



TADS - Análise e Desenvolvimento de Sistemas

**Professor Carlos Verissimo**

# Programação Web

## DOM/JavaScript

**Hernandez Oliveira**

**Gustavo Aquino**

## Índice:

O que é DOM? .....	2
Relação entre DOM e JavaScript.....	3
Elementos da Implementação do JavaScript/DOM.....	3
Caso Prático.....	4
Bibliografia.....	5

## O que é DOM?

O DOM ou (Document Object Model) é uma coleção de objetos que fornece ao JavaScript um controle completo e total sobre a estrutura e o conteúdo das páginas WEB.

Ele é uma interface independente de plataforma e linguagem que permite aos programadores e scripts acessar e atualizar dinamicamente a estrutura, o conteúdo e a estilização de documentos.

O DOM subdivide-se em três subconjuntos: DOM Core (que significa o coração, designa as funções gerais de uma Interface de Programação), DOM HTML e DOM XML.

O DOM fornece aos programadores maneiras simples de acessar a estrutura, criar, modificar, adicionar, retirar e manipular elementos e conteúdos de documentos HTML e XML.

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

O DOM oferece uma visão amistosa do script na estrutura e no conteúdo de uma página Web, que será importante se você quiser usar o JavaScript para alterar dinamicamente uma página. Através da lente do DOM, uma página parece uma hierarquia de elementos na forma de uma árvore.

Cada folha na árvore é um nó, que se relaciona diretamente a cada elemento em uma página. Quando um nó aparece sob outro na árvore, ele é considerado um filho desse nó.

Então, enquanto o DOM representa e estrutura os componentes da página, o JavaScript acessa e manipula esses conteúdos.

## **Elementos da Implementação do JavaScript/DOM:**

Todos os elementos de páginas dentro de um documento herdam uma API e propriedades de Elementos. A maioria das funcionalidades tem a ver com a leitura e gravação dos atributos, ou a verificação da existência de atributos.

- `getAttribute(nome)`
- `setAttribute(nome, valor)`
- `removeAttribute(nome)`
- `getAttributeNode(nome)`
- `setAttributeNode(atributo)`
- `removeAttributeNode(atributo)`
- `hasAttribute(nome)`

À medida que mais páginas Web são projetadas usando CSS e construídas em camadas com elementos aninhados em outros, trabalhar com elementos em contexto é uma forma de manter algum nível de controle sobre quais componentes da página são afetados pela aplicação JavaScript. Isso é mais perceptível quando você usa essa abordagem para modificar o documento.

## Caso Prático:

### Cod07\_CalculoMedia.html

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Números com JS</title>
  <style>
    body {font: 12pt Arial; }
    button {font-size: 12pt; padding: 30px; }
  </style>
  <!-- Na linha 9 e 10 estamos fazendo a estrutura do botão para calcular a média--></head>

<body>
  <h1>Senac - TADS - PW - 2º Semestre </h1>
  <h2>Aula #02 - Introdução ao JS</h2>
  <h3>Cálculo de Média</h3>
  <button onclick="media()">Calcular média</button>
  <section id="situacao">
    <p>O resultado vai aparecer aqui... </p>
  </section>
  <!--Da linha 15 até a linha 17 estamos colocando o título/tema do site, linha 18 estamos recitando o botão em nosso programa que levará o nome de "Calcular Média"
  e as linhas 19 e 20 estamos colocando um id chamado situacao para mexermos nesse texto no CSS de uma forma mais simples-->

  <script>
    function media() {
      let nom = window.prompt('qual é o aluno?')
      let n1 = Number(window.prompt('Qual foi a primeira de ${nom}?'))
      let n2 = Number(window.prompt('Além de ${n1}, qual foi a outra nota de ${nom}?'))
      med = (n1 + n2)/2
      // Da linha 28 a 30 declaramos as variáveis que iremos utilizar para calcular a média dos alunos e para pegar as informações do aluno que iremos calcular a média
      // ...e na linha 31 mostramos a fórmula de como queremos que máquina calcule
      let res = document.getElementById('situacao')
      res.innerHTML = <p> calculando a média final de <mark>${nom}</mark>.</p>
      res.innerHTML += <p> As notas obtidas foram <mark>${n1}</mark> e <mark>${n2}</mark>.</p>
      res.innerHTML += <p> A média final será <mark>${med}</mark>.</p>
      // Linha 35 até a 37 estamos pedindo que a nossa máquina armazene os dados colocados pelo usuário e mostre as informações para o aluno ou para o responsável
    }
  </script>

</body>
</html>
```

Neste exemplo é utilizado os elementos do DOM, fazendo cálculo de média final de determinados alunos e apresentando o resultado na tela, assim o código depois de seus processos apresenta a média final para o usuário.

**Bibliografia:**

Aprendendo JavaScript (por Shelley Powers)

JavaScript – Guia do Programador (por Maurício Samy Silva)

O melhor do JavaScript (por Douglas Crockford)

JavaScript em Alto Desempenho (por Nicholas C. Zacks)

**GitHub:**

<https://github.com/ProfCarlosVerissimo/SENAC-PWD156-Prog-web>