# Aula 10 - eventos dom

Breve revisão:

- O que significa a silga DOM? Documet Object ‘Manipulation’(errei, era model)

- Sabe montar uma árvore DOM em um website simples?

(Praticar isso) Seria algo com window.document.getElementBy...

- O que são elementos Parent e os elementos Child em um arvore DOM? O que está dentro de um elemento é filho, e então, o que abriga ou está ‘guardando’ o elemento é pai.

- Qual os cinco principais métodos de selecionar elementos DOM dentro do JavaScript?

window.document.getElementbById, ou TagName ou Name ou ClassName ou apenas querySelector.

## eventos dom

Vamos supor que criamos uma div em um documento HTML, tudo o que pode acontecer com essa div podemos chamar de Evento DOM, o mais comum por exemplo, eventos(ações) com o mouse. Exemplo:

Mouse entrou na div:



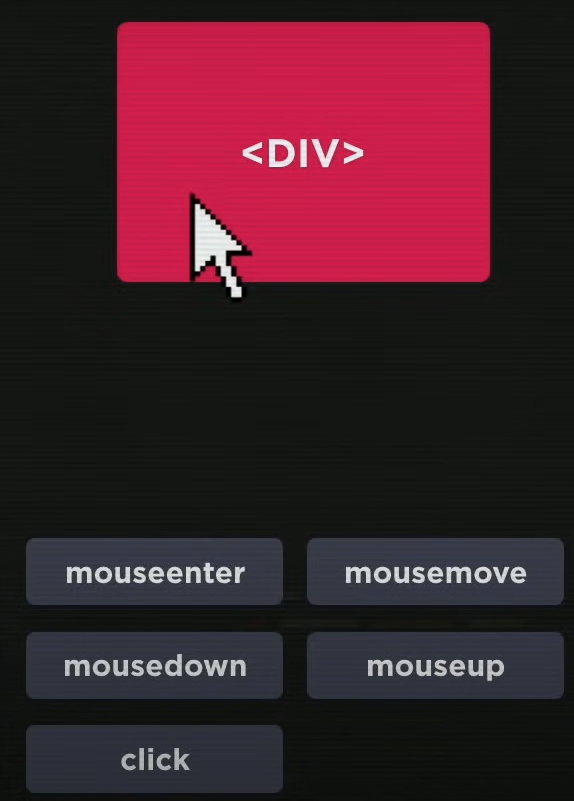
Mexendo o mouse dentro da div:



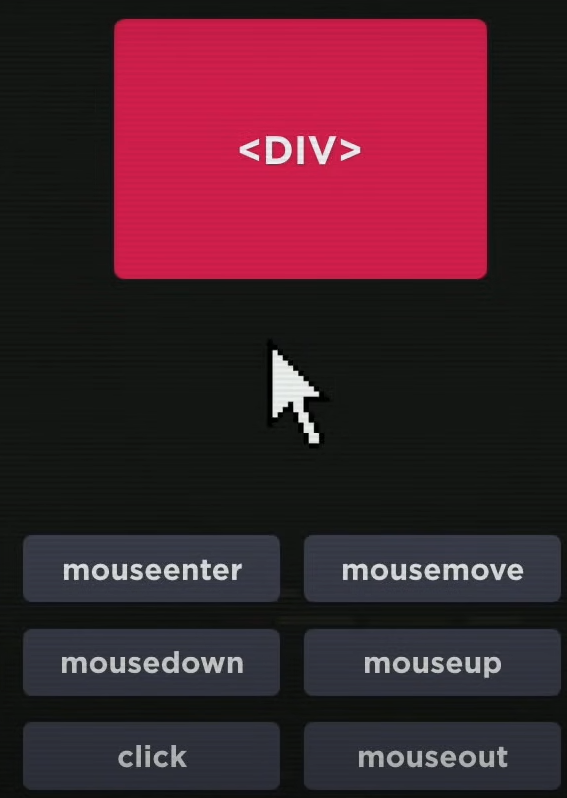
Apertei o mouse(down) 5 segundo e soltei o mouse(up):



Simplesmente cliquei:



Sai da div:

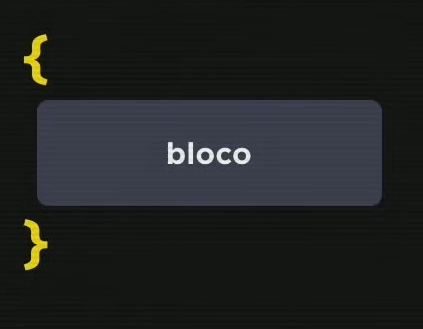


Para ver mais eventos dom, podemos procurar no google: Javascript events dom list, e veremos muitos mas muitos mesmo tipos de eventos DOM.

## função

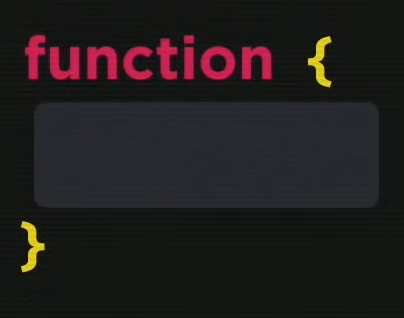
É um conjunto de linhas ou códigos que vão ser executadas quando um evento ocorrer, e para executar essas linhas somente quando um evento ocorrer, precisamos colocar esses códigos dentro de um bloco.

Repare na próxima imagem que o bloco está dentro de chaves na primeira e última linha, um bloco em JavaScript é delimitado por essas chaves.



Devemos nomear esse bloco com a palavra **function**

**para nomear esse bloco antes das chaves.**



Também precisamos dar um nome para uma função desta forma:



E opcionalmente podemos adicionar alguns parâmetros(argumentos) nessa função:



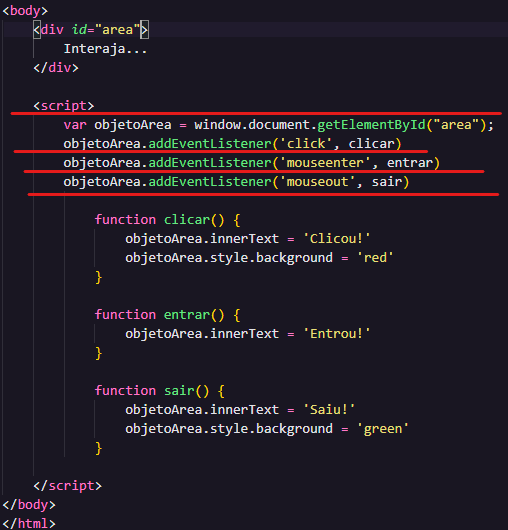
## exemplo de functions no código

Sublinhado em vermelho, alguns eventos DOM.



Mas para deixar nosso HTML mais limpo, em vez de criar todos aqueles eventos ‘onclick’ etc, podemos criar o ‘addEventListener’.

Como o nome diz, ele seria como um ‘ouvidor’ que fica ali esperando ‘ouvir’ algum evento pra fazer algo.



Deste jeito iremos fazer a mesma coisa que no código anterior, porém, deixando o HTML mais limpo sem tanta poluição.

## observação

O JavaScript puro geralmente não costuma apresentar erros no editor de código, por isso, sempre que estiver mexendo com JS e não saber onde está um erro, use o dev tools do chrome para ver onde ele aponta algum erro no nosso código.

Exemplo exercicio007:

