



**CENTRO UNIVERSITÁRIO SENAC - SANTO AMARO**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROGRAMAÇÃO WEB**

**ALUNO: PEDRO HENRIQUE ANDREATTI**

**PROFESSOR: CARLOS HENRIQUE VERISSIMO PEREIRA**

**CONCEITOS SOBRE DOCUMENTO OBJECT MODEL E  
JAVASCRIPT**

## SUMÁRIO

<b>1.Conceitos.....</b>	<b>1</b>
<b>1.1 O que é o DOM</b>	
<b>?.....</b>	<b>1</b>
<b>1.2 Relação entre DOM e Javascript.....</b>	<b>1</b>
<b>2.Elementos da implementação do JavaScript/DOM.....</b>	<b>2</b>
<b>3.Explicando a interação do JavaScript com o HTML.....</b>	<b>3</b>
<b>4.Bibliografia.....</b>	<b>4</b>

# Conceitos

## • O que é o DOM ?

O Document Object Model(DOM)é uma interface de aplicação (API) independente de linguagem para trabalho com documentos XML e HTML. No navegador, você trabalha principalmente com documentos HTML, ainda que não seja raro que aplicações web recuperem documentos XML e utilizem as APIs DOM para acessar dados desses documentos.

Desenvolvemos os codigos no JavaScript e essas informações aparecem na pagina com o uso do DOM que é nada mais nada menos que um modelo de representação por objetos podendo manipular elementos de um documento web, é pra isso que ele serve para ver as informações na pagina e manipular.

Atravéz do DOM, o JavaScript você pode modificar todos os elementos de uma página e realizar ações como:

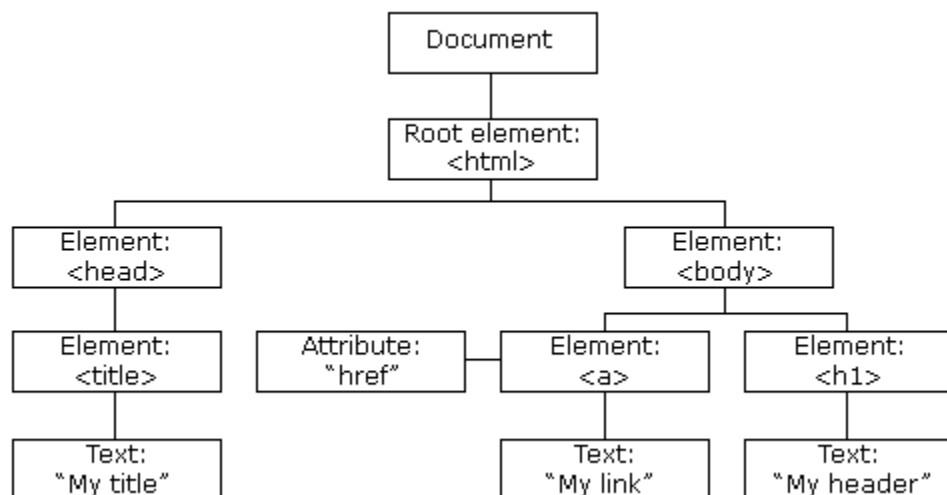
**-Adicionar ou remover elementos HTML**

**-Altera elementos HTML da página**

**-Alterar atributos dos elementos HTML**

**-Alterar o CSS**

**-Reagir a eventos na página(clique,rolagem,input em formulários,passar o cursor sobre elementos, etc)**



- **Relação entre DOM e Javascript**

O JavaScript é uma linguagem de programação que é usada para fazer codigos, criar o script do site em si e o DOM não é uma linguagem de programação, ele é um modelo de representação por objetos, definindo a estrutura que os codigos construidos no JavaScript será lido pelo navegador.

Ou seja sem o DOM, pode ser qualquer linguagem de programação que for se não tiver o DOM não conseguiremos representar essas informações no navegador, o Dom representa a estrutura e os elementos da página e o JavaScript acessa e manipula esses conteúdos.

## • Elementos da implementação do JavaScript/DOM

A primeira coisa que você precisa fazer é encontrar o elemento com o qual quer trabalhar . Para isso , ao trabalhar com o DOM , você pode procurá - las pelo atributo id , pelo nome da tag ( h1 , p , img , div , etc. ) ou pelo nome da classe .

Contudo , para manipular elementos do documento , eles precisam de algum modo obter ou selecionar os objetos Element que se referem a esses elementos de documento . O DOM define várias maneiras de selecionar elementos em HTML , são eles `getElementById` ; `getElementsByTagName` ; `getElementsByClassName` .

### -getElementById()

Esse método retorna o elemento que estiver contendo o nome do ID passado. Como os IDs devem ser únicos, é um método muito útil para pegar apenas o elemento desejado.

```
var myStart = document.getElementById('start');
```

`myStart`: elemento específico que se equipara com o seletor passado.

`start`: seletor passado, caso não houvesse nenhum ele retornaria null.

### -getElementsByTagName()

Na mesma maneira do método acima, ele também retorna uma `HTMLCollection` mas com uma diferença: esse método retorna todos elementos contendo a tag name passada.

```
var parag = document.getElementsByTagName('p');
```

parag: elemento específico que se equipara com o seletor passado.

p: tag name passada.

