



CENTRO UNIVERSITÁRIO SENAC – CAMPUS SANTO AMARO  
TADS – Análise e Desenvolvimento de Sistemas

# DOM X Javascript

**Nomes:** Ana Laura Campoy Ramos e Victória Oliveira França

**Professor:** Carlos Henrique Veríssimo

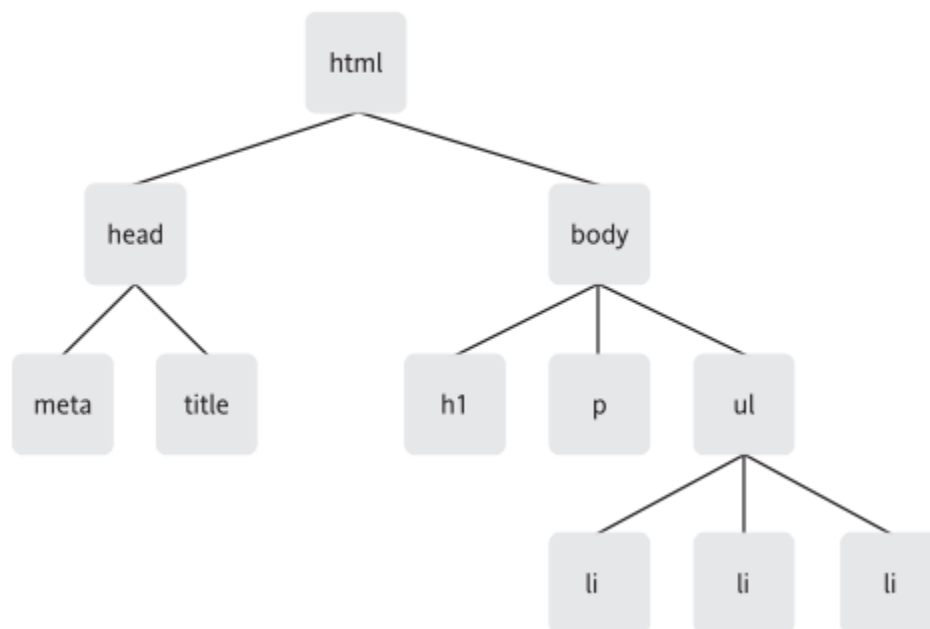
**Atividade Discente Orientada de Programação Web**

São Paulo, 30 de agosto de 2022.

# Document Object Model (DOM)

DOM (Document Object Model) é um modelo que, como todo modelo, representa algo e que nesse caso representa a página web carregada no navegador. O navegador provê um mapa (ou modelo) da página e você pode usar o JavaScript para ler esse mapa.

Uma característica muito importante do DOM é a sua representação, que é em forma de árvore. Ele apresenta os elementos HTML hierarquicamente, um abaixo do outro conforme seus “parentescos”, como numa árvore genealógica:



O elemento raiz é o “html”, isso porque ele não possui elementos irmãos ou pais, todos os elementos provêm dele, ou seja, são seus filhos. Um nível abaixo encontramos dois ramos, “head” e “body”, ambos são filhos de “html” porém também possuem filhos contidos em si.

É importante estabelecer e entender essas relações pois, muitas vezes, o JavaScript manipula e acessa elementos no DOM a partir delas.

O JavaScript necessita do DOM para conseguir acessar o documento HTML e assim, acessar, manipular, editar, criar e até excluir elementos deste documento. Esse acesso ocorre a partir de métodos do DOM, sendo alguns deles: `getElementById`, `getElementsByClass` e `getElementsByTagName`.

Com esses métodos o JavaScript pode identificar os elementos a partir de seus atributos específicos e assim, facilita e torna mais precisa a manipulação de informações do HTML. Com o JavaScript também é possível criar novos atributos para os elementos já existentes, isso é feito a partir do método `setAttribute`.

# Caso Prático

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Números com JS</title>
7   <style>
8     body { font: 12pt Arial; }
9     button { font-size: 12pt; padding: 30px; }
10  </style>
11 </head>
12 <body>
13   <h1>Senac - TADS - PW - 2º Semestre </h1>
14   <h2>Aula #02 - Introdução ao JS</h2>
15   <h3>Fazendo Cálculos</h3>
16   <button onclick="calculadora()">Clique para calcular</button>
17   <section id="res">
18     <p>Atenção: O resultado será colocado aqui...</p>
19   </section>
20
21   <script>
22     // O que fica aqui fora vai executar automaticamente, SEMPRE que o site for carregado
23     window.alert('Seja bem-vindo(a) ao meu site!')
24     // Já a função calcular() só vai executar quando o usuário pressionar o botão
25     function calculadora() {
26       let n1 = Number(window.prompt('Digite um número: '))
27       let res = document.querySelector('section#res')
28
29       res.innerHTML = `<p>O dobro de ${n1} é ${n1*2} e a metade é ${n1/2}!</p>`
30     }
31   </script>
32 </body>
33 </html>
```

A variável “res” no Javascript é criada para armazenar a *section* do **HTML** que foi selecionada a partir do seu *id*. Inicialmente, o usuário digitou um número que foi armazenado em outra variável, a *n1*. No último comando desse programa, o JavaScript altera o conteúdo da *section* “res” para que no parágrafo (*tag “p”*) contido nela, apareçam as seguintes informações: o número que foi digitado pelo usuário, o dobro desse número e a metade dele.

## **Bibliografia:**

DOM Scripting: Web Design with JavaScript and the Document Object Model, de **Jeremy Keith**