

# IW-II

# INTERFACE WEB II

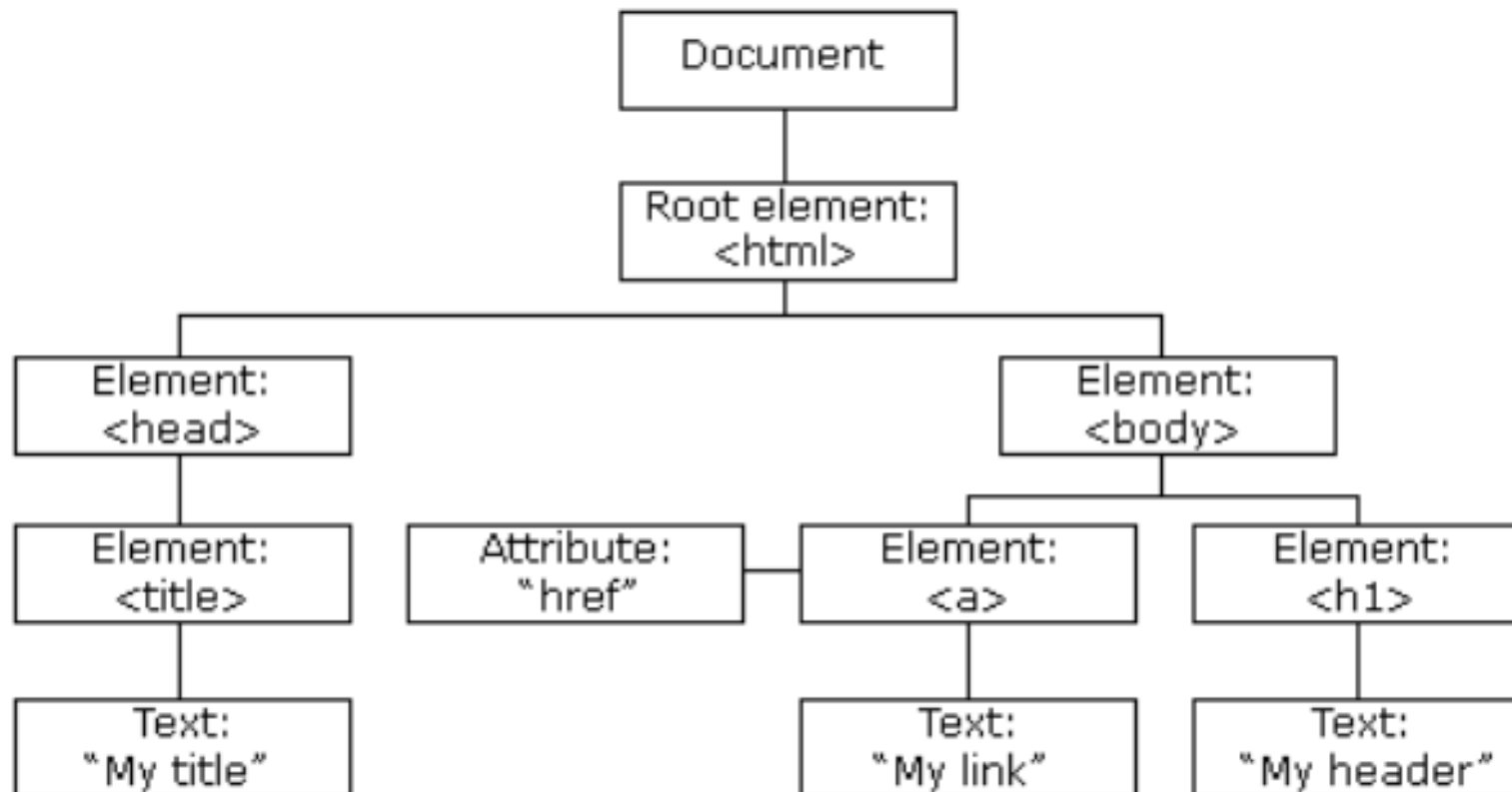
---

Prof. Anderson Vanin

**AULA 06 – MANIPULAÇÃO DO DOM**

# Document Object Model (DOM)

- Quando uma página é carregada, o navegador cria um **Document Object Model** da página.
- O modelo HTML DOM é construído como uma árvore de objetos (**Objects**):



