**O que são eventos?**

É comum, quando estamos programando em Javascript, querer adicionar um comportamento ao clique de um elemento, ao usuário passar o mouse em cima de um elemento, quando o texto de um <input> for alterado e etc, estas ações são chamadas de eventos, que serão disparados em determinado momento.

Nesse capítulo aprenderemos a como podemos ouvir um evento e executar uma ação a partir dele.

**Calculando o IMC a partir de um clique**

Até agora nós fomos capazes de calcular o IMC de nossos pacientes assim que a página fosse carregada, mas seria mais legal ainda se nós tivéssemos um botão "Calcular IMCs" que, só após ser clicado, preencheria a nossa tabela com os IMCs dos respectivos pacientes.

Para começar então, vamos analisar o código da nossa tabela e botão:

<table>

<tr>

<th>Nome</th>

<th>Peso</th>

<th>Altura</th>

<th>IMC</th>

</tr>

<tr>

<td id="nome-1">Leonardo</td>

<td id="peso-1">50.0</td>

<td id="altura-1">1.60</td>

<td id="imc-1"></td>

</tr>

<tr>

<td id="nome-2">Paulo</td>

<td id="peso-2">100</td>

<td id="altura-2">2.00</td>

<td id="imc-2"></td>

</tr>

</table>

<button id="calcular-imcs" class="botao">Calcular IMCs</button>

Neste momento, ao clicar em nosso botão, nada acontecerá, isso por que ainda não estamos ouvindo o clique dele!

**Como ouvir eventos?**

Para começarmos a ouvir o clique de nosso botão, vamos utilizar o atributo onclick do html, que receber como argumento um código Javascript a ser executado, podemos testar isso utilizando a função alert do javascript.

<button onclick="alert('Calculando Imcs')" id="calcular-imcs" class="botao">Calcular IMCs</button>

Mas isso não irá calcular realmente os imcs dos pacientes, para fazer isso precisamos invocar uma função que pode se chamar, por exemplo, calculaTodosImcs:

<button onclick="calculaTodosImcs()" id="calcular-imcs" class="botao">Calcular IMCs</button>

Porém, para funcionar, precisamos envolver nosso código Javascript em uma função com esse nome:

function calculaTodosImcs(){

//pega trsPacientes

//pega trsPrsPacientes

percorreArray(trsPacientes, function(trAtual){

//monta paciente

var imc = paciente.pegaImc(); // pega o imc do paciente atual

var trDoImc = trDoPaciente.getElementsByClassName("info-imc")[0];

imcTd.textContent = imc; // atualiza imc

console.log(imc); // imprime o imc do paciente atual

});

}

Legal, agora estamos chamando essa função no clique do botão. Mas espere! Note que estamos colocando**Javascript no meio de nosso html**, o que é uma má prática.

O ideal seria que nosso Javascript ficasse totalmente separado da nossa estrutura html, para isso teremos que ouvir o evento do botão sem utilizar o atributo onclick.

Precisamos então manipular um elemento no nosso script, lembra de como fazemos isso? Isso mesmo! Precisamos selecioná-lo!

var botao = document.getElementById("calcular-imc");

Legal, agora já temos nosso botão, mas falta ouvirmos o clique nele, para isso usaremos o atributo onclick, que recebe uma função. Note que, dessa vez, estamos fazendo isso *apenas* no Javascript:

var botao = document.getElementById("calcular-imc");

botao.onclick = calculaTodosImcs();

Pronto! Agora se alguém clicar no botão, uma mensagem dizendo "Alguém clicou no botão!" aparecerá no console!

O problema de se utilizar essa abordagem é que nós só conseguimos adicionar uma função por evento, ou seja, não conseguimos efetuar mais de uma função no clique de um usuário.

Por exemplo, se atribuirmos duas vezes uma função ao atributo onclick do botão, como no seguinte código, somente a ultima delas será invocada ao se clicar no botão:

var botao = document.getElementById("calcular-imc");

botao.onclick = calculaTodosImcs;

botao.onclick = function(){

console.log("Calculando imcs");

};

Para resolver esse problema podemos utilizar a função addEventListener, presente em navegadores modernos. Essa função recebe dois parâmetros: o evento que se quer ouvir e a ação que deseja fazer:

var botao = document.getElementById("calcular-imc");

botao.addEventListener("click", function(){

console.log("Alguém clicou no botão!");

});

botao.addEventListener("click", function(){

console.log("Essa é outra função");

});

Deste modo podemos executar varias funções no mesmo evento! Note que quem vai executar essa função é o próprio navegador ao detectar o evento de clique. Esse tipo de função anônima que passamos como argumento para ser executada após determinada ação é denominada *função de callback*

**Usando o evento mouseover**

Além do evento de clique("click"), ainda temos outros que poderemos usar, por exemplo:

* dblclick (clique duplo)
* mouseover (ao passar o mouse por cima de um elemento)
* keydown (ao pressionar uma tecla)

Entre outros

Com isso nós podemos, por exemplo, alterar a cor de um <tr> ao passarmos o mouse por cima dele, para isso vamos primeiro selecionar todos os <tr>s, lembre-se de que a função getElementsByTagName retorna sempre umArray

var trs = document.getElementsByTagName("tr");

Agora precisamos escutar o evento mouseover em cada um dos trs, portanto precisamos percorrer o nossoArray, lembre-se de que nós temos a função percorreArray, que faz exatamente isso:

var trs = document.getElementsByTagName("tr");

percorreArray(trs, function(trAtual){

// ouve o evento no trAtual

})

Agora basta ouvir o evento no trAtual e passar a função que queremos executar:

var trs = document.getElementsByTagName("tr");

percorreArray(trs, function(trAtual){

trAtual.addEventListener("mouseover", function(){

//aqui queremos pintar a tr de cinza

});

Legal, estamos escutando o evento, agora precisamos escrever o código para pintar a <tr> que o usuário passou o mouse de cinza, essa tr pode ser acessada através da palavra reservada this dentro da função de callback. Para pintá-la, então, basta mudar o atributo bgcolor dela para grey, o que pode ser feito a partir da funçãosetAttribute:

var trs = document.getElementsByTagName("tr");

percorreArray(trs, function(trAtual){

trAtual.addEventListener("mouseover", function(){

this.setAttribute("bgcolor", "grey");

});

});

**Alterando o DOM a partir de um evento.**

Agora que já sabemos como ouvir eventos em elementos, podemos finalmente chamar a nossa função que calcula os IMCs no clique do botão. Para isso, basta passarmos a nossa função como segundo parâmetro doaddEventListener:

function calculaTodosImcs(){

//pega trsPacientes

percorreArray(trsPacientes, function(paciente){

//monta paciente

var imc = paciente.pegaImc(); // pega o imc do paciente atual

var trDoImc = trDoPaciente.getElementsByClassName("info-imc")[0];

imcTd.textContent = imc; // atualiza imc

console.log(imc); // imprime o imc do paciente atual

});

}

var botao = document.getElementById("calcular-imc");

botao.addEventListener("click", calculaTodosImcs);