



PROGRAMAÇÃO FRONT-END

GIOVANNI RIBEIRO



DOM



Podemos dizer que o DOM (Document Object Model) representa a forma de organização de elementos e marcações HTML, bem como se dá a leitura pelos navegadores.

O Modelo de Objeto de Documento (DOM) pode ser considerada como uma interface de programação para arquivos HTML.

Sua organização se dá em um formato de árvore, estruturado em nós e objetos. Os métodos disponíveis permitem acessar e manipular esses elementos por meio de propriedades e métodos. Ou seja, podemos manipular os elementos HTML e a visualização da janela do navegador.

Lembre-se de que uma página web é um arquivo HTML, manipulado e visualizado em navegadores.

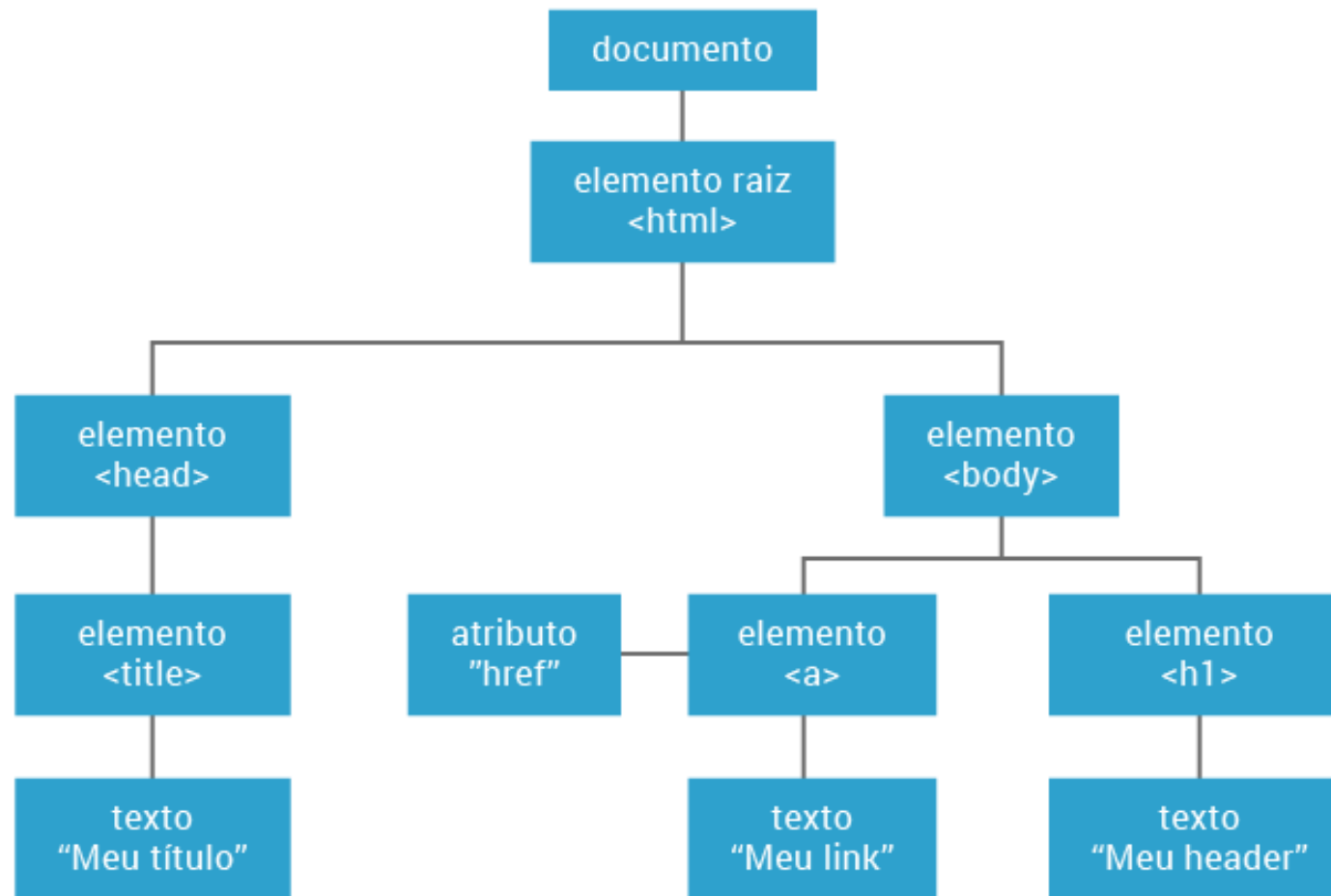
Objeto Document



Através do objeto document pode ter acesso a um grande número de propriedades. Segue abaixo algumas propriedades que podem ser utilizadas com o objeto document:

