



Centro Universitário SENAC – Campus St. Amaro (São Paulo)

TADS - Análise e Desenvolvimento de Sistemas

Document Object Model (DOM)

Autores: Matheus F. Santos e Lucas Ryu Muraoka

Professor: Carlos Henrique Veríssimo Pereira

21/08/2022

Índice

O que é DOM.....	2
Relação entre DOM e JavaScript.....	2
Elementos de implementação DOM/JavaScript.....	3
Caso Prático.....	4
Bibliografia.....	5

Desenvolvimento

O que é DOM

DOM (Document Object Model) é uma API criada com a finalidade de auxiliar o desenvolvimento de aplicação na natureza geral, ou seja, é uma API que padroniza a estrutura de documentos HTML e XML, simplificando a tarefa de se acessar e manipular tais documentos e fornece um mecanismo padronizado de acessar e modificar os dados da página Web.

Segundo a W3C (World Wide Web Consortium) o DOM é “Uma interface independente de plataforma e linguagem que permite aos programas e scripts acessar e atualizar dinamicamente a estrutura, o conteúdo e a estilização de documentos”.

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

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

- Adicionar ou remover elementos dentro do HTML
- Alterar os elementos HTML da página
- Alterar os atributos dos elementos HTML (como ‘src’ de uma imagem por exemplo)
- Alterar o CSS
- Reagir a eventos na página (clique, rolagem, input em formulários, passar o cursor sobre elementos, etc.)

Fazendo uma relação entre ambos utilizando alguns elementos para modificar uma página, para tornar o seu desenvolvimento mais produtivo. Veja alguns elementos no próximo tópico do trabalho

Desenvolvimento

Elementos de Implementação

No DOM existem alguns elementos que na maioria tem a funcionalidade de leitura e gravação dos atributos, ou a verificação da existência de atributos. Alguns elementos do DOM abaixo

- `getAttribute(nome);`
- `setAttribute (nome, valor);`
- `removeAttribute(nome);`
- `getAttributeNode(nome);`
- `setAttributeNode(atributo);`
- `removeAttributeNode(atributo);`
- `hasAttribute(nome);`
- `getElementById();`
- `getElementsByClassName();`
- `getElementsByTagName();`
- `querySelector();`
- `querySelectorAll();`
- `childNodes;`
- `firstChild;`
- `nodeName;`
- `nodeValue;`
- `nodeType.`

Essas são algumas dos Elementos do DOM e que são mais recomendadas pois podem causar menos confusão e são mais simples de serem entendidas possibilitando criar diversas APIs, veja um exemplo no próximo tópico, Caso Prático.

Caso Prático

Nesse exemplo abaixo é apresentado, na prática, como realmente usa-se os elementos DOM em um script, onde, nesse exemplo, faz o cálculo da média final de um determinado aluno e apresenta na tela, após todos os processos serem concluídos, a média final desse determinado aluno.

Script:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <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>
</head>
<body>
  <h1>Senac - TADS - PW - 2º Semestre </h1>
  <h2>Aula #02 - Introdução ao JS</h2>
  <h3>Cálculo da Média</h3>
  <button onclick="media()">Calcular média</button>
  <section id="situacao">
    <p>O resultado vai aparecer aqui...</p>
  </section>

  <script>

    function media() {
      let nom = window.prompt('Qual é o nome do aluno?')
      let n1 = Number(window.prompt(`Qual foi a primeira nota de ${nom}?`))
      let n2 = Number(window.prompt(`Além de ${n1}, qual foi a outra nota de ${nom}?`))
      med = (n1 + n2)/2

      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>`
    }
  </script>
</body>
</html>
```

Autor: Carlos Henrique Veríssimo Pereira.

Bibliografia

- JavaScript - Guia do Programador: Guia Completo das Funcionalidades de Linguagem JavaScript (por Maurício Samy Silva, 23 de setembro de 2010)
- Aprendendo JavaScript (por Shelley Powers)
- Use a cabeça JavaScript (por Michael Morrison)
- O melhor do JavaScript (por Douglas Crockford)
- JavaScript em Alto Desempenho (por Nicholas C. Zakas)
- <http://w3schools.com>
- <https://github.com/ProfCarlosVerissimo/SENAC-PWA107-Prog-Web/tree/main/PWA107-codigos/Aula2-codigos>