

**TECNOLOGIA EM SISTEMAS PARA INTERNET
ESCOLA DO MAR, CIÊNCIA E
TECNOLOGIA
UNIVALI**

PROGRAMAÇÃO WEB

Professor: Carlos Henrique Bughi

Modelo de Objetos de Documento (DOM)

DOM

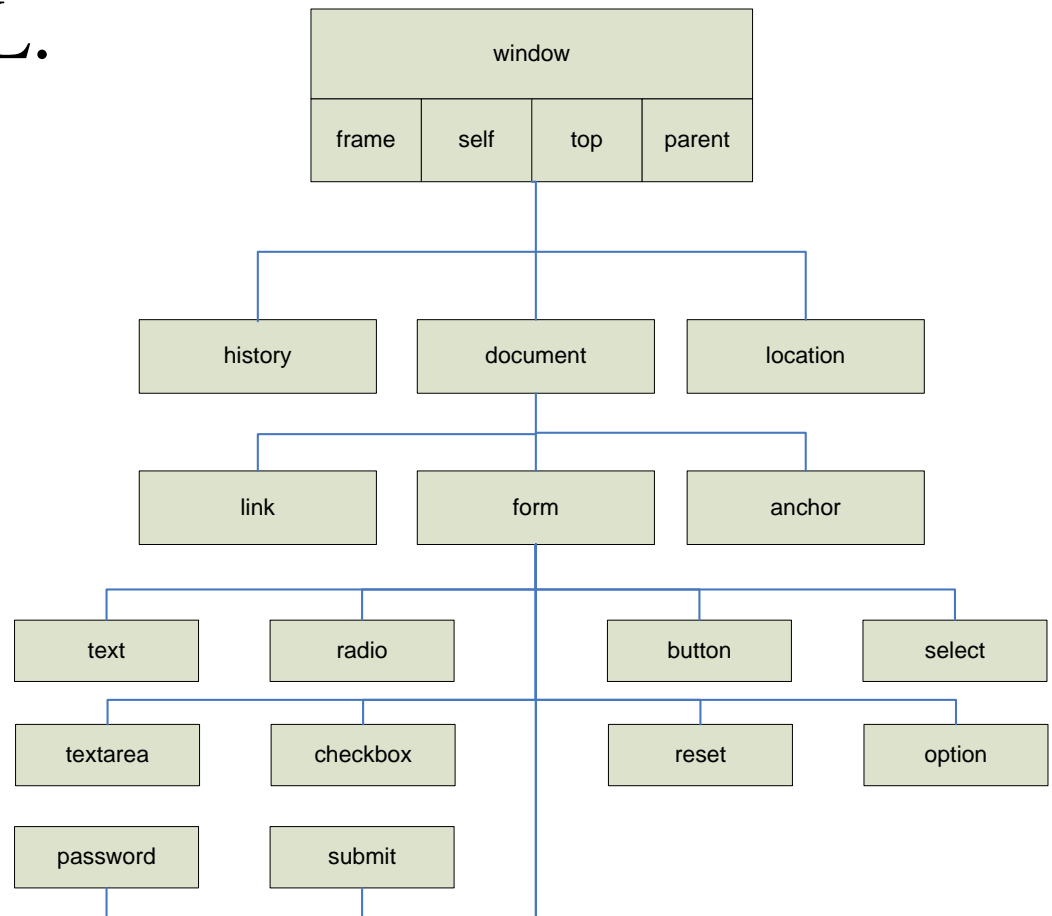
- O Document Object Model (DOM) é uma interface de programação de aplicação (API) para documentos HTML e XML padronizada pela W3C.
- O DOM define a estrutura lógica dos documentos e o modo como um documento é acessado e manipulado.
- Com o DOM, programadores podem construir documentos, navegar por sua estrutura, e adicionar, modificar ou excluir elementos e conteúdo.

