



# Programação Novas Tecnologias

Prof : Pedro Albino



# Tópicos da Aula.

- getElementById
- getElementsByTagName
- getElementsByClassName
- addEventListener

# JavaScript

Com o DOM HTML o JavaScript pode acessar e alterar todos os elementos de um documento HTML.

# JavaScript

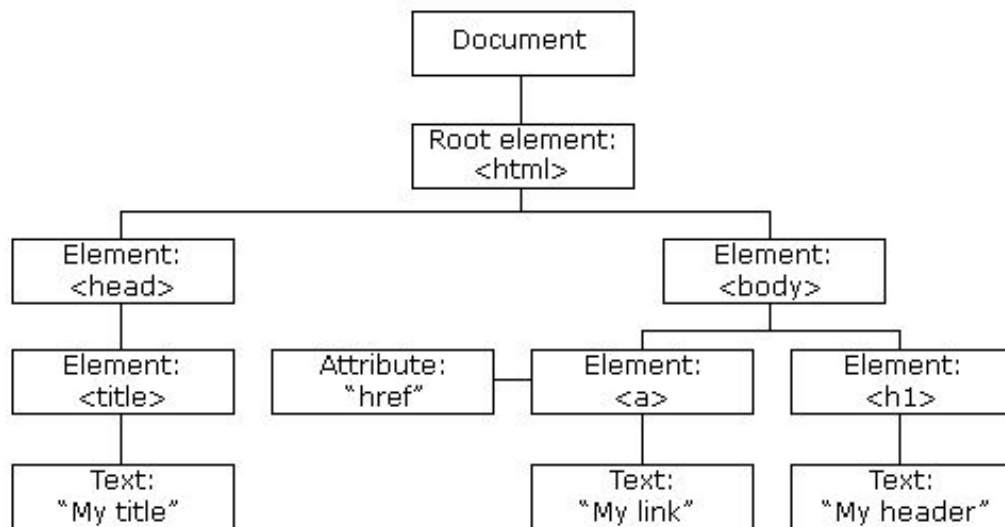
## O DOM HTML (Document Object Model)

Quando uma página da web é carregado, o navegador cria um **Document Object Modelo** da página.

O modelo **DOM HTML** é construído como uma árvore de objetos :

# JavaScript

## A árvore de objetos DOM HTML



# JavaScript

**Com o modelo de objeto, o JavaScript obtém todo o poder de que precisa para criar HTML dinâmico:**

- JavaScript pode alterar todos os elementos HTML na página
- JavaScript pode alterar todos os atributos HTML na página
- JavaScript pode mudar todos os estilos CSS na página
- JavaScript pode remover elementos e atributos HTML existentes
- JavaScript pode adicionar novos elementos e atributos HTML
- JavaScript pode reagir a todos os eventos HTML existentes na página
- JavaScript pode criar novos eventos HTML na página

# JavaScript

## O que veremos ?

- Como alterar o conteúdo dos elementos HTML
- Como reagir a eventos HTML DOM
- Como alterar o estilo (CSS) dos elementos HTML
- Como adicionar e excluir elementos HTML

# JavaScript

## Métodos DOM HTML

- Os métodos HTML DOM são ações que você pode executar (em elementos HTML).
- Propriedades HTML DOM são valores de elementos HTML que você pode definir ou alterar.



# JavaScript

## A Interface de Programação DOM

- O HTML DOM pode ser acessado com JavaScript.
- No DOM, todos os elementos HTML são definidos como objetos .
- A interface de programação são as propriedades e métodos de cada objeto.
- Uma propriedade é um valor que você pode obter ou definir como alterar o conteúdo de um elemento HTML.
- Um método é uma ação que você pode realizar como adicionar ou excluir um elemento HTML.

