



**CENTRO UNIVERSITÁRIO SENAC**  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**PROGRAMAÇÃO WEB**

**RELAÇÃO ENTRE “DOCUMENT OBJECT MODEL (DOM)” E JAVASCRIPT**

**ALUNO(A): BRUNA DOMINGOS BRAVO**

**ORIENTADOR: CARLOS HENRIQUE VERÍSSIMO PEREIRA**

**São Paulo**

**2022**

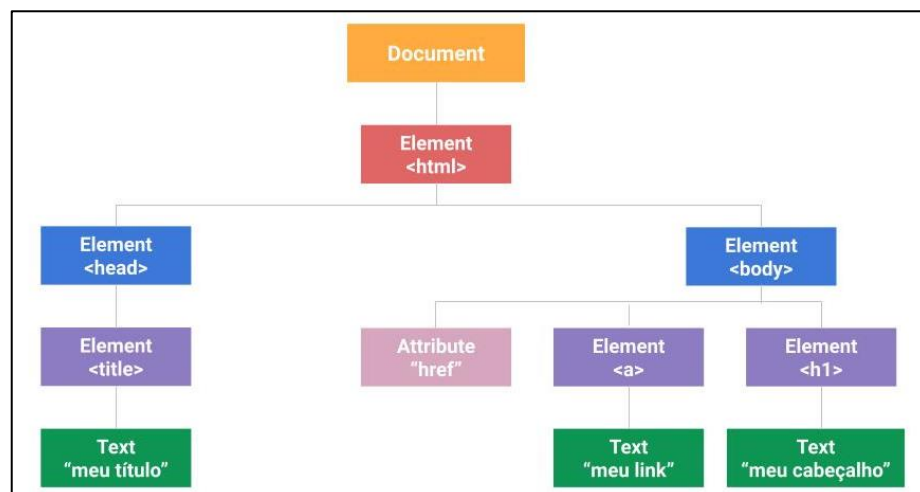
## O que é DOM?

Document Object Model - mais conhecido como DOM - traduzido para o português como “Modelo de Documento por Objetos” é uma interface de programação que representa como um documento HTML e XML é lido pelo navegador Web. Após ser lido pelo navegador, são criados objetos que realizam a representação estruturada do documento e definem os meios de como essa estrutura pode ser acessada, modificada e manipulada e, desta forma, pode-se ocorrer alterações na estrutura, no estilo e no conteúdo do documento. Tal manipulação torna-se possível devido a sua organização hierárquica, ou seja, em forma de árvore, na qual cada objeto dispõe de métodos e atributos.

Na árvore DOM, cada elemento (*element*) do documento é representado como um objeto, denominado nó.

- Root: nó que representa o elemento raiz <html>
- Filhos (*children*): nó que representa elementos que estão dentro de outro
- Pai (*parent*): nó que representa o elemento que possui o nó “filho”
- Irmãos (*siblings*): nó que representa um ou mais elementos filhos do mesmo pai

Estrutura da árvore DOM:



*Document*: objeto raiz, o qual representa o próprio documento HTML;

*Element*: tags que estão em arquivos HTML ou XML e se transformam em elementos da árvore;

*Text*: conteúdo das tags. Texto que fica localizado entre os elementos;

*Attribute*: objeto que representa um atributo contido em um elemento HTML.

## ***Relação entre DOM e JavaScript***

Primeiramente, vale lembrar que a linguagem JavaScript, conhecida como a linguagem de scripting para páginas web, é uma linguagem de programação de alto nível, dinâmica, flexível, interpretada e conveniente para estilos de programação orientada a objetos. Já o DOM, como visto anteriormente, é uma interface de representação orientada a objetos da página web, a qual pode ser modificada com uma linguagem de script, como por exemplo, JavaScript.

Logo, o DOM define a estrutura na qual o documento gerado através da linguagem JavaScript será lido pelo navegador e, sem o DOM, tanto o JavaScript quanto qualquer outra linguagem não teria um modelo para representar seus objetos, pois tal linguagem não possuiria uma noção da página web, nem dos documentos HTML ou XML nem de seus componentes.

