

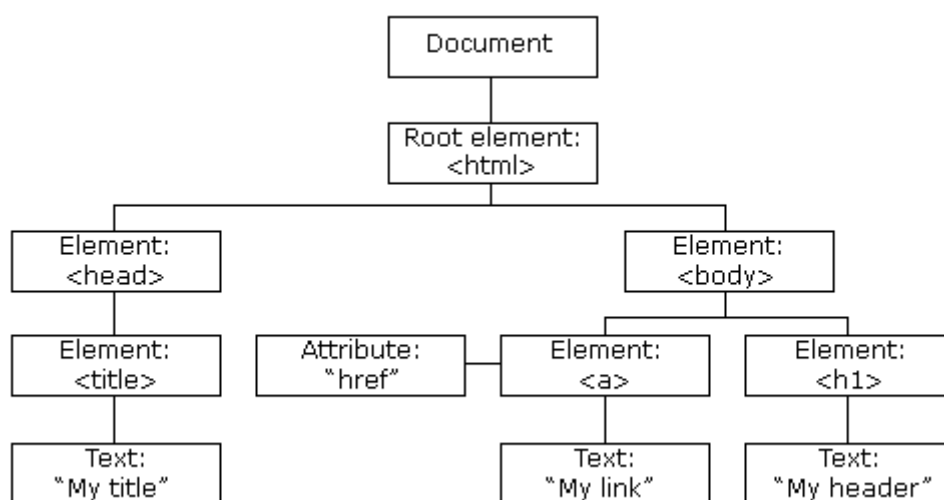
## Ficha de trabalho nº8 – JavaScript – DOM

Quando uma página é carregada o navegador cria um DOM (Document Object Model) da página.  
O HTML DOM é construído como sendo uma árvore de objetos.

Nesta ficha de trabalho vamos utilizar JavaScript DOM. Realçar que a explicação teórica e prática está disponível em vídeo em: <https://youtu.be/Kj-GSGi3ibU> e <https://youtu.be/j0Vm5tcdFvM>. A fonte de informação está disponível em: [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)

### Parte I – O que é o DOM

O DOM (Document Object Model) é o modelo criado automaticamente pelos navegadores como representação de uma página HTML carregada.



O DOM é um padrão na representação de páginas web definido pela W3C:

*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

Existem 3 standards específicos para aplicar ao DOM:

- Core DOM – aplicado a todos os documentos;
- XML DOM – aplicado a documentos XML;
- HTML DOM – aplicado a documentos HTML.

Simplificando, o DOM é a forma como podemos selecionar, adicionar, editar ou remover elementos de uma página HTML.

## Parte II – Métodos

Cada elemento de um DOM é definido como um objeto sendo que cada objeto possui propriedades e métodos associados.

- Propriedade – valor que pode ser lido ou escrito (ex.: conteúdo textual de um elemento);
- Método – ação que pode ser aplicada (ex.: adicionar ou eliminar um código).

Utilizando o exemplo seguinte:

```
<html>
<body>

<p id="demo"></p>

<script>
    document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

Ao documento (document) foi aplicado o método getElementById e a propriedade innerHTML para que ao elemento com o ID "demo" fosse inserido o código HTML "Hello World".

De notar que em nenhuma parte do HTML há referência a conteúdo do parágrafo, mas executando a página o mesmo está preenchido.

### Parte III – Document e Element

Há três classes principais para navegação e leitura de elementos numa página HTML usando DOM:

- `document.getElementById(id)` – seleciona um elemento pelo ID desse elemento;
- `document.getElementsByTagName(nome)` – seleciona um elemento pela tag do elemento;
- `document.getElementsByClassName(nome)` – seleciona um elemento pela classe do elemento.

Para realizar alterações usando propriedades:

- **elemento.innerHTML** = **novo HTML** – troca o conteúdo HTML do elemento;
- **elemento.atributo** = **novo valor** – troca o atributo escolhido pelo novo valor;
- **elemento.style.propriedade** = **novo estilo** – troca a propriedade de estilo de um elemento.

Para realizar alterações usando métodos:

- **elemento.setAttribute(atributo, valor)** – troca o valor do atributo do elemento especificado.

NOTA: elementos a negrito devem ser especificados

Para inserir e eliminar elementos da página:

- `document.createElement(elemento)` – cria um elemento HTML;
- `document.removeChild(elemento)` – elimina um elemento HTML;
- `document.appendChild(elemento)` – adicionar um elemento HTML ao elemento passado inicialmente (criar um subelemento);
- `document.replaceChild(novo, antigo)` – substitui um elemento por outro;
- `document.write(texto)` – escreve um valor no elemento.

É ainda possível criar eventos para as funções, assim:

`document.getElementById(id).onclick = function(){código}` – cria uma função que será executada quando o elemento for clicado.

Uma funcionalidade muito interessante e valiosa é a possibilidade de guardar elementos dentro de variáveis, por exemplo:

```
var meuElemento = document.getElementById("elemento");
```

Também é possível selecionar elementos pelos seletores de CSS:

```
var x = document.querySelectorAll("p.intro");
```

É possível utilizar o DOM para editar tanto os documentos HTML como CSS, como por exemplo:

<b>HTML</b>	<code>document.getElementById(id).innerHTML = new HTML</code>
<b>CSS</b>	<code>document.getElementById(id).style.property = new style</code>

Exercícios
------------

1. Crie um documento HTML com três botões. Associado a cada botão deve ser criada uma função para alterar a cor de fundo da página, pode escolher as cores ao seu gosto.
2. Adicione um novo botão que permita a apresentação de um parágrafo de Lorem quando clicado.
3. Crie um quinto botão que esconda ou mostre uma parte do texto do parágrafo anterior.
4. Adicione 2 novos botões para aumentar ou diminuir o tamanho do texto.
5. Tomando em consideração o ponto anterior, quando o tamanho do texto for aumentado deve ser alterado o alinhamento do texto para o centro. Quando o tamanho do texto for reduzido, o alinhamento do texto deve ser definido para a direita.

Bom trabalho,  
Bruno Santos