# JavaScript

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Page</h2>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Aula Dom";
</script>

</body>
</html>
```

## My First Page

Aula Dom

No exemplo acima, `getElementById` é um **método**, enquanto `innerHTML` é uma propriedade

# JavaScript

## O método `getElementById`

A maneira mais comum de acessar um elemento HTML é usar o `id` do elemento.

No exemplo acima, o `getElementById` método usado `id="demo"` para encontrar o elemento.

# JavaScript Arrays

## A propriedade innerHTML

A maneira mais fácil de obter o conteúdo de um elemento é usando a **innerHTML** propriedade.

A **innerHTML** propriedade é útil para obter ou substituir o conteúdo de elementos HTML.

# JavaScript

## Documento DOM HTML

O objeto de documento DOM HTML é o proprietário de todos os outros objetos em sua página da web.

# JavaScript Arrays

## **O objeto de documento DOM HTML**

O objeto de documento representa sua página da web.

Se você deseja acessar qualquer elemento em uma página HTML, sempre comece acessando o objeto de documento.

# JavaScript Arrays

## Encontrando Elementos HTML

Método	Descrição
<code>document.getElementById(id)</code>	Encontre um elemento por id de elemento
<code>document.getElementsByTagName(name)</code>	Encontre elementos por nome de tag
<code>document.getElementsByClassName(name)</code>	Encontre elementos por nome de classe

# JavaScript Arrays

## Alterando Elementos HTML

Propriedade	Descrição
<code>element.innerHTML = novo conteúdo html</code>	Alterar o HTML interno de um elemento
<code>element.attribute = novo valor</code>	Altere o valor do atributo de um elemento HTML
<code>element.style.property = novo estilo</code>	Alterar o estilo de um elemento HTML



# JavaScript Arrays

## Adicionando e Excluindo Elementos

Método	Descrição
<code>document.createElement(elemento)</code>	Crie um elemento HTML
<code>document.removeChild(elemento)</code>	Remove um elemento HTML
<code>document.appendChild(elemento)</code>	Adiciona um elemento HTML
<code>document.replaceChild(new, old)</code>	Substitua um elemento HTML

# JavaScript Arrays

## Encontrando Elementos HTML

Frequentemente, com JavaScript, você deseja manipular elementos HTML.

Para fazer isso, você deve primeiro encontrar os elementos. Existem várias maneiras de fazer isso:

- Encontrar elementos HTML por id
- Encontrar elementos HTML por nome de tag
- Encontrar elementos HTML por nome de classe
- Encontrar elementos HTML por seletores CSS
- Encontrar elementos HTML por coleções de objetos HTML

# JavaScript

## Encontrando Elemento HTML por Id

A maneira mais fácil de encontrar um elemento HTML no DOM é usando o id do elemento.

Este exemplo encontra o elemento com `id="pmain"`:

```
document.getElementById("pmain");
```

# JavaScript

Encontrar um elemento usando o ID.

Se o elemento for encontrado, o método retorna o elemento como um objeto

Se o elemento não for encontrado, retorna **null**.

## Programação e Novas Tecnologias

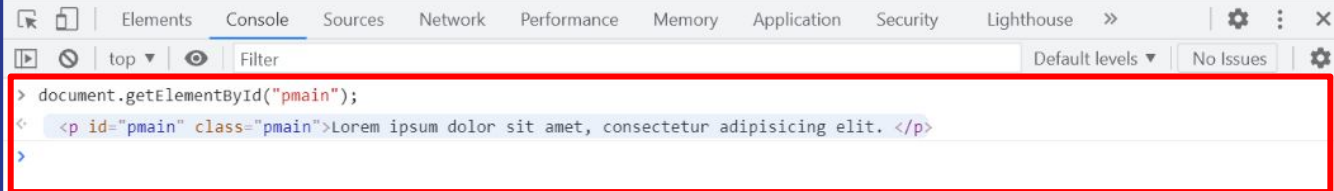
p#pmain.pmain 1350 x 18

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



The screenshot shows a web browser's developer console with the 'Console' tab selected. A red rectangle highlights the following content:

```
> document.getElementById("pmain");  
< p id="pmain" class="pmain">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>  
>
```

The console also shows a toolbar with icons for 'top', 'Filter', and 'Default levels', and a status bar indicating 'No Issues'.

# JavaScript

## Localização de elementos HTML por nome da tag