Portanto, resumidamente, a relação entre o DOM e o JavaScript se deve ao fato de que o DOM representa a estrutura e os objetos de um documento e o JavaScript pode acessar, modificar e manipular os dados que o compõe.

## ***Elementos de implementação DOM/JavaScript***

O DOM possui níveis diferentes de objetos, sendo eles: *window*, *navigator*, *history* e *document*. Entretanto, é no *document* onde estão localizados os objetos disponíveis no HTML para manipulação via JavaScript.

A seguir, algumas das principais funções para selecionar elementos HTML:

- `getElementById()`: recupera um elemento através do id;
- `getElementsByClassName()`: retorna um *HTMLCollection* de todos elementos que estiverem contendo o nome da classe passada;
- `getElementsByTagName()`: assim como a função acima, ele também retorna uma *HTMLCollection*, entretanto ele retorna todos elementos contendo a tag name passada;
- `querySelector()`: recupera um elemento através de um seletor;
- `querySelectorAll()`: esta função opera de maneira similar a anterior, porém retorna vários elementos.

## Caso Prático JavaScript/DOM

```
1  <!--
2      SENAC - TADS - Programação Web
3      Aula #02 - Introdução ao JavaScript
4      Objetivos deste código: Demonstrar JS em Páginas html
5      "Interação com Usuário - obtendo Dados"
6  -->
7
8  <!DOCTYPE html>
9  <html lang="pt-br">
10 <head>
11     <meta charset="UTF-8">
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <title>Dados com JS</title>
14     <style>
15         body { font: 12pt Arial; }
16         button { font-size: 12pt; padding: 30px; }
17     </style>
18 </head>
19 <body>
20     <h1>Senac - TADS - PW - 2º Semestre </h1>
21     <h2>Aula #02 - Introdução ao JS</h2>
22     <h3>Trabalhando com Dados - PGM#4</h3>
23     <button onclick="iniciarInteracao()">Clique para começar</button>
24     <section id="resultado">
25         <p>Observe este ponto: Aqui irá aparecer o resultado... </p>
26     </section>
27     <script>
28         function iniciarInteracao() {
29             let nome = window.prompt('Qual é o seu nome?')
30             let res = window.document.getElementById('resultado')
31
32             res.innerHTML = `<p>Olá, <strong>${nome}</strong>! Bem vindo ao Curso PW do Senac! &#x1F596;`
33         }
34     </script>
35 </body>
36 </html>
```

- Linha 27: abertura do script (`<script>`);
- Linha 28: foi criada uma função chamada “iniciarInteração”;
- Linha 29: inicialização de uma variável com escopo restrito chamada “nome”, a qual solicita que o usuário insira seu nome na caixa de diálogo que surge devido ao comando `window.prompt`;
- Linha 30: inicialização de uma variável com escopo restrito chamada “res”. O comando `getElementById()` retorna a referência do elemento ‘res’ através do seu Id;

- Linha 32: *res.innerHTML* foi um método utilizado para definir um novo conteúdo para o elemento 'res';

- Linha 34: fechamento do script (*</script>*).

Obs.: é importante ressaltar que as tags *<script>* e *</script>* poderiam ter sido inseridas tanto no *<body>* quanto no *<head>*, entretanto da maneira como foi feito (dentro do body) tem um melhor tempo de resposta no carregamento.

## **Bibliografia**

MAIA, Rômulo Francisco de Souza. Linguagens de script para web. 1ª edição. São Paulo: Editora Senac São Paulo, 13 de maio de 2021

FLANAGAN, David. JavaScript: O Guia Definitivo. 6ª edição. Porto Alegre: Bookman, 29 de agosto de 2012

Programação para Internet. Prof. Dr. Daniel A. Furtado - FACOM/UFU

CAMPOS, André. JavaScript: Start. 1ª edição. 31 de maio de 2020

TORRES, Dorian. Javascript Essencial: Guia completo de Javascript para iniciantes e profissionais. 29 de abril e 2022

DUARTE, Luiz. Programação Web com Node.js: Completo, do Front-end ao Back-end. 4ª edição. LuizTools, 10 de setembro de 2017

CAMPOS, André. JavaScript EVOLUTION. 1ª edição. 30 de setembro de 2020