Programação WEB - #PWA107

Atividade Discente Orientada ADO #1 -

Document Object Model (DOM)

Aluno: Alex Sousa Pedroso

Aluno: Lucas Eufrásio Ferreira

Professor: Carlos Henrique Veríssimo Pereira

Conceitos

- **O que é o Document Object Model (DOM)?**

O chamado DOM (Document Object Model) é uma interface padronizada de programação que navegadores utilizam para representar páginas na web, permitindo que os navegadores e scripts possam manipular o conteúdo da página web sem precisar da realização de atualizações. O DOM também tem a característica de ser neutro em relação a linguagens de programação permitindo conectá -las a página a ser desenvolvida, apesar de a linguagem mais usada com DOM ainda seja o JavaScript.

O DOM fornece uma representação estruturada dos documentos em forma de árvore que guia o navegador com ramificações e definindo metodologias para a alteração da estrutura, conteúdo e estilização de documentos. O DOM é muito usado quando o desenvolvedor necessita criar uma interface de usuário avançada ou atualizar um website, pois através do DOM é possível criar aplicações que alteram dados da página sem a necessidade de ser feita uma atualização. Além de tornar possível criar aplicações em que até mesmo o usuário possa mudar o layout da página, também sem a necessidade de atualização.

A padronização do DOM foi feita pela organização de padronização tecnológica W3C (World Wide Web Consortium) sendo que a primeira versão chamada DOM nível 1 foi lançada no final dos anos 90, importante ressaltar que o trabalho da W3C para que as empresas da época desenvolvessem uma linguagem de script padronizada (JavaScript), que fez com que a compatibilidade entre os navegadores do mercado facilitasse o trabalho da W3C com o DOM nível 1.

- **Dissertar sobre a relação entre DOM e Javascript**

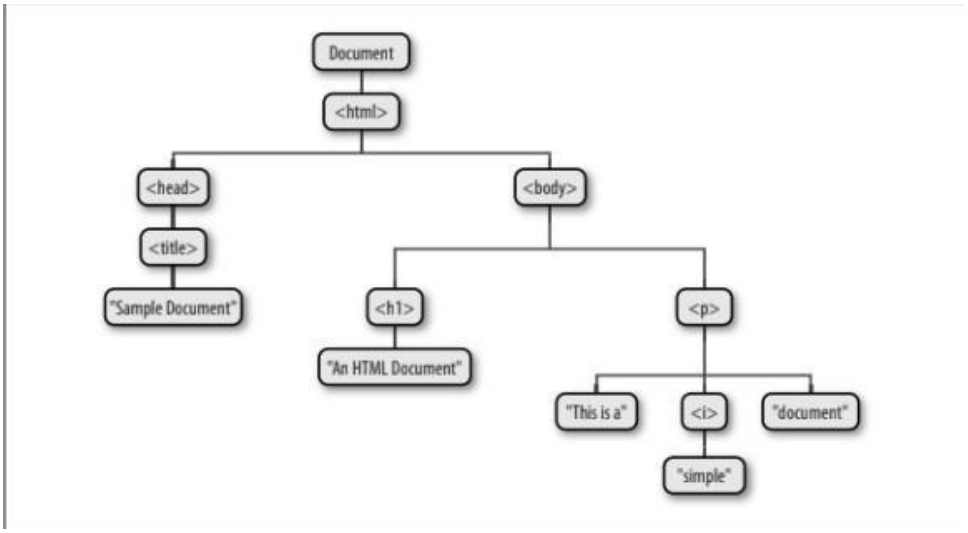
No início de vida do DOM ele a linguagem JavaScript eram entidades muito interligadas, mas apesar dos dois ainda serem bastante usados juntos eles com o passar do tempo se tornaram entidades bem diferentes, visto que o DOM é um modelo de representação de objetos, enquanto o JavaScript é uma linguagem de programação propriamente dita, ainda que a sua interface no navegador seja implementada em JavaScript. O conteúdo da página é acessado e manipulado pelo JavaScript enquanto o DOM tem a função de armazenar e definir a estrutura em que o documento construído pelo JavaScript será lido pelo navegador. Sem o DOM, a linguagem de programação usada não teria uma base para representar os seus objetos, ou seja, perderia a noção da relação da página web e dos seus componentes.

- **Elementos da implementação do JavaScript/DOM**

Os elementos de implementação do DOM são aninhados de um documento HTML ou XML que representam o DOM como uma árvore de objetos, essa representação da árvore contém marcações ou elementos HTML, como por exemplo <body> e <p>.

A estrutura em árvore da programação de um computador empresta a terminologia das árvores genealógicas, tudo que tiver abaixo são filhos desse nodo. Os que estão no mesmo nível e filhos do mesmo nodo são irmãos. O conjunto de nós a qualquer número de níveis abaixo de outro nodo são os descendentes desse nodo E o pai, avô e todos os outros nós acima de um nodo são os ascendentes desse nodo.

Segue abaixo um exemplo:



Logo a seguir teremos algumas propriedades usadas pelo DOM:

CreateElement : Cria um nodo elemento na página.

CreateAttribute : Cria um nodo atributo na página.

CreateTextNode : Cria um nodo texto na página.

ParentNode : Retorna o nodo pai de um nodo.

DocumentElement : Captura o elemento raiz <html> de um documento HTML.

GetElementById : Busca um elemento da página Web com o uso do atributo id do elemento.

- **Caso Prático**

Código 4 – Obtendo Dados

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dados 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>Tabalhando com Dados - PGM#4</h3>
    <button onclick="iniciarInteracao()">Clique para começar</button>
    <section id="resultado">
        <p>Observe este ponto: Aqui irá aparecer o resultado... </p>
    </section>
    <script>
        function iniciarInteracao() {
            let nome = window.prompt('Qual é o seu nome?')
            let res = window.document.getElementById('resultado')

            res.innerHTML = `<p>Olá, <strong>${nome}</strong>! Bem vindo ao Curso PW do Senac! &#x1F596;`
        }
    </script>
</body>
</html>
```

Neste exemplo o DOM começa com o getElementById que tem a função de buscar o elemento com o uso do atributo id, já o innerHTML puxa o texto juntamente com as tags diferentemente do innerText que somente puxa o texto, interessante pontuar que tanto como no innerText como no innerHTML você pode acrescentar texto sem precisar mexer na tag original.

- **Referência Bibliográfica**

FLANAGAN, David. **JavaScript**: O guia definitivo. 1.ed. Novatec .2010

ZACKAS, Nicholas C. **JavaScript em Alto Desempenho**. 6.ed. Bookman. 2013