# Document Object Model (DOM)

Com o modelo de objeto, JavaScript tem todo o poder necessário para criar HTML dinamicamente:

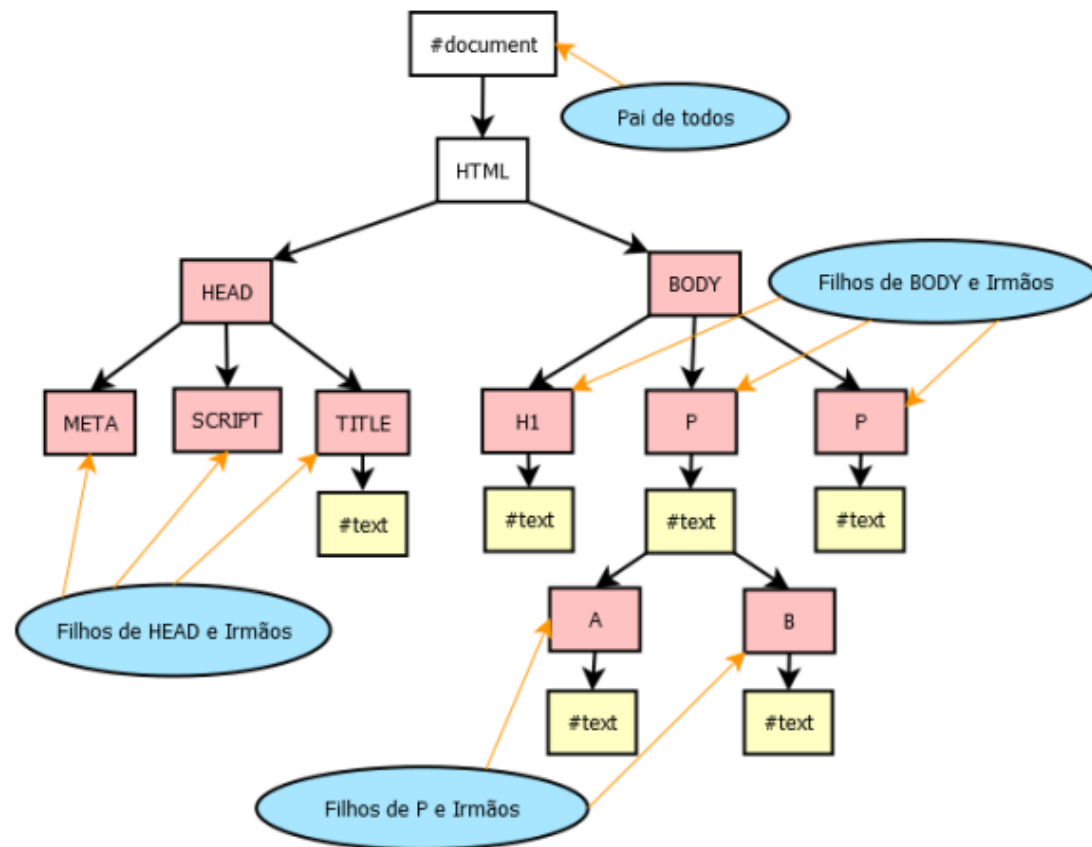
- JavaScript pode alterar todos os elementos HTML na página
- JavaScript pode alterar todos os atributos dos elementos HTML na página
- JavaScript pode alterar todos os estilos CSS
- JavaScript pode remover um elemento HTML e seus atributos
- JavaScript pode adicionar um elemento HTML e seus atributos
- JavaScript pode reagir a todos os eventos que ocorrerem em uma página
- JavaScript pode criar novos eventos na page

# Document Object Model (DOM)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM</title>
    <script src="meuscript.js"></script>
  </head>
  <body>
    <h1 id="id_h1" class="classe_h1">Sou um cabeçalho!</h1>
    <p id="id_p1" class="classe_p">
      Um texto qualquer dentro de uma tag de parágrafo. Aqui também
      temos outras tags, como <a href="#">um link<a>, ou um texto
      <b>em negrito</b>.
    </p>
    