```
document.getElementsByTagName('p');
```

# JavaScript

## Programação e Novas Tecnologias

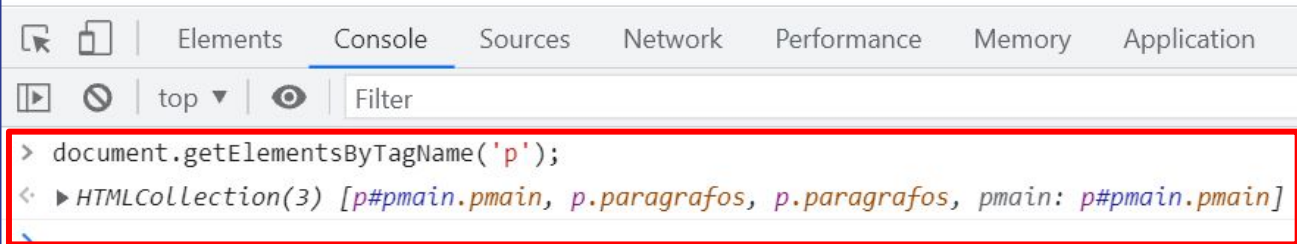
### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



# JavaScript

Foi encontrado o elemento usando `getElementsByName()`, e passado a posição do elemento que foi desejado a mudança.

## Programação e Novas Tecnologias

p#pmain.pmain 1350 x 18

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

subtitulo

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
document.getElementsByTagName('p');
HTMLCollection(3) [p#pmain.pmain, p.paragrafos, p.paragrafos, pmain: p#pmain.pmain]
document.getElementsByTagName('p')[0];
<p id="pmain" class="pmain">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
```

# JavaScript

## Programação e Novas Tecnologias

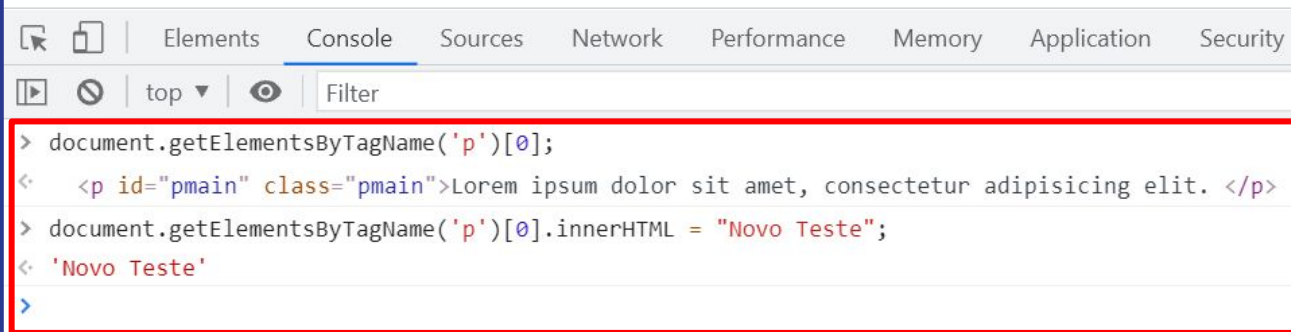
### subtitulo

Novo Teste

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following JavaScript code and its output:

```
> document.getElementsByTagName('p')[0];  
< <p id="pmain" class="pmain">Lorem ipsum dolor sit amet, consectetur adipisicing elit. </p>  
> document.getElementsByTagName('p')[0].innerHTML = "Novo Teste";  
< 'Novo Teste'  
>
```



# JavaScript

## Encontrando Elementos HTML por Nome de Classe

Se você quiser encontrar todos os elementos HTML com o mesmo nome de classe, use `getElementsByClassName()`.

