



Programação web

DOM e JavaScript

Aluno: Guilherme de Oliveira Coelho

Professor: Carlos Henrique Verissimo Pereira

30/08/2022

Sumário

O que é DOM e relação entre ele e o JavaScript	página 1
Elementos de implementação do Dom	página 2
Métodos do Dom	página 3 e 4
Elementos de implementação do JavaScript	página 5
Interação do JavaScript com o HTML	página 6
Referências bibliográficas	página 7

O que é DOM

O Document Object Model (**DOM**) é uma interface de programação para os documentos HTML e XML. Representa a página de forma que os programas possam alterar a estrutura do documento, através dele temos métodos para acessar a árvore de elementos e assim alterar o estilo e conteúdo.

O DOM é uma representação orientada a objetos da página da web, que pode ser modificada com uma linguagem de script como JavaScript. Todas as propriedades, métodos e eventos disponíveis para manipular e criar páginas da Web são organizados em objetos.

Relação entre DOM e JAVASCRIPT

Todas as propriedades, métodos e eventos disponíveis para manipular e criar páginas da Web são organizados em objetos.

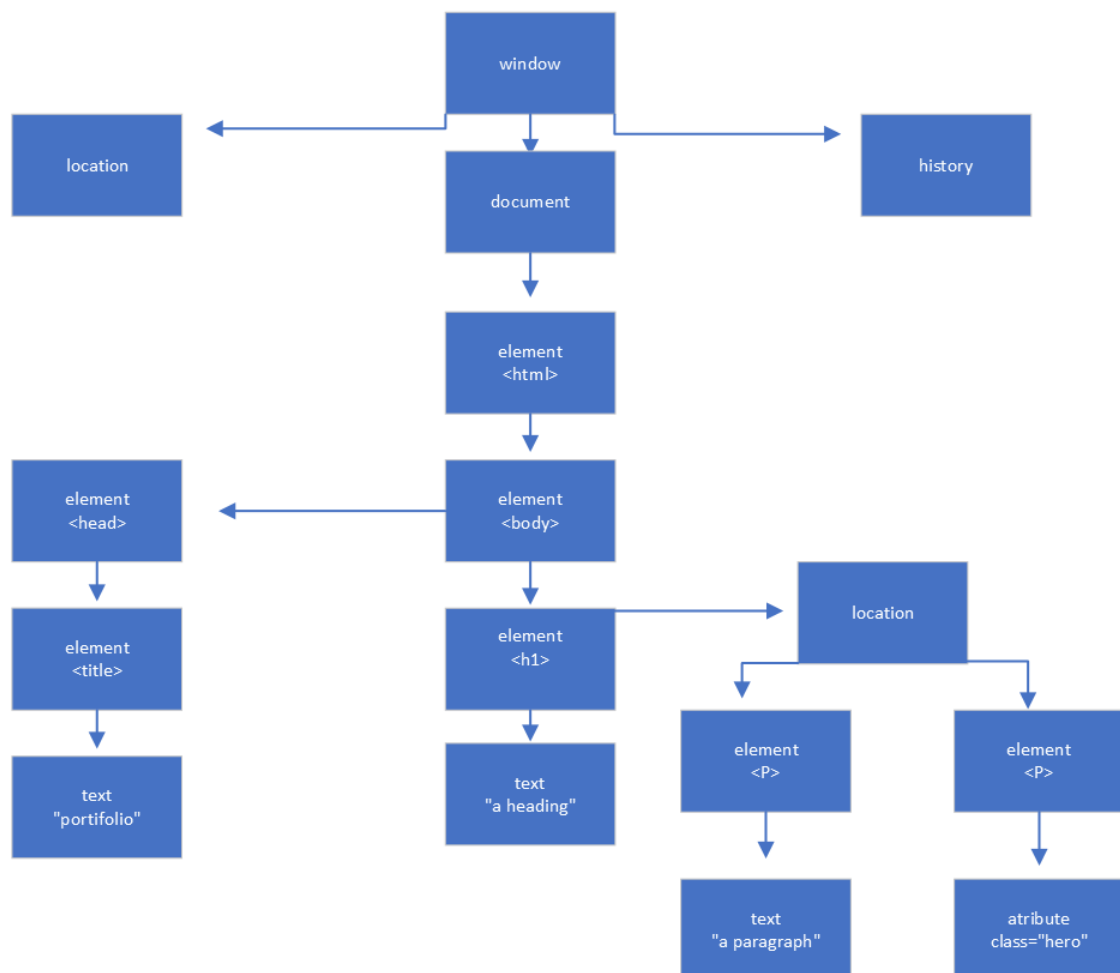
Cada elemento de um documento, o documento como um todo, o cabeçalho, as tabelas do documento, os cabeçalhos da tabela, o texto nas células da tabela faz parte do modelo de objeto do documento desse documento, para que todos possam ser acessados e manipulados usando o método DOM e uma linguagem de script como JavaScript, sem o dom a linguagem JavaScript não teria nenhum modelo ou noção de páginas da web.

