UNINOVE

DOM

INTRODUÇÃO E MANIPULAÇÃO DE DOCUMENT OBJECT MODEL (DOM)

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

DOM

Podemos dizer que o DOM (Document Object Model) representa a forma de organização de elementos e marcações HTML e XML, 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 e XML.

Sua organização se dá em um formato de arvore, 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. O DOM neste caso representa como podemos manipular este documento, e disponibilizar propriedades, métodos e eventos.

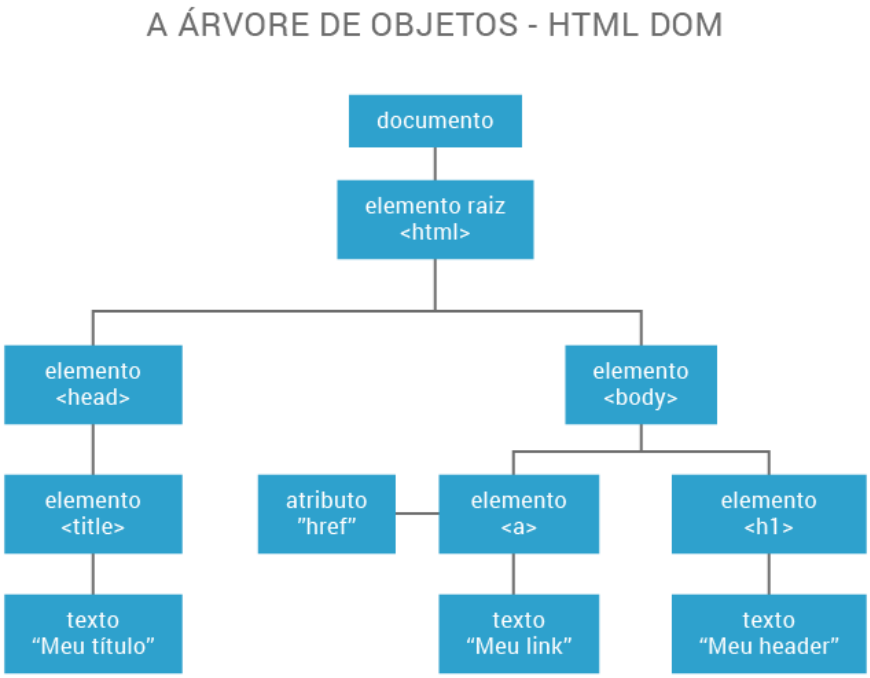
Na prática, utilizamos o JavaScript para acessar os elementos HTML com auxílio do DOM, ou seja, o DOM não é uma linguagem de programação, no entanto, seu desenho permite que seja utilizado por outras linguagens de programação.

Segue um exemplo simples:

```
1. <body onload="window.alert("Olá mundo");" >
```

Neste caso, o alert( ) é uma função do objeto window.

Veja esta estrutura sobre a arvore de objetos HTML DOM:



Nesta imagem podemos observar o elemento h1 descende de body, que descende de html. No outro nó, temos a propriedade text, determinada como 'My header'. A mesma lógica vale para os demais elementos HTML DOM.

Elemento

Podemos com o uso do JavaScript manipular e ler elementos do HTML.

Na sequência iremos, primeiramente, apresentar algumas formas de acessar os elementos HTML

HTML por id

Os elementos html possuem um id, e podemos utiliza-lo para encontrar este dado.

```
1. <script type="text/javascript">
2.     var nome = document.getElementById("idNome");
3. </script>
```

HTML por Tag Name (etiqueta)

Utilize as tag para encontrar os elementos.

```
1. <script type="text/javascript">
2.     var valor = document.getElementsByTagName("h1");
3. </script>
```

HTML por nome de classe

Encontrar os elementos pertencentes a uma classe.

```
1. <script type="text/javascript">
2.     var a = document.getElementsByClassName("form");
3. </script>
```

HTML por Seletores CSS

Encontrar os seletores css.

```
1. <script type="text/javascript">
2.     var a = document.querySelectorAll("div.form-group");
3. </script>
```

Podemos alterar um conteúdo HTML com uso da propriedade innerHTML.

Vamos analisar alguns exemplos:

InnerHTML

Nesta propriedade é possível alterar o conteúdo de um elemento HTML.

```
1. <script type="text/javascript">
2.     document.getElementById("idCPF").innerHTML = novoCPF;
3. </script>
```

Podemos unir as duas técnicas, veja neste script:

```
1. <script>
2. var nome = document.getElementById("idNome");
3. nome.innerHTML = "Pedro Alvares";
4. </script>
```

No exemplo acima, capturamos o elemento de id “idNome” e o adicionamos em uma variável, na sequência alteramos a propriedade innerHTML.