    <p id="id_p2" class="classe_p">
      Este é outro parágrafo.
    </p>
  </body>
</html>
```

# Document Object Model (DOM)

Existem elementos pai (**parent**), filhos (**childs**) e irmãos (**siblings**). Estes elementos são caracterizados na forma como estão na árvore, veja o exemplo na imagem abaixo:



nodeName	id	class
#document		
html		
#HTML		
HEAD		
#text		
META		
#text		
TITLE		
#text		
SCRIPT		
#text		
BODY		
#text		
H1	id_h1	classe_h1
#text		
P	id_p	classe_p
#text		
A		
#text		
B		
#text		
#text		
P	id_p2	classe_p2
#text		
#text		

# Encontrando elementos através da tag

- Para localizar um elemento pelo nome da *tag*, utiliza-se o método ***getElementsByTagName***.
- O parâmetro a ser passado é o nome da *tag* que se deseja buscar.
- O método sempre irá retornar um *array* contendo os elementos daquela determinada *tag* que houverem na página.

## **getElementsByTagName**

Para parágrafos, passa-se o parâmetro **p**

- **document.getElementsByTagName("p")**

Para divs, passa-se o parâmetro **div**

- **document.getElementsByTagName("div")**

E assim por diante...

# Encontrando elementos através da tag

HTML	<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;   &lt;title&gt;Document&lt;/title&gt;   &lt;script src="meuscript.js"&gt;&lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;h1 id="id_h1" class="classe_h1"&gt;Mundo Mágico do Javascript&lt;/h1&gt;   &lt;p id="corvinal" class="classe_p"&gt;Corvinal&lt;/p&gt;   &lt;p id="grifinoria" class="classe_p"&gt;Grifinória&lt;/p&gt;   &lt;p id="sonserina" class="classe_p"&gt;Sonserina&lt;/p&gt;   &lt;p id="lufalufa" class="classe_p"&gt;Lufa-Lufa&lt;/p&gt;   &lt;button type="button" id="bdefinir" onclick="recuperarPs()"&gt;Recuperar &lt;/button&gt; &lt;/body&gt; &lt;/html&gt;</pre>
JAVASCRIPT	<pre>//criar uma variável que vai receber um array com todas as tags p var paragrafos = document.getElementsByTagName("p"); console.log(paragrafos);</pre>

meuscript.js

# Encontrando elementos através da tag

## Mundo Mágico do Javascript

Corvinal

Grifinória

Sonserina

Lufa-Lufa

Recuperar

DevTools is now available in Portuguese! [Always match Ch](#)

Elements Console Sources Network Per  
top Filter

```
▼ HTMLCollection [] ⓘ  
  ▶ 0: p#corvinal.classe_p  
  ▶ 1: p#grifinoria.classe_p  
  ▶ 2: p#sonserina.classe_p  
  ▶ 3: p#lufalufa.classe_p  
  ▶ corvinal: p#corvinal.classe_p  
  ▶ grifinoria: p#grifinoria.classe_p  
  ▶ lufalufa: p#lufalufa.classe_p  
  ▶ sonserina: p#sonserina.classe_p  
  length: 4  
  ▶ [[Prototype]]: HTMLCollection
```

>



# Encontrando elementos através da classe

`document.getElementsByClassName("classe")`

```
//criar uma variável que vai receber um array com todas as classes: "classe_p"  
var classes = document.getElementsByClassName("classe_p");  
console.log(classes);
```