Propriedade	Descrição
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.
createElement	Cria um nodo elemento na página.
createAttribute	Cria um nodo atributo na página.
createTextNode	Cria um nodo texto na página.
getElementsByTagName	Retorna um array dos elementos com o mesmo nome.
appendChild	Insere um novo elemento filho.
removeChild	Remove um elemento filho.
parentNode	Retorna o nodo pai de um nodo.

A árvore de objetos – HTML DOM



DOM



Defina uma cor de fundo para um documento:

```
document.body.style.backgroundColor = "orange";
```

Defina uma cor de fundo de um elemento **<div>** específico:

```
<div id="minhaDiv">
  <h1>Olá</h1>
</div>
```

```
<style>
  #minhaDiv {
    width: 300px;
    height: 300px;
    background-color: #fff;
    color: #000000;
  }
</style>
```

```
function minhaFuncao() {
  document.getElementById("minhaDiv").style.backgroundColor = "yellow";
}
```

DOM



O método `querySelector()` retorna o primeiro elemento que corresponde a um seletor no documento.

```
<p class="texto">lorem ipsun</p>
<p class="texto">lorem ipsun</p>

<button onclick="minhaFuncao()">aplicar</button>

<script>
  function minhaFuncao() {
    var titulo = document.querySelector(".texto");
    titulo.style.backgroundColor = "red";
  }
</script>
```

DOM



O método `querySelectorAll()` retorna todos os elementos no documento que correspondem aos seletores CSS especificados.

```
<p class="texto">lorem ipsun</p>
<p class="texto">lorem ipsun</p>

<button onclick="minhaFuncao()">aplicar</button>

<script>
  function minhaFuncao() {
    var titulo = document.querySelectorAll(".texto");
    titulo[1].style.backgroundColor = "red";
  }
</script>
```

DOM



O método `createAttribute ()` cria um atributo com o nome específico e retorna o atributo.

```
<style>
  .exemploClass {
    color: ■ red;
  }
</style>
</head>
<body>
  <h1>Olá Mundo!</h1>
  <button onclick="minhaFuncao()">aplicar</button>

  <script>
    function minhaFuncao() {
      var h1 = document.getElementsByTagName("h1")[0];
      var atributo = document.createAttribute("class");
      atributo.value = "exemploClass";
      h1.setAttributeNode(atributo);
    }
  </script>
```


DOM



O método **appendChild()** insere um novo nó na estrutura do DOM de um documento.

A função **appendChild()** insere um elemento filho (children) ao elemento pai (parent) na última posição, ela auxilia na criação de um elemento **DOM**, veja o esquema da função:

```
var elemento_pai = document.body;
var titulo = document.createElement('h1');

// Criar o nó de texto
var texto = document.createTextNode("Título principal");

// Anexar o nó de texto ao elemento h1
titulo.appendChild(texto);

elemento_pai.appendChild(titulo);
```

← → ↻ ⓘ 127.0.0.1:5500/JAVASCRIPT/dom/appendChild.html

Título principal

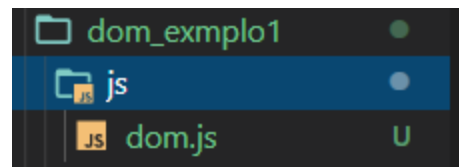
DOM



Exemplo Prático - HTML

```
<form name="meuForm">
  <input type="text" name="texto" />
  <input type="button" value="Inserir" onclick="incluir()" >
</form>
<ul id="minhaLista">
</ul>

<script src="js/dom.js"></script>
```



DOM

Exemplo Prático - Javascript

```
function incluir() {  
    var texto = document.meuForm.texto.value;  
    var lista = document.getElementById("minhaLista");  
    var novoItem = document.createElement("li");  
    var novoTexto = document.createTextNode(texto);  
    lista.appendChild(novoItem);  
    novoItem.appendChild(novoTexto);  
}
```