# JavaScript

## Programação e Novas Tecnologias

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the result of a JavaScript command: `document.getElementsByClassName('pmain');`. The output is an `HTMLCollection` containing two elements: `p#pmain.pmain` and `pmain: p#pmain.pmain`. The console interface includes standard icons for back, forward, and search, as well as tabs for Elements, Console, Sources, Network, and Performance.

```
> document.getElementsByClassName('pmain');  
< ▶ HTMLCollection [p#pmain.pmain, pmain: p#pmain.pmain]  
>
```

# JavaScript

## Programação e Novas Tecnologias

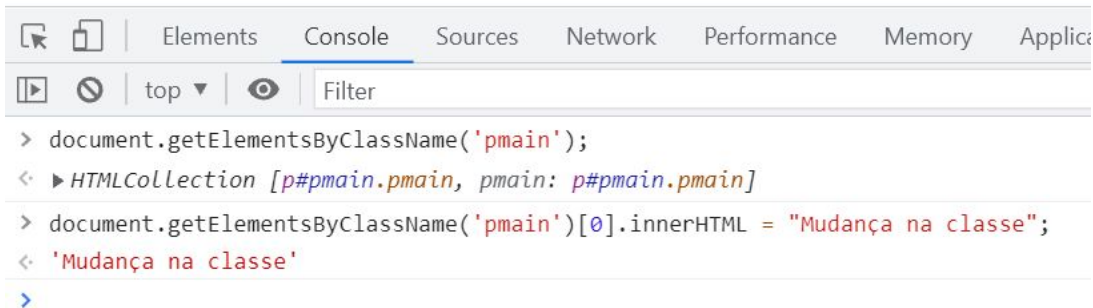
### subtitulo

Mudança na classe

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following JavaScript code and its output:

```
> document.getElementsByClassName('pmain');  
◀ ▶HTMLCollection [p#pmain.pmain, pmain: p#pmain.pmain]  
> document.getElementsByClassName('pmain')[0].innerHTML = "Mudança na classe";  
◀ ▶'Mudança na classe'  
>
```

# JavaScript

Exercícios realizar mudanças em uma página em HTML usando o que já foi visto.

# JavaScript

## Eventos JavaScript DOM HTML

O HTML DOM permite que o JavaScript reaja a eventos HTML:

Reagindo aos eventos

Um JavaScript pode ser executado quando ocorre um evento, como quando um usuário clica em um elemento HTML.

# JavaScript

Para executar o código quando um usuário clica em um elemento, adicione o código JavaScript a um atributo de evento HTML:

## **onclick=()**

Exemplos de eventos HTML:

- Quando um usuário clica no mouse
- Quando uma página da web é carregada
- Quando uma imagem foi carregada
- Quando o mouse se move sobre um elemento
- Quando um campo de entrada é alterado
- Quando um formulário HTML é enviado
- Quando um usuário pressiona uma tecla

# JavaScript

Neste exemplo, o conteúdo do **<h1>** elemento é alterado quando um usuário clica nele:

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="this.innerHTML = 'Ooops!'">Clique no texto</h1>

</body>
</html>
```

**Clique no texto**

**Ooops!**

# JavaScript

Neste exemplo, uma **função** é chamada a partir do manipulador de eventos:

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="changeText(this)">Clique no texto</h1>

<script>
function changeText(id) {
  id.innerHTML = "Ooops!";
}
</script>

</body>
</html>
```

Clique no texto

Ooops!



# JavaScript

Exemplo de soma usando o `onclick()`

```
<!DOCTYPE html>
<html>
<head>
  <title>Desafio 01</title>
</head>
<body>

  <h1>Soma</h1>
  <input id="n1" type="number"/> +
  <input id="n2" type="number">
  <button onclick="calcula()">Calcular</button>

  <p>0 resultado é: <span id="r" ></span></p>
  <script src="js/soma.js"></script>
</body>
</html>
```

```
//pegando o id N1
var n1 = document.getElementById('n1');

//pegando o id N2
var n2 = document.getElementById('n2');

//pegando o id r
var r = document.getElementById('r');

//função calcula
function calcula(){
  //valueAsNumber - interpreta o valor em um numero
  var resultado = n1.valueAsNumber + n2.valueAsNumber;

  if(isNaN(resultado)){
    r.innerText = ' ';
  }else{
    r.innerText = resultado;
  }
}
```

# JavaScript Exercícios

Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:

- para homens:  $(72.7 * h) - 58$ ;
- para mulheres:  $(62.1 * h) - 44.7$ .

# JavaScript Exercícios

Escreva um programa que receba o nome a turma e calcule a média entre dois valores e escreva o resultado da média. Se a média for maior ou igual 6 enviar uma mensagem de “**aprovado**”.

