

# Aula 03 - Manipulação de eventos do DOM

**eventos** são ações ou ocorrências que acontecem na página da web e podem ser detectadas pelo código JavaScript. Eles podem ser acionados pelo usuário (como cliques, pressionamento de teclas, movimentação do mouse) ou pelo próprio navegador (como o carregamento da página ou mudanças em elementos).

Os eventos permitem que os desenvolvedores criem páginas interativas e dinâmicas, respondendo a ações dos usuários em tempo real.

## Tipos de Eventos no JavaScript

Os eventos no JavaScript podem ser classificados de várias formas, incluindo:

### 1. Eventos de Mouse

- `click` → Disparado quando um elemento é clicado.
- `dblclick` → Disparado quando um elemento é clicado duas vezes rapidamente.
- `mouseover` → Disparado quando o mouse passa sobre um elemento.
- `mouseout` → Disparado quando o mouse sai de um elemento.

### 2. Eventos de Teclado

- `keydown` → Disparado quando uma tecla é pressionada.
- `keyup` → Disparado quando uma tecla é solta.

### 3. Eventos de Formulário

- `submit` → Disparado quando um formulário é enviado.
- `change` → Disparado quando um valor de campo de formulário muda.
- `focus` → Disparado quando um campo recebe foco (exemplo: um input é clicado).
- `blur` → Disparado quando um campo perde o foco.

### 4. Eventos de Janela e Documento

- `load` → Disparado quando a página ou um recurso termina de carregar.
- `resize` → Disparado quando a janela do navegador é redimensionada.
- `scroll` → Disparado quando ocorre um deslocamento na página.

## Adicionando um Evento

```
document.querySelector("button").addEventListener("click", function() {  
    alert("Botão clicado!");  
});
```

## Exemplos Práticos

### Exemplo 1: Alterar o texto de um elemento ao clicar em um botão

```
button id="meuBotao">Clique para mudar o texto</button>  
<p id="mensagem">Texto original</p>  
  
<script>  
    document.getElementById("meuBotao").addEventListener("click", function() {  
        document.getElementById("mensagem").innerText = "Texto alterado!";  
    });  
</script>
```

### Exemplo 2: Mudar a cor de fundo ao passar o mouse

```
<div id="area" style="width: 200px; height: 100px; background-color: light gray; text-align: center; line-height: 100px;">  
    Passe o mouse aqui  
</div>
```

```
<script>
  let area = document.getElementById("area");

  area.addEventListener("mouseover", function() {
    area.style.backgroundColor = "yellow";
  });

  area.addEventListener("mouseout", function() {
    area.style.backgroundColor = "lightgray";
  });
</script>
```

### Exemplo 3: Capturar o que o usuário digita em tempo real

```
<input type="text" id="campoTexto" placeholder="Digite algo...">
<p id="saida"></p>

<script>
  document.getElementById("campoTexto").addEventListener("input", function(event) {
    document.getElementById("saida").innerText = "Você digitou: " + event.target.value;
  });
</script>
```

## Conclusão

Os eventos no JavaScript são essenciais para criar páginas dinâmicas e interativas. Eles permitem capturar ações do usuário e responder de forma imediata, tornando a experiência do usuário mais fluida.

A melhor prática é sempre utilizar `addEventListener`, pois ele permite adicionar múltiplos eventos a um mesmo elemento sem sobrescrever funções já

existentes.

## **Atividades práticas**

1. A partir de uma lista de objetos de produtos no estoque, adicionar as linhas na tabela de produtos. Chamar a função de atualizar estoque total ao final.
2. Criar um botão para remover um produto da lista em cada linha. Ao remover deve atualizar a quantidade total no estoque.
3. Criar dois botões para incrementar e decrementar a quantidade no estoque. A cada alteração atualizar a quantidade total de elementos. A cada operação chamar uma função que verifica a quantidade do estoque e pinta o fundo da cor correspondente.