## **-getElementsByClassName**

assim como o `getElements`

`ByTagName` , retorna um array de elementos .

## **-querySelector**

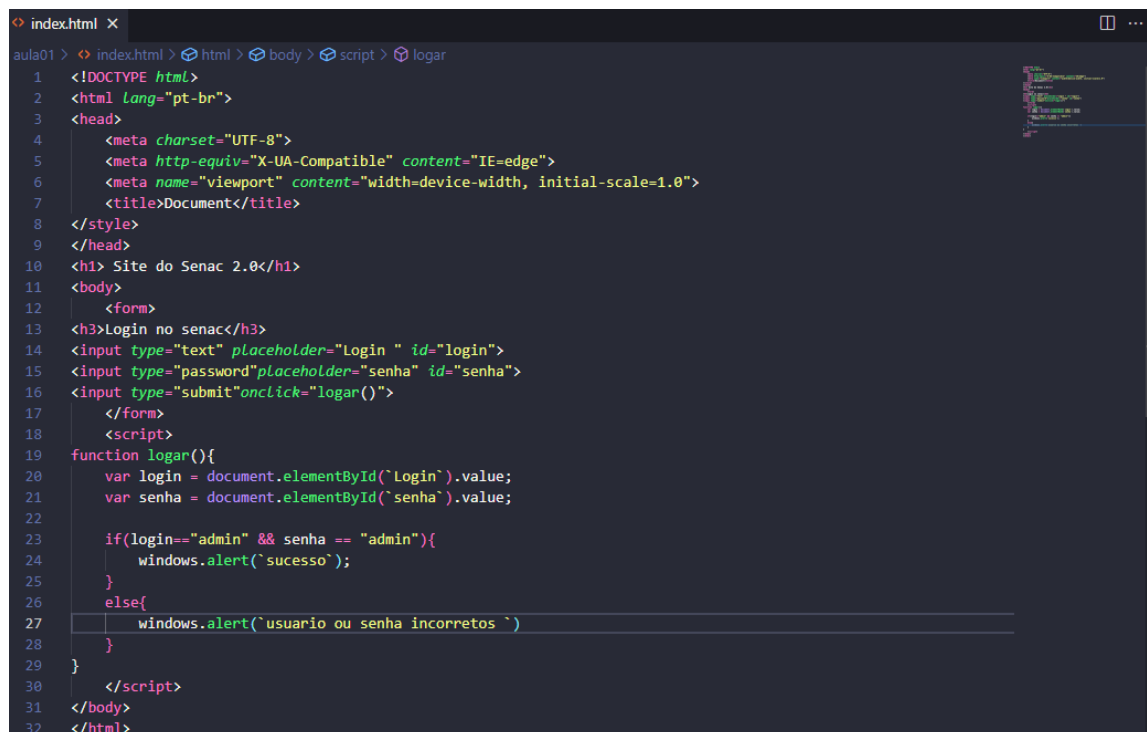
Ele faz com que retorne o primeiro elemento do documento que corresponde ao grupo especificado de seletores.

Exemplo

Neste exemplo, o primeiro elemento no documento com a classe "myclass" é retornado:

```
var el = document.querySelector(".myclass");
```

## Explicando a interação do JavaScript com o HTML



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </style>
9 </head>
10 <h1> Site do Senac 2.0</h1>
11 <body>
12   <form>
13   <h3>Login no senac</h3>
14   <input type="text" placeholder="Login " id="login">
15   <input type="password" placeholder="senha" id="senha">
16   <input type="submit" onClick="logar()">
17   </form>
18   <script>
19   function logar(){
20     var login = document.getElementById('Login').value;
21     var senha = document.getElementById('senha').value;
22
23     if(login=="admin" && senha == "admin"){
24       windows.alert('sucesso');
25     }
26     else{
27       windows.alert('usuario ou senha incorretos `');
28     }
29   }
30   </script>
31 </body>
32 </html>
```

-A implementação começa na linha 18 quando abre o `<script>`

-Na linha 19 criamos uma função para validar se o login está correto e com a interação do DOM conseguimos uma janela mostrando essa informação e a pessoa pode interagir colocando suas informações

-Na linha 24 , é puxado dos elementos da árvore do DOM, o `window.alert` que usando o Javascript e o DOM, envia uma mensagem em janela para o Usuário.

## Referencial Bibliografico

Foram utilizados livros academicos

-JavaScript:interatividade para web

<https://www.bibliotecadigitalsenac.com.br/?from=busca%3FcontentInfo%3D3602%26term%3Djavascript&page=20&section=0#/legacy/3602>

-JavaScript de Alto Desempenho

-<https://books.google.com.br/books?hl=pt-BR&lr=&id=4B5WDQAAQBAJ&oi=fnd&pg=PA19&dq=rela%çao+entre+DOM+e+JavaScript&ots=0MOqezOTzN&sig=UjwRo6Jm9S-n8stwY3BOEPkhWmA#v=onepage&q&f=false>

-JavaScript: O Guia Definitivo

<https://books.google.com.br/books?hl=pt-BR&lr=&id=zWNyDgAAQBAJ&oi=fnd&pg=PR1&dq=JavaScript+&ots=IAAcE-JamM&sig=B4AJaf6tkTZzIKtelqvMFMYqwG4#v=onepage&q=JavaScript&f=false>