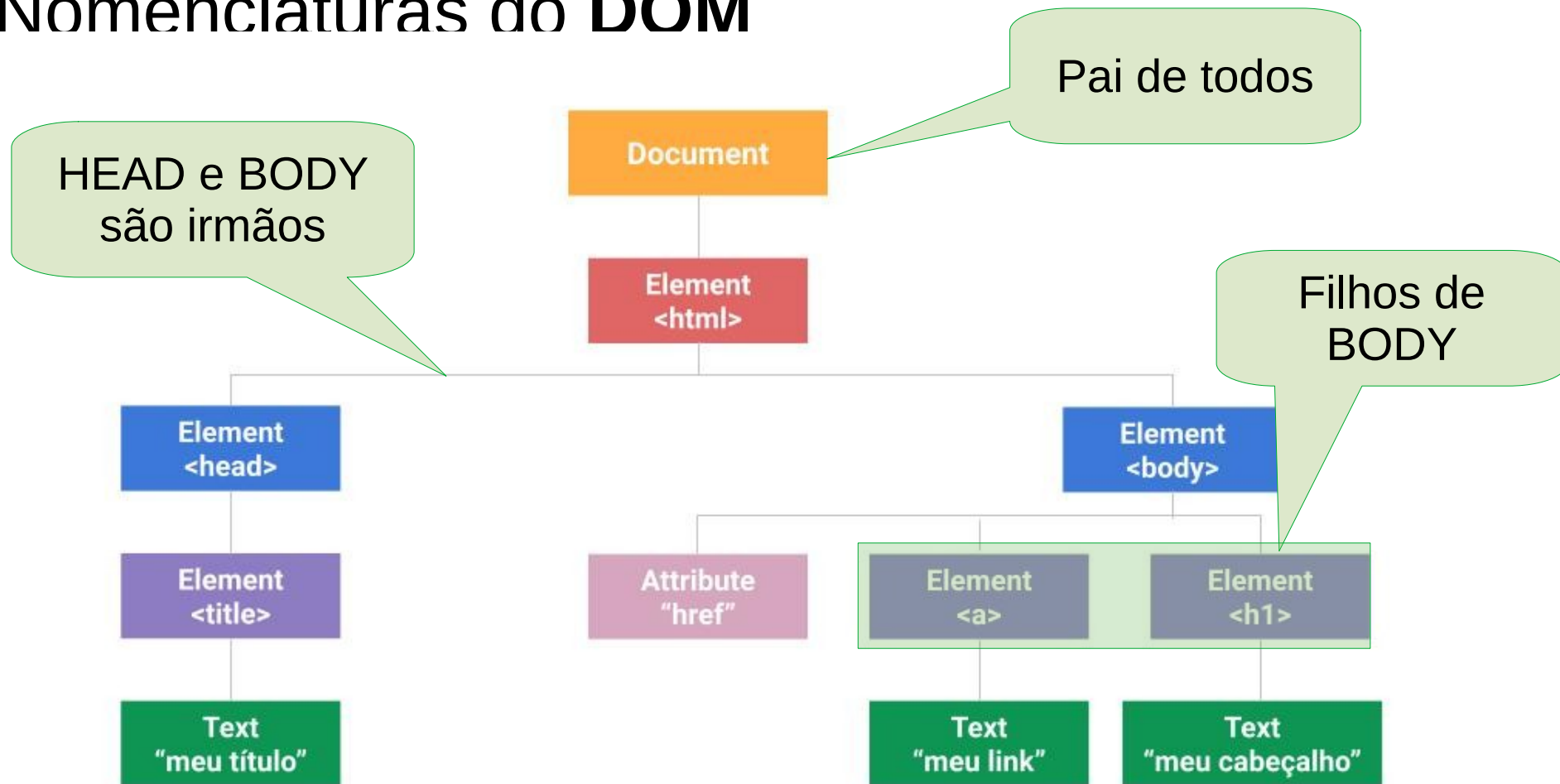
- Página HTML representada pelo **DOM**



```
<html>
  <head>
    <title>meu título</title>
  </head>
  <body>
    <a href="">meu link</a>
    <h1>meu cabeçalho</h1>
  </body>
</html>
```

DOM

- Nomenclaturas do **DOM**



Acessar elementos do DOM

- Acesso ao DOM se dá pelo objeto **document**
 - **document** possui vários métodos para acessar os elementos HTML

```
//Cria um elemento de acordo com a tag, retornando-o para a variável  
var elemento = document.createElement('nome_tag');  
//Busca um elemento pelo seu ID, retornando-o (NULL caso não encontrado)  
var elemento = document.getElementById('id_elemento');  
//Busca elementos pela TAG HTML, retornando um array  
var elementos = document.getElementsByTagName('nome_tag');  
//Busca elementos pela classe do elemento, retornando um array  
var elementos = document.getElementsByClassName('nome_classe');  
//Busca elementos pelo name do elemento, retornando um array  
var elementos = document.getElementsByName('valor_name');
```

Acessar elementos do DOM

- Acesso ao DOM se dá pelo objeto ***document***
 - Acesso pelos seletores de CSS (tag, #id, .classe)

```
//Busca um elemento utilizando o seletor CSS de TAG  
// retorna o primeiro elemento encontrado  
var elemento = document.querySelector('nome_tag');  
  
//Busca elementos utilizando o seletor CSS de CLASSE  
//Retorna um array com todos os elementos encontrados na página  
var elementos = document.querySelectorAll('.classe');
```

Modificar elementos

- Após encontrar os elementos, pode-se alterá-los:

//Altera o conteúdo de um elemento

elemento.innerHTML = 'Conteúdo de um elemento';

//Altera o estilo (CSS) de um elemento

elemento.style.propriedade = 'valor_propriedade';

//Lista de propriedades: https://www.w3schools.com/jsref/dom_obj_style.asp

//Adiciona um elemento como filho de outro elemento

elementoPai.appendChild(elementoFilho);

//Remove um elemento filho de um elemento pai

elementoPai.removeChild(elementoFilho);

Dados de formulários

- Formulários

```
//Retornar o valor preenchido com um input de ID 'txtIdade'  
var idade = document.getElementById('txtIdade').value;  
  
//Verificar se a idade foi preenchida e maior que 18 anos  
if(idade == "" || parseInt(idade) <= 18) {  
    alert("Informe uma idade maior que 18 anos");  
}  
  
//Preencher um valor para o campos 'txtIdade'  
document.getElementById('txtIdade').value = 18;
```

Exercícios

- 1) Crie uma página HTML com dois campos input denominados **Senha** e **Confirmar senha**. Ao clicar em um botão, acione uma função que:
 - 1.1: Valide se ambos os campos foram preenchidos;
 - 1.2: Valide que ambas as senhas informadas são iguais;

Caso as validações não atendam aos requisitos, informe na tela (página HTML) uma mensagem pertinente para alertar o usuário.

Caso a senha e confirmação de senha sejam iguais, exiba um alerta de que as senhas são iguais e limpe os valores digitados nos campos.