

## Introdução ao DOM

O Document Object Model (DOM) é uma interface de programação para os documentos HTML e XML, estruturando a página web de maneira que os sistemas possam alterar a estrutura do documento(HTML), alterar o estilo(CSS) e conteúdo(dados). O DOM representa o documento com nós e objetos, viabilizando as conexões pelas linguagens de programação.<sup>1</sup>

Vale salientar que o DOM é uma representação orientada a objetos de uma página da web, que pode ser modificada com uma linguagem de script como JavaScript.

Para normalizar a tecnologia os padrões W3C DOM<sup>2</sup> e WHATWG DOM<sup>3</sup> são implementados na maioria dos navegadores, mas há possibilidade de modificação destes padrões, gerando a necessidade de cuidados, pois navegadores com DOMs diferentes entregarão resultados diferentes na execução de um sistema.

O DOM foi projetado para ser independente de qualquer linguagem de programação específica, disponibilizando a representação estrutural do documento a partir de uma única API consistente. Vale ressaltar que o DOM pode ser acessado por outras linguagens além do JavaScript.

Para usar o DOM basta criar um código, neste contexto, JavaScript como pode-se observar no exemplo a seguir:

```
<!DOCTYPE html>
<html>
<body>
  <p id="ver1"></p>
  <p id="ver2"></p>
  <script>
    let url1 = window.document.URL;
    let url2 = document.URL;
    document.getElementById("ver1").innerHTML = url1;
    document.getElementById("ver2").innerHTML = url2;
  </script>
</body>
</html>
```

```
https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_doc_url1
https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_doc_url1
```

<sup>1</sup>Mozilla Foundation.. © 2022. [https://developer.mozilla.org/pt-BR/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction). Acesso em 20 de jul. de 2022.

<sup>2</sup> <https://dom.spec.whatwg.org/>. Acesso em 20 de jul. de 2022.

<sup>3</sup> <https://dom.spec.whatwg.org/>. Acesso em 20 de jul. de 2022.

Utilizando `window.document` ou `document` consegue-se acessar a estrutura DOM e realizar as manipulações permitidas pela tecnologia.

Se houver a necessidade de inserir dados no documento web, no caso abordado .html, o DOM associado a linguagem JavaScript permite a inserção de diversas maneiras, uma delas através do método/função `addEventListener` do `document`. A função/método `addEventListener` permite o acionamento de uma função através de um evento.

Alguns eventos JavaScript são: *mousedown*, *mouseup*, *click*, *dblclick*, *mouseover*, *mouseout*, *keydown*, *keypress*, *keyup*, *focus*, *blur*, *change*, *submit*, *scroll*, *resize*, *load* e *unload*.<sup>4</sup>

No exemplo abaixo o evento `click` aciona a função:

```
<!DOCTYPE html>
<html>
<body>
<p>Clique para aparecer a mensagem</p>

<p id="ver">

<script>
  document.addEventListener("click", mensagem);

  function mensagem() {
    document.getElementById("ver").innerHTML = "Aula de WEB I";
  }
</script>

</body>
</html>
```

Clique para aparecer a mensagem

Aula de WEB I

Ao clicar em qualquer lugar do documento, inclusive sobre a frase “Clique para aparecer a mensagem”, o conteúdo “Aula de WEB I” é inserido na tag `<p>` com `id` igual a “ver”.

Existe uma estrutura ou árvore ou tree DOM que pode ser parcialmente compreendida quando utilizamos um código JavaScript para identificar os nós DOM:

<sup>4</sup> <https://developer.mozilla.org/en-US/docs/Web/Events>. Acessado em 20.08.2022.

```
<!DOCTYPE html>
<html>
<body>
  <h1>Aula</h1>
  <h2>Web I</h2>
  <p>DOM e JS</p>
  <p id="demo"></p>
<script>
const lista = document.body.childNodes;

let texto="" ;
lista.forEach(
  function(nos, indice) {
    texto += indice + " " + nos + "<br>";
  }
);

document.getElementById("demo").innerHTML = texto;
</script>

</body>
</html>
```

## Aula

### Web I

DOM e JS

```
0 [object Text]
1 [object HTMLHeadingElement]
2 [object Text]
3 [object HTMLHeadingElement]
4 [object Text]
5 [object HTMLParagraphElement]
6 [object Text]
7 [object HTMLParagraphElement]
8 [object Text]
9 [object HTMLScriptElement]
```

No código acima observa-se que os objetos DOM são identificados pelo algoritmo JavaScript, permitindo a ampliação da compreensão inerente ao comportamento do DOM na renderização do documento.

Entender o DOM é importante para construção de um layout eficiente, consideradas as variedades de telas e dispositivos ofertados no mercado atual.

Ainda vale salientar que é possível alterar os valores das propriedades utilizando, HTML, CSS, JS e DOM, conforme exemplifica-se:

```
<!DOCTYPE html>
<html>
<body>

  <p>Mudar a cor do text na classe aula</p>
  <h2 class="aula">É importante praticar</h2>
  <p class="corpo">Leia o título</p>
  <p class="aula">Mas isso também é importante</p>
<script>
const listaDeNos = document.querySelectorAll(".aula");
for (let i = 0; i < listaDeNos.length; i++) {
  listaDeNos[i].style.color = "red";
}
</script>

</body>
</html>
```

Mudar a cor do text na classe aula

**É importante praticar**

Leia o título

Mas isso também é importante