**DOM(Document Object Model)**

DOM é a forma como os elementos HTML estão **estruturados** na página, que permite com que **programas** e **scripts** possam **acessar** e **alterar** tais elementos dinamicamente.

Quando uma página é carregada, o browser cria um DOM da página HTML.

O DOM para HTML define:

**DOM Tree**

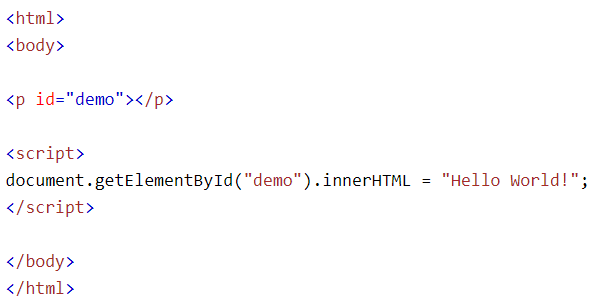
* Elementos HTML como **objetos**;
* **Atributos/propriedades** dos elementos HTML;
* **Métodos** para acessar esses elementos;
* **Eventos** dos elementos.

**DOM Tree:** é um mapa que mostra todos os elementos(objetos) pai e filho, atributos e conteúdo.

**Resumido:** DOM HTML é uma forma de acessar, modificar, adicionar e deletar elementos HTML.

Se todo elemento HTML é considerado como um objeto, então, assim como todo objeto em **POO**(Programação Orientada para Objetos), todo elemento HTML tem:

* **Atributos:** valores que podem ser modificados
* **Métodos:** ações que podemos realizar nesses objetos

Ex.:

Observe que ***getElementById()*** é um método do objeto documento, enquanto ***innerHTML*** é um atributo do objeto com id = ‘demo’

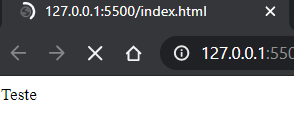
**O objeto document**

É o principal objeto da página e possui todos os outros elementos HTML.

**Principais métodos de document**

* **Document.write(‘texto’)** => escreve alguma coisa dentro do elemento <body>

Ex.:



**Nota:** se usado depois que o documento HTML for carregado, vai substituir todos os elementos HTML

* **Métodos de acessar elementos (será visto abaixo);**
* **Métodos de criar e deletar elementos(também será visto abaixo);**

**Principais Propriedades de document**

* **Document.head =>** retorna o elemento <head>
* **Document.body =>** retorna o elemento <body>
* **Document.title =>** retorna o elemento <title>
* **Document.URL** => retorna a URL completa da página
* **ETC...**

**Para ver mais métodos e propriedades, acesse** [**Object Document**](https://www.w3schools.com/js/js_htmldom_document.asp)

**Como acesso os elementos HTML?**

Todo elemento da página pertence ao objeto ***document***, então primeiro é preciso acessá-lo antes do elemento que queremos!

Existem 4 formas principais de armazenar um elemento HTML dentro de uma variável:

* Pelo **ID**;
* Pelo nome da **Tag**;
* Pela **Classe**;
* Pelo **Seletor CSS**.

**document.getElementById(nome do id):** pega o elemento pelo seu ID.

Ex.:

**document.getElementsByTagName(‘nome da tag’):** pega **todos os elementos** de uma mesma tag.

**Ex.:**

**Nota:** Vai criar uma lista com todos os elementos da mesma tag.

**document.getElementsByClasseName(‘nome da classe’):** pega **todos os elementos** de uma mesma classe.

**Ex.:**

**Nota:** Vai criar uma lista com todos os elementos da mesma classe.

**Pegando elemento por Seletor CSS**

Chama-se seletor CS porque tem a mesma sintaxe dos Seletores CSS. Existem duas formas:

* **document.querrySelector(‘nome da classe ou id’):** pega o **primeiro elemento** que tiver o ID ou Classe especificado.

Ex.:





* **document.querrySelectorAll(‘nome da classe ou id’):** pega todos os elementos que tiverem a mesma classe ou id.

**Ex.**:



**Nota:** retorna uma lista do tipo Nodelist.

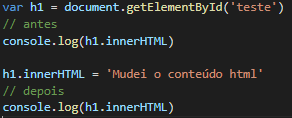
**Como modifico o elemento que eu peguei?**

É possível alterar tanto HTML do elemento e o CSS.

**Alterando o HTML:** usamos os atributos. Os principais são:

* **.innerHTML:** modifica o conteúdo do elemento HTML;

**Ex.**:

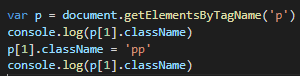


* **.nome do atributo:** é possível modificar o atributo de um elemento html;

**Ex.**: mudei o nome da classe(atributo className) do segundo elemento <p> que está dentro da variável p

**classeName original = nenhum**

**classeName depois = ‘pp’**



**Como crio elementos¿**

[**criar, deletar e substituir elementos HTML**](https://www.w3schools.com/js/js_htmldom_document.asp)

**Como altero o CSS de um elemento?**

Para isso, precisamos acessar, dentro do atributo style, a propriedade que queremos.

**Ex.:**

**Para saber o nome de todos os atributos CSS DOM, acesse** [**CSS atributes**](https://www.w3schools.com/jsref/dom_obj_style.asp)