O DOM não necessariamente está atrelado ao Javascript , pois ele foi projetado para ser independente de qualquer linguagem de programação específica, disponibilizando a representação estrutural do documento a partir de uma única API consistente.

Elementos de implementação do DOM

A estrutura que o DOM constrói a partir da leitura do seu documento HTML.

1. **Document:** Cuida de documentos HTML.
2. **Elements:** São todas as tags que estão em arquivos HTML ou XML se transformam em elementos da árvore DOM.
3. **Texts:** É o texto que vai entre os elementos. Todo o conteúdo das tags.
4. **Attributes:** É a junção de todos atributos para um nó específico.



Métodos

O DOM possui muitos métodos, são eles que fazem a ligação entre os nodes (elementos) e os eventos, os mais usados são:

getElementById()

Esse método retorna o elemento que estiver contendo o nome do ID passado.

```
var myStart = document.getElementById('start');
```

myStart: elemento específico que se equipara com o seletor passado.

start: seletor passado, caso não houvesse nenhum ele retornaria null.

getElementsByClassName()

Esse método retorna um HTMLCollection de todos elementos que estiverem contendo o nome da classe passada.

```
var myContainer = document.getElementsByClassName('container');
```

myContainer: elemento específico que se equipara com o seletor passado.

.container: seletor passado, caso não houvesse nenhum ele retornaria null.

getElementsByTagName()

Ele retorna uma HTMLCollection mas com uma diferença: esse método retorna todos elementos contendo a tag name passada.

```
var buttons = document.getElementsByTagName('button');
```

buttons: elemento específico que se equipara com o seletor passado.

button: tag name passada.

`querySelector()`

Retorna o primeiro elemento que se equipara ao seletor CSS passado. Lembrando que o seletor deve seguir a sintaxe CSS. Caso não tenha nenhum seletor, ele retornará `_null.` _

```
var resetButton = document.querySelector('#reset');
```

`resetButton`: primeiro elemento que se equipara com o seletor passado.

`querySelectorAll()`

Muito igual ao `querySelector()`, mas ele tem apenas uma diferença: retorna todos elementos que se equiparam ao seletor CSS passado. O seletor também deve seguir a sintaxe CSS, caso não haja nenhum ele retornará `null`.

```
var myButtons = document.querySelectorAll('#buttons');
```

`myButtons`: elementos que se equiparam com o seletor passado.

`#buttons`: seletor passado, como nesse caso não há nenhum seletor com esse nome, ele retorna `null`.

Elementos de implementação do Javascript

JavaScript é uma linguagem de script orientada a objetos, multiplataforma, tem uma biblioteca padrão de objetos, como: Array, Date e Math, e um conjunto de elementos que formam o núcleo da linguagem, tais como: operadores, estruturas de controle e declarações.

Array

O objeto Array do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

Criando um Array

```
var frutas = ['Maçã', 'Banana'];  
  
console.log(frutas.length);
```

Date

Cria uma instância JavaScript de Date que representa um único momento no tempo.

Construtor

```
new Date();  
new Date(valor);  
new Date(dataString);  
new Date(ano, mês, dia, hora, minuto, segundo, milissegundo);
```

Math

Math

Math é um objeto embutido que tem propriedades e métodos para constantes e funções matemáticas. Não é um objeto de função.

Interação do JavaScript com o HTML

No **código JavaScript** pode-se atribuir valores aos eventos como se fossem propriedades, desde que o valor atribuído seja um código a ser executado.

O código JavaScript permite tornar uma página mais dinâmica, respondendo a interações do usuário. A sintaxe, bastante parecida com outras linguagens e muito intuitiva, facilita o aprendizado e utilização.

A primeira função apresentada é a `alert`, que serve para exibir uma mensagem em uma janela popup. Essa função recebe apenas um parâmetro, que é o texto a ser exibido.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <script type="text/javascript">
      alert("Olá, seja bem vindo ao Linha de Código.")
    </script>
  </head>
  <body>
  </body>
</html>
```



Referências bibliográficas

Livro “JavaScript guia do programador” (Mauricio Samy Silva)

Livro “Aprendendo JavaScript” (Shelley Powers)

Livro Javascript: Básico ao Avançado

https://repositorio.usp.br/directbitstream/4cd7f9b7-7144-40f4-bfd0-7a1d9a6bd748/nd_72.pdf

