

Google



olabi

# <Interação com DOM>

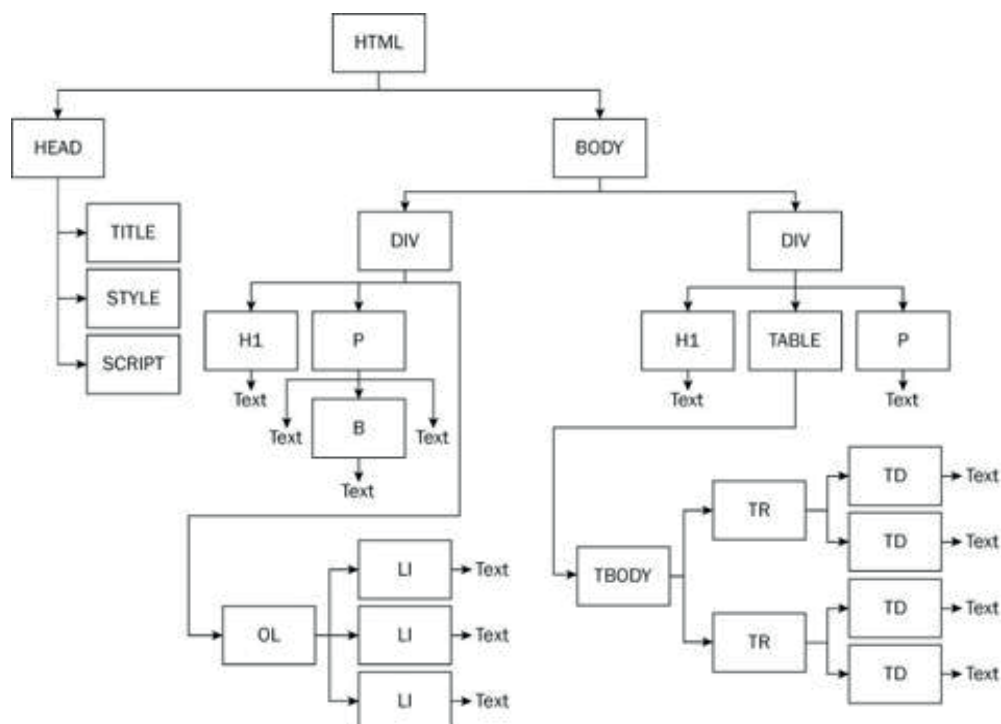


## { Interação com DOM

*Document Object Model* (DOM) é uma interface de programação para documentos HTML e XML. Ele representa a página que permite que programas alterem a estrutura do documento, estilo e conteúdo. Como o DOM representa o documento com nós e objetos, as linguagens de programação se conectam à página web de maneira que, ao carregá-la, um DOM dela é criado pelo navegador. Apesar de não ser uma linguagem de programação, sem DOM, a linguagem JavaScript não teria modelo ou noção de páginas web, documentos HTML, XML ou seus componentes (elementos, por exemplo).

Cada elemento de um documento (seu cabeçalho, tabelas, os cabeçalhos da tabela e texto em suas células) faz parte do modelo de objeto deste documento, de forma a permitir que todos sejam acessados e manipulados usando DOM e uma linguagem de *script*, como JavaScript.

O modelo HTML DOM é construído como uma árvore de objetos, conforme exemplo abaixo.



Com o modelo de objeto, o JavaScript obtém toda a energia necessária para criar um HTML dinâmico que possa:

- alterar todos os elementos HTML na página;
- alterar todos os atributos HTML na página;
- alterar todos os estilos CSS na página;
- remover elementos e atributos HTML pré-existent;
- adicionar novos elementos e atributos HTML;
- reagir a todos os eventos HTML existentes na página;
- criar eventos HTML na página.

