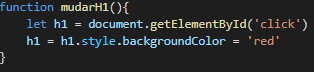
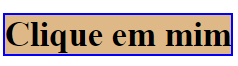
**Eventos DOM**

É possível executar códigos específicos quando alguma coisa acontece, os eventos, como: clicar em algum elemento, carregar a página, clicar em um botão etc.

Os eventos também são atributo dos elementos HTML e podem se acessados de duas formas:

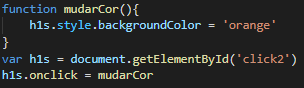
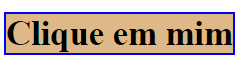
**Evento Inline:** quando alteramos dentro da tag do elemento.

Ex.:



Quando o **evento onclick**(clica com o botão esquerdo do mouse em cima) for disparado, vai executar a **função mudarH1**, que vai **alterar a cor de fundo para vermelho**.

**Evento externo:** quando lidamos com o evento fora do documento html, dentro do arquivo .js

Ex.:

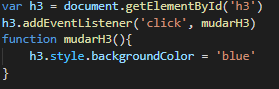
**No evento onlclick, a função mudarCor vai ser executada**

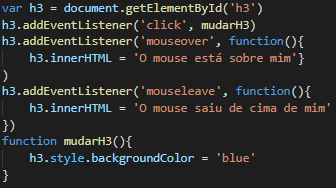
**Para mais tipos de eventos, acesse** [**Events**](https://www.w3schools.com/jsref/dom_obj_event.asp)

**Event Listener**

É um método dos elementos que serve para melhor organizar os eventos.

* Termos básicos:
  + **Event Handler:** função que executa quanto o evento é disparado;
  + **Event Listener:** método que lida com os eventos.
* É possível adicionar vários event Listeners para um mesmo elemento.

Ex.:



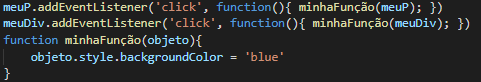




**Se eu quiser passar uma função com parâmetro?**

Use uma **função anônima** da seguinte forma:

**Elemento.addEventListener(nome do evento, function(){ nome da função(x, y, ...); })**

Ex.:

**Tipos de Eventos**

Há dois tipos de eventos em HTML DOM:

* **Event Bubbling(borbulhamento):** onde o elemento mais **interno** acontece primeiro;
* **Event Capturing(Captura):** onde o elemento mais **externo** acontece primeiro.

Se tivéssemos uma <p> dentro de uma <div>e quiséssemos fazer com que o evento do elemento <p> acontecesse primeiro?

Método addEventListener aceita três parâmetros:

***Elemento.addEventListener(nome do evento, função, useCapture)***

* Parâmetro booleano useCapture diz o tipo de evento, onde:
  + **True: evento do tipo Captura;**
  + **False(padrão): evento do tipo Borbulhamento.**

**Ex.:**

**Como removo eventos de um elemento?**

Basta usar o método **removeEventListener(nome do evento, função daquele evento).**

Ex.: