

# JAVASCRIPT - MÓDULO 1 - AULA 4

## 1. EVENTOS DOM

- **O que são eventos DOM?**

Eventos DOM são ações ou ocorrências que acontecem em um documento HTML e podem ser manipuladas por meio de JavaScript. Eles permitem que você crie páginas web interativas e responsivas.

- **Por que usar eventos?**

Os eventos permitem que os usuários interajam com a página, como clicar em botões, preencher formulários, mover o mouse, entre outras ações.

## 2. FUNÇÕES

Antes de entendermos alguns tipos de eventos DOM, precisamos compreender como funcionam as funções. Pois, para executar quaisquer desses exemplos eu preciso fazê-la dentro de uma função. Para que não fiquemos presos a um modelo de desenvolvimento linear, em que ele precisa executar tudo para só assim entrar no que queremos. Com a função, executaremos apenas se for “solicitado” a execução.

Mas primeiro de tudo. O que são funções? As funções são blocos de código reutilizáveis que podem ser definidos e chamados em diferentes partes do seu programa. Existem várias maneiras de definir funções em JavaScript, mas a estrutura básica segue um modelo de sintaxe semelhante.

### 2.1. Sintaxe de uma função:

```
function nomeDaFuncao(parametro1, parametro2, ...) {  
  
    // Corpo da função  
    // Código a ser executado  
  
    return resultado; // Opcional, pode retornar um valor  
}
```

Repare como é simples a sintaxe de uma função. Essa é a sintaxe mais comum. Mas para entender melhor o funcionamento, vamos ver alguns exemplos.

## 2.2. Exemplos de funções:

### EXEMPLO 1

```
function somar(a, b) {  
  return a + b;  
}
```

```
var resultado = somar(3, 4);  
window.document.write("O resultado da soma é:" + resultado)
```

---

### EXEMPLO 2

#### SINTAXE:

```
var nomeDaFunção = function(parametro1, parametro2){  
  
  // Corpo da função  
  // Código a ser executado  
  
  return resultado; // variável opção, pode retornar valor direto  
}
```

#### USO PRÁTICO:

```
var multiplicar = function(x, y) {  
  return x * y;  
}  
  
var resultado = multiplicar(5, 6);  
window.document.write("O resultado da soma é:" + resultado)
```

---

## 3. TIPOS DE EVENTOS

Existem inúmeros eventos que podem ser aplicados em nossa programação com JavaScript. Não veremos todos em nosso curso, abaixo serão destacados alguns deles. E o objetivo na aula de hoje é entender primeiramente os eventos de mouse. Mas caso tenha interesse em se aprofundar, pode acessar o site oficial do Mozilla que conta com todos os eventos prontos e com exemplos.

**SITE** - <https://developer.mozilla.org/en-US/docs/Web/Events>

Portanto seguem abaixo alguns exemplos dos tipos de eventos mais utilizados. Que veremos no curso.

- **Eventos de Mouse:**

**mousedown:** Acionado quando é clicado o botão do mouse e segurado.

**mouseup:** Acionado quando solta o botão do mouse que estava clicado.

**mouseenter:** Disparado quando o mouse passa por cima de um elemento.

**mouseout:** Disparado quando o mouse deixa um elemento.

**mousemove:** Mover dentro de um elemento.

**click:** Acionado quando o botão do mouse é clicado.

- **Eventos de Teclado:**

**keydown:** Disparado quando uma tecla é pressionada.

**keyup:** Acionado quando uma tecla é liberada.

**keypress:** Disparado quando uma tecla é pressionada e liberada.

- **Eventos de Formulário:**

**submit:** Disparado quando um formulário é enviado.

**input:** Disparado quando o valor de um campo de entrada muda.

**change:** Acionado quando o valor de um campo de seleção muda.

- **Eventos de Documento:**

**DOMContentLoaded:** Acionado quando o documento HTML é completamente carregado e analisado, incluindo estilos e imagens.

## 4. COMO USAR UM EVENTO DOM

Para usar um evento DOM em JavaScript com a estrutura padrão, você seguirá os seguintes passos:

1. Selecione o elemento HTML ao qual deseja adicionar o evento. Você pode fazer isso usando métodos como `getElementById`, `querySelector`, `getElementsByClassName`, etc.
2. Crie uma função JavaScript que será executada quando o evento ocorrer. Esta função é chamada de "callback" e conterá o código que você deseja executar quando o evento for acionado.

3. Adicione um event listener ao elemento selecionado. O event listener é responsável por "ouvir" o evento específico e chamar a função de callback quando o evento ocorrer.

Depois de entender todas as informações para a criação de um evento DOM vamos ver sua sintaxe:

```
// Passo 1: Selecionar o elemento HTML
var elemento = document.getElementById('meuElemento');

// Passo 2: Criar uma função de callback
function minhaFuncaoDeCallback(event) {
    // Código a ser executado quando o evento ocorrer
    window.alert('Evento DOM ocorreu!', event);
}

// Passo 3: Adicionar um event listener ao elemento
elemento.addEventListener('nomeDoEvento', minhaFuncaoDeCallback);
```

Por fim, vamos ver também um exemplo prático, utilizando tudo que aprendemos até agora.

```
<!DOCTYPE html>
<html>
<head>
  <title>Evento DOM</title>
</head>
<body>
  <button id="meuBotao">Clique-me</button>
  <!-- Outra opção para criação de um botão <input type="button" id="meuBotao" value="clica em mim">-->
  <script>
    // Passo 1: Selecionar o elemento HTML
    const botao = document.getElementById('meuBotao');

    // Passo 2: Criar uma função de callback
    function minhaFuncaoDeCallback(event) {
      // Código a ser executado quando o botão for clicado
      alert('Botão clicado!');
    }

    // Passo 3: Adicionar um event listener ao elemento
    botao.addEventListener('click', minhaFuncaoDeCallback);
  </script>
</body>
</html>
```

## 5. USANDO EVENTOS DE MOUSE

Já entendemos a definição de alguns dos eventos de mouse em JavaScript. Mas vamos agora ver como eles funcionam dentro do código. Atenção, você também deve ter entendido o uso das funções para entender o código a seguir:

### EXEMPLO 1:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Eventos DOM</title>
  <style>
    div#area{
      background: green;
      color: white;
      width: 200px;
      height: 200px;
      line-height: 200px;
      text-align: center;
    }
  </style>
</head>
<body>

  <div id="area">
    Interaja...
  </div>

  <script>
    var a = window.document.getElementById('area')
    a.addEventListener('click', clicar)
    a.addEventListener('mouseenter', entrar)
    a.addEventListener('mouseout', sair)

    function clicar(){
```

```

        a.innerText = 'Clicou!'
        a.style.background = 'red'
    }
    function entrar(){
        a.innerText = 'Entrou!'
    }
    function sair()
    {
        a.innerText = 'Saiu!'
        a.style.background = 'green'
    }
</script>
</body>
</html>

```

Repare que no código acima, utilizamos três diferentes eventos de mouse, todos eles executando uma função diferente. Tudo isso deixa nosso código independente. Tratando bloco a bloco conforme a necessidade.

## EXEMPLO 2:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Somando números</title>
    <style>
        body{
            font: normal 18pt Arial;
        }
        input{
            font: normal 18pt Arial;
            width: 100px;
        }
    </style>
</head>
<body>

```

```

<h1>Somando valores</h1>
<input type="number" name="txtn1" id="txtn1">
<input type="number" name="txtn2" id="txtn2">
<input type="button" value="somar" onclick="somar()">
<div id="res">Resultado</div>

<script>
    function somar()
    {
        var tn1 = window.document.getElementById('txtn1')
        var tn2 = window.document.querySelector('input#txtn2')
        var res = window.document.getElementById('res')
        var n1 = Number(tn1.value)
        var n2 = Number(tn2.value)
        var s = n1 + n2
        res.innerHTML = `a soma entre ${n1} e ${n2} é
igual a ${s}`
    }
</script>
</body>
</html>

```

Já no exemplo acima, repare que utilizamos caixas de texto com INPUT, (que já aprendemos em HTML e CSS no primeiro semestre.