Com o *object-document*, conseguimos acessar diversas propriedades. Dentre as mais utilizadas:

PROPRIEDADE	DESCRIÇÃO
<code>documentElement</code>	Captura o elemento raiz <code>&lt;html&gt;</code> de um documento HTML
<code>getElementById</code>	Busca um elemento da página web usando o atributo <code>id</code> do elemento
<code>createElement</code>	Cria um elemento nodo na página
<code>createAttribute</code>	Cria um atributo nodo na página
<code>createTextNode</code>	Cria um texto nodo na página
<code>getElementsByTagName</code>	Retorna array de elementos com o mesmo nome
<code>appendChild</code>	Insere um novo elemento-filho
<code>removeChild</code>	Remove um elemento-filho
<code>parentNode</code>	Retorna o nodo-pai de um nodo

## { Percorrendo árvores DOM

A árvore DOM pode ser percorrida por meio da variável *document* no código Javascript. Como cada nó tem um pai (e a maioria, um nó filho), é possível percorrer essa árvore, subindo e descendo na árvore DOM, usando essas conexões. Como os nós são agrupados pelo DOM em um objeto *node*, a partir desse *node*, é possível capturar seu nome (*nodeName*) e valor (*nodeValue*).

```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```

Pelo HTML acima, sabemos que:

- <html> é o nó raiz;
- <html> não tem pais;
- <html> é o pai de e <head><body> ;
- <head> é o primeiro filho de <html> ;
- <body> é o último filho de <html> ;
- <head> tem um filho <title>;
- <title> tem um filho (um nó de texto) "TUTORIAL DOM";
- <body> tem dois filhos e <h1><p>;
- <h1> tem um filho "DOM Lição um";
- <p> tem um filho "Olá mundo!";
- <h1> e <p> são irmãos <p>.

## <Exercícios>

- ➡1 Crie um programa que capture o evento de um *input* na tela e mostre automaticamente a tabuada desse número abaixo.
- ➡2 Crie um programa que “pegue” uma lista `<ol>` no HTML e coloque negrito dinamicamente, a partir de um nome digitado no seu *input*.  
Ex.:

A tela possui uma lista de nomes:

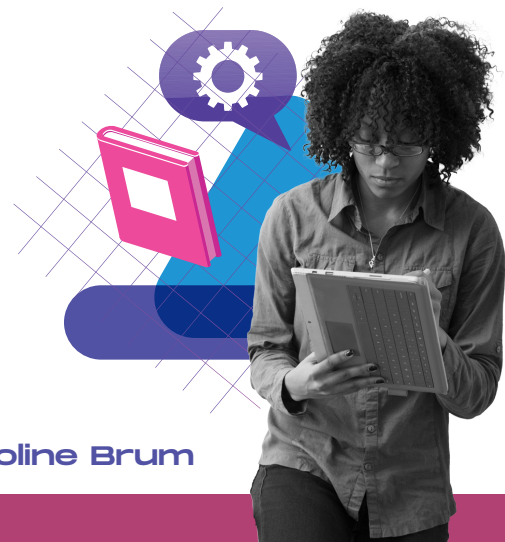
João Claudio Marcio
---------------------------

Ao utilizar um *input* e digitar “Jo”, o programa devolverá o seguinte resultado imediatamente:

<b>João</b> Claudio Marcio
----------------------------------

Nestes dois exercícios:

- Utilize um arquivo HTML e *tag script*;
- Lembre-se de que o objetivo é manipular o DOM, então dê o seu melhor para criar todas as operações com a árvore (adição, remoção, busca etc.)



## <Exercícios>

- ➡3 Crie um objeto-pinguim com nome, idade e origem. Em seguida, crie métodos para o pinguim andar e voar - para voar, ele deverá ter uma propriedade “podeVoar” verdadeira. Utilize o HTML abaixo como referência para iniciar.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1
6     .0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Pinguim</title>
9 </head>
10 <body>
11   <h1>Pinguim</h1>
12   <script src="js/script.js"></script>
13 </body>
14 </html>
```

## <Referências>

O que é DOM (Document Object Model)? Trabalhando com DOM em JavaScript ([devmedia.com.br](http://devmedia.com.br))

JavaScript HTML DOM ([w3schools.com](http://w3schools.com))

Introdução ao DOM - APIs da Web | MDN ([mozilla.org](http://mozilla.org))