Eventos

Nessa aula, veremos **o conceito de evento.** Veremos também quais os seus tipos, como **associar trechos de código aos eventos** e outros detalhes sobre **programação dinâmica**.

Objetivos da aula:

Nessa aula você aprenderá:

- o que são eventos
- quais os **tipos de eventos**
- como fazer a **associação de código** a eventos
- como usar a palavra chave `this`
- usar eventos como **propriedade de objetos**

O que são eventos?

Um **evento** é um acontecimento iniciado por alguma **atitude do usuário:**

- o movimentar do mouse,
- o clicar de um botão,
- o envio de um formulário

ou pelo **funcionamento do navegador:**

- o carregamento de uma página,
- o não conseguir carregar uma imagem, etc.

O que é "capturar" e "tratar" um evento ?

- Todo evento envolve uma **ação** e um **elemento** da página que **sofre** esta ação.
- É possível definir **trechos de código** que serão **executados** quando ocorrer um determinado evento.
- **Associar um código** a um evento é chamado de "**capturar**" um evento.
- Chamamos de "**tratamento**" de um evento à **execução do código** associado a ele.

Tipos de Eventos

- Na forma *client-side* de JavaScript os **eventos** se **originam de elementos** HTML (como botões, imagens etc.) sendo definidos como **atributos** destes elementos.
- Para permitir a **manipulação de evento**, as várias tags possíveis de responderem a eventos passaram a ter **atributos específicos**.
- O que distingue o tipo dos eventos é **principalmente os atributos** que eles "disparam".

Tipos de eventos e atributos

- **Cada elemento** de uma página HTML contém um determinado **conjunto de eventos** associados, por exemplo:

Eventos do mouse:

```
onClick
onDblClick
onMouseMove
onMouseOver
onMouseOut
onMouseDown
onMouseUp
```

Eventos do formulário:

```
onSubmit
onReset
```

Eventos do teclado:

```
onKeyDown
onKeyPress
onKeyUp
```

“tratamento” ou código associado:

- Usa-se o **nome** do atributo seguido de **“=“** e o **comando ou função** JavaScript a ser executada pelo evento.
- Pode ter **mais de um comando**.
- A um **mesmo elemento** HTML pode estar associado também **diversos eventos** e cada um disparar uma **ação diferente**.
- O conteúdo dos elementos da página podem ser acessados através de sua ordem no array correspondente ou pelo seu nome definido no atributo **name** ao ser gerado: **<FORM NAME=sai>** é equivamente a document.forms[0]

exemplo:

```
<HTML><HEAD><TITLE>Eventos do Mouse</TITLE>
<SCRIPT LANGUAGE="javascript">
nomeIm = new Array("pernalonga","patolino","tasmania");
function entrei ( n )
    {document.sai.texto.value = "esta em "+ nomeIm [ n ] ;}
function sai ( n )
    {document.sai.texto.value="saiu "+ nomeIm [ n ] ; }
function clica ( n )
    { document.sai.texto.value = "clicou " +nomeIm [ n ] ; }
function nd ( ) {     }
</SCRIPT></HEAD><BODY>Diversos Eventos do Mouse
```

como fica:

```
<A href="javascript : nd ( )" ,  onClick=" clica ( 0 ) ",
   onMouseOver="entrei (0)" , onMouseOut="sai (0)" >
   <IMG SRC="i1.gif" BORDER=0>  </A>

<A href="javascript : nd ( ) " , onClick=" clica (1) " ,
   onMouseOver="entrei (1)" , onMouseOut="sai (1) " > <IMG
   SRC="i2.gif" BORDER=0>  </A>

<A href="javascript : nd ( ) ",  onClick=" clica (2) " ,
   onMouseOver="entrei (2)" , onMouseOut="sai (2) " >
   <IMG SRC="i4.gif" BORDER=0>  </A>

<FORM NAME=sai><INPUT TYPE=TEXT NAME=texto>
</FORM></CENTER></BODY></HTML>
```