# JavaScript Exercícios

Faça um algoritmo que receba os valores A, B, C e imprima na tela se a soma de A + B for menor que C.

# JavaScript Exercícios

Encontrar o dobro de um número caso ele seja positivo e o seu triplo caso seja negativo, imprimindo o resultado.

# JavaScript Exercícios

Faça um algoritmo que leia um valor e some 5 caso seja par ou some 8 caso seja ímpar, imprimir o resultado desta operação.

# JavaScript Exercícios

O IMC – Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é  $IMC = \text{peso} / (\text{altura} * \text{altura})$  Elabore um algoritmo que leia o peso e a altura de um adulto e mostre sua condição de acordo com a tabela abaixo. IMC em adultos Condição

- Abaixo de 18,5 Abaixo do peso
- Entre 18,5 e 25 Peso normal
- Entre 25 e 30 Acima do peso
- Acima de 30 obeso

# JavaScript

## O evento onchange

O **onchange** evento é freqüentemente usado em combinação com a validação de campos de entrada.

Abaixo está um exemplo de como usar o onchange. A **toUpperCase()** função será chamada quando um usuário alterar o conteúdo de um campo de entrada.



# JavaScript

## **addEventListener**

Manipula eventos ao elemento especificado.

# JavaScript

```
<!DOCTYPE html>
<html>
<body>

Nome: <input type="text" id="fname" onchange="upperCase()">

<script>
function upperCase() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>

</body>
</html>
```

Nome:

# JavaScript

## Os eventos onmouseover e onmouseout

Os eventos **onmouseover** e **onmouseout** podem ser usados para acionar uma função quando o usuário passa o mouse sobre ou fora de um elemento HTML:

# JavaScript

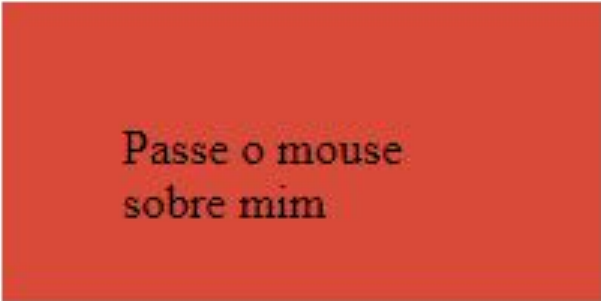
```
<!DOCTYPE html>
<html>
<body>

<div onmouseover="mOver(this)" onmouseout="mOut(this)"
style="background-color: #D94A38;width:120px;height:20px;padding:40px;">
Passe o mouse sobre mim</div>

<script>
function mOver(obj) {
    obj.innerHTML = "Obrigado"
}

function mOut(obj) {
    obj.innerHTML = "Passe o mouse sobre mim"
}
</script>

</body>
</html>
```



Passe o mouse  
sobre mim

# JavaScript

Encontrar um elemento usando o ID.

Se o elemento for encontrado, o método retorna o elemento como um objeto

Se o elemento não for encontrado, retorna **null**.

## Programação e Novas Tecnologias

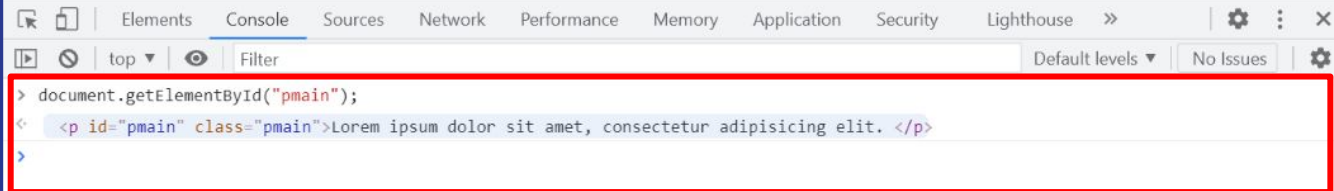
p#pmain.pmain 1350 x 18

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### subtitulo

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



```
> document.getElementById("pmain");  
< p id="pmain" class="pmain">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>  
>
```