

**Bootcamp HTML Web Developer**

**DOM**

**- Manipulação no DOM**

**DOM (Document Object Model) é um modo pelo qual é possível acessar ou modificar os elementos de uma página HTML.**

**Na página HTML temos uma árvore de hierarquia onde cada elemento é um nó.**

**BOM (Browser Object Model) é a árvore de hierarquia na janela do navegador. Tendo o objeto Window (document - history - location - screen - navigator), onde document é o DOM.**

**- Métodos**

**Dentro de uma estrutura HTML temos:**

**Tag – representam um elemento de uma página HTML.**

**Id – é um atributo que dever ter um nome único dentro da página para identificar um elemento dentro da página.**

**Class – é uma identificação de uma configuração que altera a apresentação de mais de um elemento dentro da página.**

* **Para se selecionar um elemento dentro do documento com código javaScript pode se usar o id:**

**document.getElementById(“nome\_do\_id”);**

* **Para se selecionar todos os elementos de uma determinada tag dentro do documento usa-se:**

**document.getElementsByTagName(“nome\_da\_tag);**

**Este processo vai retornar um array com todos esses tipos de tags declarados dentro da página. Para se selecionar uma destas tags especificamente usa-se o índice do array.**

* **Para se selecionar uma classe, usa-se:**

**document.getElementsByClassName(“nome\_da\_classe);**

* **Para se selecionar um elemento que aponte para mais de uma classe ou tag e classe, usa-se.**

**document.querySelectionAll(“.nome\_classe1 .nome\_classe2”);**

**document.querySelectionAll(“nome\_tag .nome\_classe);**

**- Adicionar ou Deletar elementos:**

**\* Criar um novo elemento HTML:**

**document.createElement(tag\_do\_elemento);**

**\* Remover um elemento:**

**document.removeChild(nome\_do\_elemento);**

**\* Adicionar um elemento:**

**document.appendChild(nome\_do\_elemento);**

**\* Substituir um elemento:**

**document.replaceChild(new, old);**

**- Manipulando Classes de Estilos:**

**É possível se fazer a inserção ou remoção de uma classe a um elemento HTML.**

**Para isso usa-se o comando “classList” após o nome do elemento e seguida do método que se irá se utilizar.**

**Pode-se fazer uso de uma constante para armazenar o nome do elemento.**

**const meuElemento = document.getElementById(“id\_do\_elemento\_desejado);**

**\* Para adicionar uma classe de estilo ao elemento meuElemento, usa-se add:**

**meuElemento.classList.add(“nome\_classe\_estilo”);**

**\* Para se remover uma classe de estilo do elemento meuElemento, usa-se remove:**

**meuElemento.classList.remove(“nome\_classe\_estilo”);**

**\* Para se adicionar uma classe, se o elemento não a tiver, ou remover uma classe, se o elemento a tiver, usa-se toggle:**

**meuElemento.classList.toggle(“nome\_classe\_estilo”);**

**- Acessando diretamente o CCS de um elemento**

**\* Pode-se manipular as características de uma tag através do comando:**

**document.getElementsByTagName(“p”).style.color = “red”**

**- Manipulação de Eventos**

**Evento é qualquer tipo de ação permitida para interagir com um elemento da página.**

**\* Tipos de eventos mais comuns:**

**- de mouse**

**mouseover, mouseout**

**- de click**

**click, dbclick**

**- de atualização**

**change(sempre que um campo mudar, load (sempre que uma página carregue)**

**\* Acionamento de eventos:**

**Faz-se uso da função addEventListenner, que é chamada quando uma ação do usuário é realizada.**

**Primeiro captura-se o elemento da página com o qual quer se manipular evento em uma constante, através de sua id.**

**const botao = document.getElementById(“meuBotao”);**

**E usa-se a função addEventListernner para determinar qual evento deve ser escutado e qual ação deve ser realizada.**

**botao.addEventListernner(“click”, funcaoDoClick);**

**\* Acionando eventos diretamente do HTML**

**Para isso coloca-se na frente do nome do evento um “on”, onclick, onmouseover**

**<h1 onclick=”mudaTexto(this)”> Click Aqui </h1>**