Podemos ainda alterar algumas propriedades de elementos, como as imagens. Veja no exemplo a seguir:

```
1. <!DOCTYPE html>
2. <html>
3.     <body>
4.         <img src="semfoto.jpg" id="idProcurar">
5.         <script>
6.             document.getElementById("idProcurar").src = "pergunta.jpg";
7.         </script>
8.     </body>
9. </html>
```

No exemplo acima, o script altera a propriedade src, que indica o caminho e nome do arquivo de imagem a ser exibido.

Estilo HTML

Podemos alterar os estilo dos elementos HTML, inclusive em conjunto com outros elementos. Veja o exemplo a seguir:

```
1. <!DOCTYPE html>
2. <html>
3.     <body>
4.         <p id="idIMG">Clique na imagem</p>
5.         <img src="semfoto.jpg" id="idProcurar" >
6.         <script>
7.             document.getElementById("idProcurar").src = "pergunta.jpg";
8.             document.getElementById("idIMG").style.color = "red";
9.         </script>
10.     </body>
11. </html>
```

A linha 8 altera a cor do elemento id="idIMG" para a cor vermelha.

Obviamente podemos utilizar este exemplo não apenas nos scripts, mais também associados a eventos, como os utilizados nos eventos onclick.

```
1. <!DOCTYPE html>
2. <html>
3.     <body>
4.         <p id="idIMG">Clique na imagem</p>
5.         <button type="button" onclick="document.getElementById('idIMG').st
6.         <img src="semfoto.png" id="idProcurar" >
7.
8.         <script>
9.             document.getElementById("idProcurar").src = "pergunta.jpg";
10.            document.getElementById("idIMG").style.color = "red";
11.        </script>
12.    </body>
13. </html>
```

Vamos analisar mais um exemplo de alteração de css por meio de evento com DOM HTML.

Neste código simples iremos alterar a largura das bordas de uma determinada imagem. Observe que temos os eventos de elevar e retornar ao tamanho original. Você pode utilizar qualquer imagem no seu exemplo. Vamos ver o código:

```
1. <!DOCTYPE html>
2. <html>
3.     <body>
4.         <p>
5.             <img id="exemploImg"
6.                  src="pergunta.jpg"
7.                  style="border: 5px solid gold;"
8.                  alt="border_Img">
9.         </p>
10.        <form name="Form_01">
11.            <input type="button" value="Alterar borda" onclick="alterarBo
12.            <input type="button" value="Retornar borda"  onclick="alterarB
13.        </form>
14.        <script>
15.            function alterarBordas(medida) {
16.                document.getElementById("exemploImg").style.borderWidth =
17.            }
18.        </script>
19.    </body>
20. </html>
```

A imagem a seguir exibe o código html sem alteração do css, com as bordas em dourado na imagem. Veja abaixo:

Na imagem seguinte, temos o css alterado, para uma margem maior, determinado no evento associado ao botão. Veja como ficou:

Alterar borda Retornar borda

DICA:

Para saber mais, veja em:
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction (https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Examples (https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Examples)
https://www.w3schools.com/js/js_htmldom.asp (https://www.w3schools.com/js/js_htmldom.asp)

Conclusão

O uso de DOM como recurso para manipulação dos elementos HTML é um recurso importante e muito útil, principalmente quando houver necessidades especificas e de alteração na composição HTML. A combinação de possibilidades é extensa e, portanto, necessita de prática.

ATIVIDADE

Escolha a alternativa que indique o comando adequado para capturar o valor, utilizando DOM HTML, por meio da propriedade id.

A. var nome = document.getElementBy("idNome").id;
B. var nome = document.getElementById("idNome");
C. var nome = document.getElementBy.Id("idNome");
D. var nome = document.getElementBy_Id("idNome");

ATIVIDADE

Escolha a alternativa que indique o comando adequado para capturar o valor, utilizando DOM HTML, por meio da propriedade TagName.

A. var valor = document.getElementsByTagName("h1");

B. var valor = document.getElementsByTag("h1");

C. var valor = document.getElementsByName("h1").Tag;

D. var valor = document.getElementsByTag("h1").Name;

ATIVIDADE

De acordo com o código HTML apresentado a seguir, escolha a alternativa que indique qual comando altera, via DOM HTML, o tamanho da margem em torno da imagem "pergunta.jpg".

```
<!DOCTYPE html>
<html>
<body>
<p>
<img id="exemploImg"
src="pergunta.jpg"
style="border: 5px solid gold;"
alt="border_Img">
</p>
<form name="Form_01">
<input type="button" value="Alterar borda" onclick="alterarBordas(20);"
/>
<input type="button" value="Retornar borda"
onclick="alterarBordas(5);" />
```

```
</form>
<script>
function alterarBordas(medida) {
document.getElementById("exemploImg").style.borderWidth = medida
+ "px";
}
</script>
</body>
</html>
```

A. style="border: 5px solid gold;"

B. <input type="button" value="Retornar borda" onclick="alterarBordas();" />

C. document.getElementById("exemploImg").style.borderWidth = medida + "px";

D. alt="border_Img">

REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.