Agora iremos entender um pouco o que são e como funcionam os **eventos** em JavaScript.

**O que são eventos?**

Os eventos são basicamente um conjunto de ações que são realizadas em um determinado elemento da página web, seja ele um texto, uma imagem, ou uma div, por exemplo. Muitas das interações do usuário que está visitando sua página com o conteúdo do seu site podem ser consideradas eventos.

Existe uma infinidade de eventos definidos para uso em JavaScript, abaixo podemos ver alguns dos principais e mais utilizados:

|  |  |
| --- | --- |
| **onBlur** | remove o foco do elemento |
| **onChange** | muda o valor do elemento |
| **onClick** | o elemento é clicado pelo usuário |
| **onFocus** | o elemento é focado |
| **onKeyPress** | o usuário pressiona uma tecla sobre o elemento |
| **onLoad** | carrega o elemento por completo |
| **onMouseOver** | define ação quando o usuário passa o mouse sobre o elemento |
| **onMouseOut** | define ação quando o usuário retira o mouse sobre o elemento |
| **onSubmit** | define ação ao enviar um formulário |

**Como usar os eventos em JavaScript**

Existem diversas maneiras de se aplicar esses eventos aos elementos **HTML**, são elas:

* Inline
* Em um arquivo externo, usando um manipulador de eventos

**Evento Inline**

Vamos ver um exemplo de como usar um evento de maneira inline em JavaScript. Dessa forma é preciso que tudo seja definido diretamente na tag do elemento, dessa forma:

<!DOCTYPE html>

<html>

<head>

<title>Usando o evento onClick no Javascript</title>

</head>

<body>

<h1 onclick="this.innerHTML='Isso acontece quando usamos o evento onClick!'"

>Clique nesse link para testar o evento onClick!</h1>

</body>

</html>

**Listagem 1**. Eventos inline JavaScript

No exemplo anterior nós vimos de maneira inline como utilizar o evento onclick.

(obs.: O **innerHTML** é uma propriedade do Element que define ou retorna o conteúdo HTML de um elemento. Essa propriedade geralmente é escrita no DOM (Document Object Model), através do JavaScript.)

**Usando evento externo**

Para chamar um evento externo iremos precisar de um manipulador de eventos, nesse caso vamos usar o event listener que adiciona ou remove um evento sobre qualquer elemento. O Event Listener disponibiliza duas funções principais, são elas:

* addEvent - Adiciona uma função que será disparada quando ocorrer determinado evento no objeto.
* removeEvent - Remove um listener previamente adicionado em um objeto e retorna true em caso de sucesso.

Agora vamos ver um exemplo de como utilizar o Event Listener.

<script type="text/javascript" src="/path/to/event-listener.js"></script>

<form>

<input type="text" name="a" />

<input type="submit" />

</form>

<script type="text/javascript">

var f = document.forms[0];

addEvent(f, "submit", function( e ) { alert(f.a.value);return false; });

addEvent(f.a, "keyup", function( e ) { f.a.value=f.a.value.toUpperCase(); });

</script>

**Listagem 2**. Usando Event Listener

Dessa forma nosso formulário fica limpo e sem código JavaScript, sendo feita toda a validação de forma externa.

**Alguns exemplos**

No exemplo 1 vimos como utilizar o evento onClick, abaixo iremos ver alguns mais exemplos de eventos e como usá-los.

**onLoad**

O evento onload pode ser usado quando queremos que algo seja carregado junto com o carregamento da página.

<!DOCTYPE html>

<html>

<head>

<title>Usando eventos no Javascript</title>

</head>

<body onload="checkCookies()">

<script>

function checkCookies()

{

if (navigator.cookieEnabled==true)

{

alert("Cookies são permitidos")

}

else

{

alert("Cookies não são permitidos")

}

}

</script>

<p>Irá aparecer um alert dizendo se os cookies estão ou não liberados em seu navegador</p>

</body>

</html>

**Listagem 3**. usando evento onLoad

**onChange**

O evento onChange é utilizado para que seja realizada determinada ação após alguma mudança. No exemplo abaixo iremos fazer essa mudança acontecer ao clicarmos fora do input text.

<!DOCTYPE html>

<html>

<head>

<title>Usando eventos no Javascript</title>

<script>

function myFunction()

{

var x=document.getElementById("fname");

x.value=x.value.toUpperCase();

}

</script>

</head>

<body>

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>

</body>

</html>

**Listagem 4**. Evento onChange

**onMouseOver e onMouseOut**

Esses eventos são responsáveis pelas ações que o usuário faz com o mouse, passando o mouse no elemento e tirando o mouse do elemento, respectivamente.

<!DOCTYPE html>

<html>

<head>

<title>Usando eventos no Javascript</title>

</head>

<body>

<div onmouseover="mOver(this)" onmouseout="mOut(this)"

style="background-color:#D94A38;width:120px;height:20px;padding:40px;">Passe o mouse em mim</div>

<script>

function mOver(obj)

{

obj.innerHTML="Obrigado"

}

function mOut(obj)

{

obj.innerHTML="Passe o mouse em mim"

}

</script>

</body>

</html>

**Listagem 5**. Eventos onmouseover e onmouseout

**onmousedown, onmouseup e onclick**

Nesse exemplo iremos ver como trabalhar com o clique, o segurar o clique e o soltar do clique.

<!DOCTYPE html>

<html>

<head>

<title>Usando eventos no Javascript</title>

</head>

<body>

<div onmousedown="mDown(this)" onmouseup="mUp(this)"

style="background-color:#D94A38;width:90px;height:20px;padding:40px;">Clique aqui</div>

<script>

function mDown(obj)

{

obj.style.backgroundColor="#1ec5e5";

obj.innerHTML="Solte o clique"

}

function mUp(obj)

{

obj.style.backgroundColor="#D94A38";

obj.innerHTML="Obrigado"

}

</script>

</body>

</html>

**Listagem 6**. Eventos onmousedown, onmouseup e onclick

Fonte: https://www.devmedia.com.br/trabalhando-com-eventos-em-javascript/28521