```
▼ HTMLCollection [] ⓘ  
  ▶ 0: p#corvinal.classe_p  
  ▶ 1: p#grifinoria.classe_p  
  ▶ 2: p#sonserina.classe_p  
  ▶ 3: p#lufalufa.classe_p  
  ▶ corvinal: p#corvinal.classe_p  
  ▶ grifinoria: p#grifinoria.classe_p  
  ▶ lufalufa: p#lufalufa.classe_p  
  ▶ sonserina: p#sonserina.classe_p  
  length: 4  
  ▶ [[Prototype]]: HTMLCollection
```

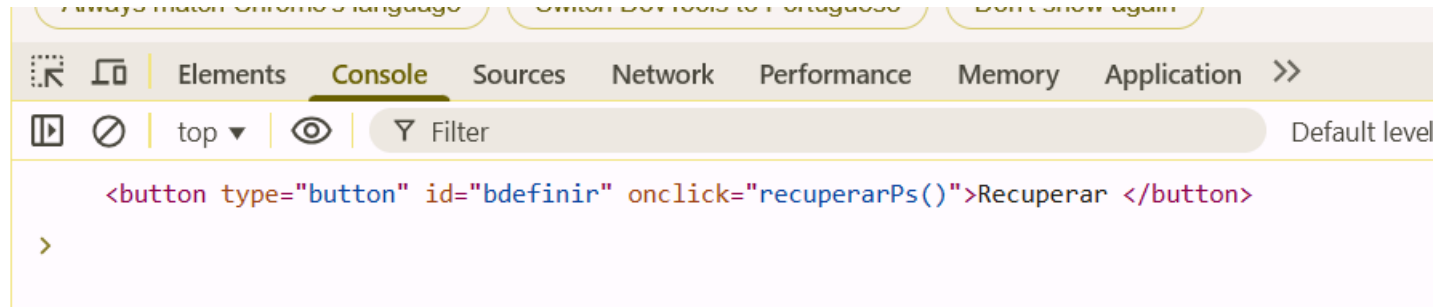
>

# Encontrando elementos através de seletores CSS

```
document.querySelector('button[type="button"]');
```

**IMPORTANTE: A CHAMADA DESTES SCRIPT DEVE FICAR ABAIXO DO HTML, POIS O BOTÃO DEVE EXISTIR PARA SER CHAMADO!**

```
//criar uma variável que vai receber um elemento pelo seletor: "type=button"  
var botao = document.querySelector('button[type="button"]');  
console.log(botao);
```



# Alterando propriedades

- Exemplo alterar cor dos parágrafos:

```
//criar uma variável que vai receber um array com todas as tags: "p"
var paragrafos = document.getElementsByTagName("p");
// console.log(paragrafos);
var cores = ["blue", "red", "green", "yellow"];
var i;
for (i=0; i<paragrafos.length; i++){
    paragrafos[i].style.backgroundColor = cores[i];
}
```

## Mundo Mágico do Javascript

Corvinal

Grifinória

Sonserina

Lufa-Lufa

Recuperar

# Alterando propriedades

- Exemplo alterar estilo de um elemento:

<> exemplo03.html >  html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="estilo3.css">
8 </head>
9 <body>
10  <h1 id="id_h1" class="classe_h1">Mundo Mágico do Javascript</h1>
11  <p id="escondido" class="oculto">Estou escondido</p>
12  <p id="grifinoria" class="classe_p">Grifinória</p>
13  <p id="sonserina" class="classe_p">Sonserina</p>
14  <p id="lufalufa" class="classe_p">Lufa-Lufa</p>
15  <button type="button" id="botao" onclick="aparecer()">Aparecer </button>
16  <script src="meuscript3.js"></script>
17 </body>
18 </html>
```

JS meuscript3.js >  aparecer

```
1 function aparecer(){
2   var paragrafos = document.getElementsByClassName("oculto");
3   var i;
4   for (i=0; i<paragrafos.length; i++){
5     paragrafos[i].style.display = "block";
6   }
7 }
```

# estilo3.css > ...

```
1 .oculto{
2   display: none;
3 }
4
```

← → ↻ ⓘ Arquivo C:/Users/Anderson/Desktop/ETEC%20

## Mundo Mágico do Javascript

Grifinória

Sonserina

Lufa-Lufa

Aparecer

## Mundo Mágico do Javascript

Estou escondido

Grifinória

Sonserina

Lufa-Lufa

Aparecer

# Criando elementos no documento

Há ainda funções para criar e remover elementos da página, através dos métodos a seguir.

## **document.createElement(element)**

- cria um elemento HTML. Deve-se passar por parâmetro a tag que se deseja criar
- o método retorna o elemento criado
- Exemplo: `document.createElement("p")` , cria um elemento do tipo parágrafo

## **element.removeChild(elementToRemove)**

- Remove o filho de um elemento

## **element.appendChild(newElement)**

- Adiciona um novo elemento como último filho do elemento da chamada do método

## **document.replaceChild(newElement, oldElement)**

- Substitui um elemento por outro

Para outras funções: [http://www.w3schools.com/js/js\\_htmldom\\_document.asp](http://www.w3schools.com/js/js_htmldom_document.asp)