**JavaScript – DOM**

**É uma sigla para *Document Object Model*: Modelo de Objeto Documento, ou seja, são os objetos dentro presentes no site.**

**Nota: em javascript, objetos são declarados com o ‘var’.**

**Precisamos conhecer os principais métodos e propriedades do objeto window:**

* ***window* = objeto principal;**
* ***documen*t, *location*, *history* etc. => atributos ou propriedades do objeto window;**
* ***alert*, *prompt* etc => são métodos do objeto window.**

**Dentro de document, temos o html e CSS (eventualmente) em si. Os principais métodos e atributos de document:**

* **head (atributo);**
* **body (atributo);**

**Principais métodos:**

* **write()** = escreve algo na tela, sem mostrar um pop-up como o ***alert()***;
  + É usado apenas para testes!
* **elemento.innerText()** = acessa apenas o texto do elemento HTML;
* **elemento.innerHTML()** = acessa, além do texto, todas as formatações daquele elemento;
* **getElementsByTagName(nometag)[posição]** = acessa o conteúdo de uma tag;
  + É preciso especificar qual tag você quer com [posição];
  + Posição é numérica e começa do 0.
* **getElementById()** = acessa um elemento HTML pelo id;
* **getElementsByName() =** acessa um elemento pelo seu id.
* É preciso especificar qual elemento você quer;
  + Bom para caso haja vários elementos com mesmo id.
* **getElementByClass()**
* **querrySelector(‘elementoxnome’) =** pega elementos pelo id ou pela classe
  + x = **#(hastag)** caso seja id; **.(ponto)** caso seja uma classe;
  + nome = nome do id ou classe

**Eventos DOM**

São ações ou ocorrências que podem acontecer quando há interação com elementos do site, e quando isso acontece, sinais são acionados para que algum código automático possa ser executado. Por exemplo:

* O usuário clicar num botão;
* O usuário apertar alguma tecla;
* Ao terminar de carregar a página;
* Etc.

**Event Handler:** é um bloco de códigos que vai tratar de um evento, por exemplo uma **função**.

***{***

***Bloco de códigos***

***}***

*Iuando estamos configurando um Event Handler, dizemos que estamos* ***Registrando um Manipulador de Eventos****.*

***Event Listener:***é a parte do código queouve o evento, diz qual o Event Handler vai ter que lidar.

Existem três formas de usar **Event Handler**(Manipulador de Eventos):

* Os eventos são propriedades/atributos dos elementos, portanto, podemos modifica-los.

Ex.:

***<div id=’div1’>Teste</div>***

***Function clique() {***

***Var a = document.getElementById(‘div1’);***

***a.innerText=’você clicou em mim’***

***}***

***Teste.onclick = clique;***

*Quando a div for clicada, vai executar a função ‘clique’, que por sua vez vai modificar o texto interno para ‘você clicou em mim’.*

* ***Manipulando os eventos in-line:*** é possível escrever dentro do própria tag do elemento qual evento quero manipular e o que vai acontecer.

***Ex.***

***<div id=’div2’ onclick=’clique()’> Teste </div>***

***Function clique(){***

***Var a = document.getElementById(‘div2’)***

***a.innerText=‘você clicou em mim’***

***}***

Perceba que dentro da própria tag, dentro do documento HTML, podemos manipular o evento.

* **Manipulando eventos com .*addEventListener()*:** é possível manipular os eventos fora do documento HTML.

**Ex.**:

**<div id=’div3’> Teste </div>**

**Var a = document.getElementById(‘div3’);**

**a.addEventListener(\*nome do evento, função a ser executada);**

**\*nome do evento = nome do vento na forma de string(é um pouquinho diferente da forma de atributo);**

**Function clique() {**

**Var a = document.getElementById(‘div3’);**

**a.innerText=’você clicou em mim’**

**}**

**Function sair() {**

**Var a = document.getElementById(‘div3’);**

**a.style.backgroundcolor=’blue’**

**}**

**a.addEventListener(‘click’, clique);**

**a.addEventListener(‘mouseout’, clique);**

**Qual devo usar?**

* ***Manipulador in-line:*** é um método antigo e não deve ser usado porque polui o código HTML já que preciso dizer qual evento vou lidar dentro do próprio documento HTML;
* ***Dois outros métodos:*** são os métodos que devem ser usados. Vantagens:

**Vantagem do manipulador de propriedade:**

* **É mais compatível com os navegadores;**

**Vantagem do método .addListener():**

* + ***É possível remover o listener sem muito esforço com o método .removeListener().***

***Ex.:***

***a.removeListener(‘click’, clique)***

**Para programas pequenos, não faz muita diferença, mas para programas maiores, é muit útil.**

* + **É possível atribuir várias funções para o mesmo evento ao mesmo tempo.**

**Ex.:**

**Usando o manipulador de propriedade:**

**a.onclick = functionA()**

**a.onclick = function()**

**O último iria substituir o primeiro. No entanto, usando o método .addListerner(), isso seria possível:**

**a.addListener(‘click’, functionA)**

**a.addListener(‘click’, functionB)**