DOM

- O HTML-DOM (HTML Document Object Model) define formas padronizadas para acessar e manipular documentos HTML;
- O DOM apresenta um documento HTML como uma estrutura de árvore, com elementos, atributos e textos;

DOM

- Diagrama dos principais elementos de um documento HTML.



DOM

- Antes de falarmos sobre DOM e suas funções, é necessário conhecer os objetos JavaScript que dão suporte ao HTML DOM:
 - Window
 - Representa a janela do navegador, é o objeto de mais auto nível na hierarquia do javascript;
 - Navigator
 - Contém informações sobre o navegador cliente
 - Screen
 - Contém informações sobre a tela do cliente
 - History
 - Armazena as URL já visitadas pelo navegador
 - Location
 - Contém informações sobre a URL atual

Objeto Window

- Link para propriedades e funções:
 - http://www.w3schools.com/jsref/obj_window.asp
- Principais funções
 - alert('mensagem') – Mostra uma tela de alerta;
 - confirm('mensagem') – Mostra uma tela de confirmação (SIM/NÃO)
 - Prompt('texto', 'valor padrão') – Mensagem que requer ação do usuário;
 - open(): Abre uma nova janela do navegador
 - setInterval e clearInterval: avalia uma expressão em intervalos específicos de tempo (clearInterval cancela);
 - setTimeout() e clearTimeout(): avalia uma expressão após um intervalo de tempo específico (clearTimeout cancela);

Objeto Navigator

- Link para propriedades e funções:
 - http://www.w3schools.com/jsref/obj_navigator.asp
- Principais propriedades
 - appName: nome do navegador
 - appVersion: versão do navegador
 - Platform: Sistema operacional
 - cookieEnabled: Retorna true se o navegador está habilitado para cookies;

Objeto Screen

- Link para propriedades e funções:
 - http://www.w3schools.com/jsref/obj_screen.asp
- Principais propriedades
 - width e height: resolução do monitor;
 - availWidth e availHeight: área disponível para uso;

Objeto History

- Link para propriedades e funções:
 - http://www.w3schools.com/jsref/obj_history.asp
- Principais propriedades
 - length: número de elementos no histórico;
- Principais funções
 - back(): carrega a URL anterior;
 - forward(): carrega a próxima URL;
 - go(número|url): carrega um página específica a partir da atual
 - go(-1) // pagina anterior
 - go(1) //proxima pagina

Objeto Location

- Link para propriedades e funções:
 - http://www.w3schools.com/jsref/obj_location.asp
- Principais propriedades
 - host, hostname, href, pathname, port, protocol, search: retorna ou atribui os atributos da URL atual;
- Principais funções
 - assign(): carrega um novo documento;
 - reload(): atualiza o documento atual;

HTML-DOM

- Funções para manipulação de objetos
 - *getElementById*
 - Retorna o primeiro objeto com o ID especificado
 - *getElementsByName*
 - Retorna uma lista de objetos com o nome especificado
 - *getElementsByTagName*
 - Retorna uma lista de todos os elementos de uma tag
 - *createElement*
 - Cria um elemento no node especificado. Retorna o elemento em si
 - *appendChild*
 - Insere um elemento no node especificado após o último elemento existente.
 - *removeChild(elemento)*
 - Remove um elemento do node especificado.
 - *createTextNode*
 - Cria um objeto do tipo texto;

HTML-DOM

Veja mais em:

- https://www.w3schools.com/whatis/whatis_html5.asp
- https://www.w3schools.com/js/js_html5.asp
- <https://tableless.com.br/entendendo-o-dom-document-object-model/>
- https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

HTML-DOM

- **getElementById**

Sintaxe: `document.getElementById(id)`

Retorno: um objeto;

```
<html>
  <head>
    <script type="text/javascript">
      function quemSouEu() {
        var x= document.getElementById("cabecalho")
        alert("Eu sou um elemento " + x.tagName)
      }
    </script>
  </head>
  <body>
    <h1 id="cabecalho" onclick="quemSouEu()">
      Clique para saber que elemento eu sou</h1>
    </body>
</html>
```

