

# Eventos

ESTUDO SOBRE A UTILIZAÇÃO DE EVENTOS EM JAVASCRIPT

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

## Eventos

Como observado no tópico anterior, a captura de eventos é possível e um importante meio para controlar o ambiente HTML. Outros eventos podemos ser associados aos diversos elementos disponíveis HTML.

Vamos analisar o código a seguir:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body onLoad="alert('Mensagem inicial!')">
8.         <script type="text/javascript">
9.             function clicar_Down(){
10.                 alert("Você clicou no botão");
11.             }
12.
13.             function clicar_Up(){
14.                 alert("Você SOLTOU o botão");
15.             }
16.         </script>
17.         <form>
18.             <input type="button" value="Clique aqui" onMouseDown="clicar_Down"
19.             </form>
20.
21.     </body>
22. </html>
```

Neste programa, temos alguns novos eventos.

`onLoad`

Evento capaz de iniciar uma funcionalidade na inicialização do corpo do HTML (body).

`onMouseDown`

Evento destinado a capturar o momento em que o usuário PRESSIONA o botão do mouse sobre o elemento observado.

`onMouseUp`

Evento destinado a capturar o momento em que o usuário SOLTA o botão do mouse sobre o elemento observado. Inverso do `onMouseDown`.

`onFocus`

Evento destinado a disparar a ação de foco em um elemento. Sua execução permanecerá ativa enquanto o objeto estiver focado. Em algumas situações, se não tratado, pode se tornar um inconveniente.

Vamos criar um novo programa para testar novas possibilidades:

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body onLoad="alert('Mensagem inicial!')">
8.         <script type="text/javascript">
9.             function clicar_Down(){
10.                 alert("Você clicou no botão");
11.             }
12.
13.             function clicar_Up(){
14.                 alert("Você SOLTOU o botão");
15.             }
16.
17.             function focar_Elemento(){
18.                 alert("Você focou um elemento!");
19.             }
20.
21.         </script>
22.         <form>
23.             <b>Nome</b>
24.             <input type="text" id="nome" onFocus="focar_Elemento()">
25.             <b>Endereço</b>
26.             <input type="text" id="endereco" >
27.
28.             <input type="button" name="btn" value="Clique aqui" onMouseDov
29.         </form>
30.
31.     </body>
32. </html>

```

Neste código observamos os eventos iniciados por ações específicas. Em destaque o `onLoad`, que carrega um evento no carregamento do `body` (corpo do HTML). Os demais eventos são exemplos de captura de eventos dos objetos.

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 7</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body onLoad="alert('Mensagem inicial!')">
8.         <script type="text/javascript">
9.             var aviso = false;
10.            function clicar_Down(){
11.                alert("Você clicou no botão");
12.            }
13.
14.            function clicar_Up(){
15.                alert("Você SOLTOU o botão");
16.            }
17.
18.            function focar_Elemento(){
19.                if (aviso == false){
20.                    alert("Você focou um elemento!");
21.                    aviso = true;
22.                }
23.                document.getElementById("endereco").style.background = "yellow";
24.            }
25.        </script>
26.        <form>
27.            <b>Nome</b>
28.            <input type="text" id="nome" onFocus="focar_Elemento()">
29.            <b>Endereço</b>
30.            <input type="text" id="endereco" >
31.
32.            <input type="button" name="btn" value="Clique aqui" onMouseDown="clicar_Down()">
33.        </form>
34.
35.    </body>
36. </html>

```

Neste próximo código vamos verificar o uso do evento onChange, onde a mudança em um objeto é capturada e disparada em um evento. Para tanto, vamos utilizar um elemento select.

Veja o código a seguir:

```
1. <!DOCTYPE html>
2. <html>

3.     <body>
4.
5.         <p>Selecione um novo treinamento a realizar.</p>
6.
7.         <select id="curso" onChange="alterarCurso()">
8.             <option value="Java">Java
9.             <option value="C#">C#
10.            <option value="Linguagem C">Linguagem C
11.            <option value="JavaScript">JavaScript
12.        </select>
13.
14.        <p>Esperamos poder participar de mais um treinamento!</p>
15.        <p id="texto"></p>
16.
17.        <script>
18.            function alterarCurso() {
19.                var x = document.getElementById("curso").value;
20.                document.getElementById("texto").innerHTML = "Escolha: "
21.            }
22.        </script>
23.    </body>
24. </html>
```

## DICA:

Para saber mais, veja em:

[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp) ([https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp))

[https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

([https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp))

Estes códigos possuem vários eventos simples delimitados e alinhados aos elementos do formulário.

Há uma grande série de eventos disponíveis já na versão atual do HTML5, distribuídos nos seguintes grupos:

- Atributos de Janelas
- Formulários
- Teclado
- Mouse
- Arrasto
- Área de Transferência
- Media

Para se aprofundar mais verifique a documentação detalhada em:

[https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

([https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp))

## Conclusão

Os eventos podem capturar momentos e ações diferentes de elementos HTML. Utilize este recurso para alterar, validar ou controlar seus elementos e objetos. Tome a precaução de evitar códigos instáveis e incorretos, pois o seu uso pode gerar erros significativos.

## ATIVIDADE

Qual evento trata da alteração de estado de elementos HTML?

- A. onChanlenger="funcao()"
- B. onChar="funcao()"
- C. onChange=true
- D. onChange="funcao()"

## ATIVIDADE

Qual evento proporciona a ação de soltar o botão do mouse?

- A. onMouseUP="clicar\_Down()"
- B.MouseDown="clicar\_Down()"
- C. onMouse="clicar\_Down()"
- D. onMouseDown="clicar\_Down()"

# ATIVIDADE

Qual evento permite capturar o foco em um elemento HTML?

- A. onFocus="focar\_Elemento()"
- B. Focus="focar\_Elemento()"
- C. onFocalizar="focar\_Elemento()"
- D. onFocar="focar\_Elemento()"

# REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO, C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.





