

Linguagens de Marcação e Scripts

Prof. Aníbal Cavalcante de Oliveira

UFC - QXD0164 - 2019.2

Agenda - Aula 17

- Eventos em JavaScript na Web

Os eventos HTML DOM

O HTML DOM permite reagir a eventos HTML. Assim, um código JavaScript pode ser executado quando um evento ocorre em uma página web. No HTML DOM os principais eventos são:

- 1 - Quando um usuário clica o mouse.
- 2 - Quando uma página web foi carregada.
- 3 - Quando uma imagem tiver sido carregada.
- 4 - Quando o mouse se move ao longo de um elemento.
- 5 - Quando um campo de entrada for alterado.
- 6 - Quando um formulário é submetido ao servidor remoto.
- 7 - Quando uma tecla do teclado é pressionada.
- 8 - Quando um elemento da página recebe ou perde o foco.
- 9 - Quando o conteúdo de campo de entrada muda.

Lista de todos os eventos HTML DOM:

https://www.w3schools.com/jsref/dom_obj_event.asp

0 evento de clique

O evento **onclick** captura o evento de clique do mouse. Podemos usar o atributo **onclick** no próprio elemento, adicioná-lo via propriedade no código JavaScript, ou usar o método **addEventListener()**.

```
<element onclick="myScript">
```

```
/*Abaixo o script injeta uma função e vai reagir ao evento de click do mouse.*/
```

```
object.onclick = function(){myScript};
```

```
object.addEventListener("click", myScript);
```

O evento de descida e subida do botão do mouse.

Os eventos `mousedown` e `mouseup` são capturados quando clicamos no mouse também. Porém o `mousedown` é ativado quando pressionamos qualquer botão. Já o `mouseup` é chamado quando liberamos o mesmo botão.

```
object.addEventListener('mousedown', function (e){});
```

```
object.onmouseup = function(){myScript};
```

```
object.onmousedown = function(){myScript};
```

0 evento de duplo clique do mouse

Captura o evento de duplo clique no mouse.

```
<element ondblclick="myScript">
```

```
/*Abaixo o script injeta uma função e vai reagir ao evento de click do mouse.*/
```

```
object.ondblclick = function(){myScript};
```

```
object.addEventListener("dblclick", myScript);
```

Os Eventos onload e onunload (carregar e sair de uma página)

O **onload** e **onunload** são eventos acionados quando o usuário entra ou sai da página.

O evento **onload** pode ser usado para verificar a versão do navegador e carregar a versão apropriada da página web com base nessas informações.

O **onload** e **onunload** também são eventos que podem ser usados para lidar com cookies.

```
<script>
function checkCookies() {
  var text = "";
  if (navigator.cookieEnabled == true) {
    text = "Cookies estão habilitados.";
  } else {
    text = "Cookies estão habilitados.";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
```

Exemplo do onunload

Alterar um dos atributo de uma tag html existente.

```
<!DOCTYPE html>
<html>
<body>

<h1>Bem-Vindo à minha página inicial</h1>

<p> Feche esta janela ou pressione F5 para recarregar a página. </p>
<p><strong> Observação: </strong> devido a diferentes configurações do navegador, este evento
pode nem sempre funcionar como esperado. </p>

<script>
document.body.onunload = function() {
    console.log("Obrigado por ter me visitado!!!");
}
</script>
</body>
</html>
```


Evento do mouse se movendo sobre um elemento da tela.

Exemplo de `onmouseover` e `onmouseout` já vistos em sala.

```
<!DOCTYPE html>
<html>
<body>

<h1 onmouseover="style.color='red'" onmouseout="style.color='black'">
  Mouse sobre o texto.
</h1>

</body>
</html>
```

Evento do mouse se movendo sobre um elemento da tela.

Podemos recuperar a posição do mouse em coordenadas no **eixo X e Y** de um elemento.

```
<div id="div" style="width: 300px; height: 300px; background-color: yellow"></div>
```

```
<script>
```

```
document.getElementById("div").onclick = function(event) {  
    var x = event.clientX;  
    var y = event.clientY;  
    var coords = "X coords: " + x + ", Y coords: " + y;  
    document.getElementById("div").innerHTML = coords;  
}
```

```
</script>
```

Evento disparado quando um elemento da página recebe o foco.

Onfocus é disparado quando um elemento recebe o foco.

```
Digite seu nome: <input type="text" onfocus="myFunction(this)">
```

```
<p>
```

Quando o campo de entrada recebe o foco, é acionada uma função que altera a cor de fundo.

```
</p>
```

```
<script>
```

```
function myFunction(x) {
```

```
    x.style.background = "yellow";
```

```
}
```

```
</script>
```

Evento disparado quando ocorre uma mudança no elemento.

O evento **onChange** é utilizado para que seja realizada determinada ação após alguma mudança de seleção de um elemento.

```
Insira seu Nome: <input type="text" id="fname" onchange="myFunction()">
<p>
    Ao clicarmos fora do input text o texto escrito nele ficará todo em caixa alta.
</p>

<script>
    function myFunction() {
        var x=document.getElementById("fname");
        x.value=x.value.toUpperCase();
    }
</script>
```

Evento disparado quando ocorre uma mudança no elemento.

Exemplo de seleção de item em uma lista.

```
<select id="mySelect" onchange="myFunction()">
    <option value="Audi">Audi</option>
    <option value="BMW">BMW</option>
    <option value="Mercedes">Mercedes</option>
    <option value="Volvo">Volvo</option>
</select>

<script>
    function myFunction() {
        var x = document.getElementById("mySelect").value;
        document.getElementById("demo").innerHTML = "Você selecionou: " + x;
    }
</script>
```

Evento disparado quando uma tecla é pressionada.

```
<p>Pressione uma tecla do teclado e veja o código dela em Unicode.</p>

<input type="text" size="40" onkeypress="myFunction(event)">

<p id="demo"></p>

<script>
  function myFunction(event) {
    var x = event.which || event.keyCode;
    document.getElementById("demo").innerHTML = "The Unicode value is: " + x;
  }
</script>
```