HTML-DOM

- getElementByName

Sintaxe: document.getElementById(name)

Retorno: Array contendo todos os elementos encontrados;

```
<html>
  <head>
    <script type="text/javascript">
      function getElement() {
        var x=document.getElementById("meuInput")
        alert(x.length + " elementos!")
      }
    </script>
  </head>

  <body>
    <input name="meuInput" type="text" size="20"><br />
    <input name="meuInput" type="text" size="20"><br />
    <input name="meuInput" type="text" size="20"><br />
    <br />
    <input type="button" onclick="getElement()"
      value="quantos elementos tem o nome 'meuInput' ">
  </body>
</html>
```

HTML-DOM

- `getElementsByTagName`

Sintaxe: `document.getElementsByTagName(name)`

Retorno: Array contendo todos os elementos encontrados;

```
<html>
  <head>
    <script type="text/javascript">
      function getElements() {
        var x=document.getElementsByTagName("input")
        alert(x.length + " elementos!")
      }
    </script>
  </head>
  <body>
    <input name="meuInput" type="text" size="20"><br />
    <input name="meuInput" type="text" size="20"><br />
    <input name="meuInput" type="text" size="20"><br />
    <br />
    <input type="button" onclick="getElements()"
      value="quantos elementos input tem esse documento? ">
  </body>
</html>
```


HTML-DOM

- `document.createElement`
Sintaxe: `createElement(name)`
name: String que especifica um elemento
- `elementoPai.appendChild`
Sintaxe: `appendChild(elemento)`
- `elementoPai.removeChild`
Sintaxe: `removeChild(elemento)`
- `document.createTextNode`
Sintaxe: `document.createtextNode(“texto”)`

HTML-DOM

- Exemplo createElement e appendChild

```
<html>
  <head>
    <script type="text/javascript">
      function criaElemento() {
        var novoElemento=document.createElement("input")
        novoElemento.type="text"
        novoElemento.value="Ola mãe"
        document.getElementsByTagName("body")[0].appendChild(novoElemento)
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="criaElemento()"
      value="Inserir um elemento input">
  </body>
</html>
```

Lista de exercícios

1. Crie uma função javascript que escreva o total de formulários existentes em um documento e o nome de cada formulário;
2. Crie um programa que apresente na tela a hora atual. A hora deverá ser atualizada a cada 1 (um) segundo e o formato deverá ser: hh:mm:ss;
 1. Atenção, estudar as funções setTimeout e setInterval
3. Crie uma função javascript que faça um redirecionamento para a página da UNIVALI após 30 segundos. A função deverá ser chamada no evento onload do elemento body;

Lista de exercícios

4. Crie uma função javascript que escreva o total de elementos H1 existentes em um documento e o conteúdo de cada elemento;
5. Construa uma função javascript para criar um elemento imagem e adicioná-lo ao elemento body;
7. Considerando um elemento do tipo select com o atributo **id** igual a **idades**, crie uma função que receba como parâmetro o nome de uma cidade (String) e insira-o no elemento select;
8. Faça um função javascript para percorrer todos os elementos dos formulários de um documento, passando seus valores (atributo value) para maiúsculo. (É necessário criar um HTML para demonstração)

Lista de exercícios

9. Crie uma função javascript para obter todas as imagens de um documento e mostrar os atributos href de cada elemento no console.;
10. Construa uma função javascript obter todas as imagens de um documento e criar uma tabela contendo as imagens que não possuem o atributo alt definido;

Lista de exercícios

11. Crie uma função que construa dinamicamente a seguinte tabela:

Lin 1 col 1	Lin 1 col 2	Lin 1 col 3
Lin 2 col 1	Lin 2 col 2	Lin 2 col 3
Lin 3 col 1	Lin 3 col 2	Lin 3 col 3

- A tabela deverá ser inserida no elemento body do documento;
- Estude os elementos Table, TableCell e TableRow para realizar o exercício;