Clicar

como fica:

```
<A href="javascript : nd ( )" ,  onClick=" clica ( 0 ) ",
   onMouseOver="entrei (0)" , onMouseOut="sai (0)" >
   <IMG SRC="i1.gif" BORDER=0>  </A>

<A href="javascript : nd ( ) " , onClick=" clica (1) " ,
   onMouseOver="entrei (1)" , onMouseOut="sai (1) " > <IMG
   SRC="i2.gif" BORDER=0>  </A>

<A href="javascript : nd ( ) ",  onClick=" clica (2) " ,
   onMouseOver="entrei (2)" , onMouseOut="sai (2) " >
   <IMG SRC="i4.gif" BORDER=0>  </A>

<FORM NAME=sai><INPUT TYPE=TEXT NAME=texto>
</FORM></CENTER></BODY></HTML>
```

A Palavra Reservada `this`

- **Métodos** não são nada mais que **funções** JavaScript invocadas **através de um objeto**. Por isso eles têm uma propriedade muito importante: o **objeto que o chamou**.
- Este objeto permanece armazenado no valor da palavra-chave `this`.
- **`this`** serve para referenciar o **objeto corrente**.
- incluindo no exemplo anterior a linha:

```
function nd ( )
 { document.sai.texto.value=" this é : " + this }
```

A Palavra Reservada `this`

- **Métodos** não são nada mais que **funções** JavaScript invocadas **através de um objeto**. Por isso eles têm uma propriedade muito importante: o **objeto que o chamou**.
- Este objeto permanece armazenado no valor da palavra-chave `this`.
- **`this`** serve para referenciar o **objeto corrente**.
- incluindo no exemplo anterior a linha:

```
function nd ( )
 { document.sai.texto.value=" this é : " + this }
```

Clicar

Eventos como Propriedades do Objeto

- Embora a forma mais freqüente de manipular eventos seja defini-los como **atributos da *tag***, eles podem ser definidos explicitamente como funções a serem usadas como **propriedades de objetos** HTML.

- Veja os dois exemplos a seguir que produzem exatamente o mesmo resultado:

evento em tag:

```
...function nom ( )
    { document.f1.nome.value = document.f1.novo.value; }
    function end ( )
    { document.f1.mail.value = document.f1.novo.value; }
.....
<INPUT TYPE=BUTTON NAME="b1" Value="Troca nome"
    onClick= " nom ( ) ; " >
<INPUT TYPE=BUTTON NAME="b2" Value="Troca email"
    onClick= " end ( ) ; " >
.......
```

Clicar

evento em tag:

```
...function nom ( )
    { document.f1.nome.value = document.f1.novo.value; }
    function end ( )
    { document.f1.mail.value = document.f1.novo.value; }

.....
<INPUT TYPE=BUTTON NAME="b1" Value="Troca nome"
    onClick= " nom ( ) ; " >
<INPUT TYPE=BUTTON NAME="b2" Value="Troca email"
    onClick= " end ( ) ; " >

.......
```

evento como propriedade do objeto

```
.....
<INPUT TYPE=BUTTON NAME = "b1" Value = "Troca nome" >
<INPUT TYPE=BUTTON NAME = "b2" Value = "Troca email" >
</FORM><SCRIPT LANGUAGE = " javascript " >
document.f1.b1.onclick = function nom ( )
   {document.f1.nome.value=document.f1.novo.value;}
document.f1.b2.onclick = function end ( )
   {document.f1.mail.value=document.f1.novo.value;}
</SCRIPT>
```

Clicar

evento como propriedade do objeto

```
.....
<INPUT TYPE=BUTTON NAME = "b1" Value = "Troca nome" >
<INPUT TYPE=BUTTON NAME = "b2" Value = "Troca email" >
</FORM><SCRIPT LANGUAGE = " javascript " >
document.f1.b1.onclick = function nom ( )
   {document.f1.nome.value=document.f1.novo.value;}
document.f1.b2.onclick = function end ( )
   {document.f1.mail.value=document.f1.novo.value;}
</SCRIPT>
```

finalizando....

Nesta aula você viu como criar páginas que podem **interagir com o usuário**, através do uso de **eventos!**

Não se preocupe se incialmente **não ficar totalmente claro.** Este assunto é complexo e você só ficará seguro após ter sido um criador de **algumas páginas com um dos tipos de eventos**. Um dica é **reler esta aula sempre que precisar usar eventos.**

Na próxima aula falaremos da interação através dos formulários!