

Introdução a Biblioteca jQuery: Sintaxe, Eventos e Efeitos

APRESENTAR OS RECURSOS BÁSICOS DA BIBLIOTECA JAVASCRIPT JQUERY, MOSTRANDO SUAS PRINCIPAIS FUNÇÕES E APLICAÇÃO NO DESENVOLVIMENTO DE INTERFACES DINÂMICAS.

AUTOR(A): PROF. EDSON MELO DE SOUZA

Introdução

A criação de interfaces dinâmicas é uma realidade e para que os sites e páginas atendam à demanda, é necessário utilizar recursos que favoreçam, não só a rapidez no desenvolvimento, mas também a qualidade final.

É comum os desenvolvedores de sistemas e de interfaces criarem seus próprios recursos baseados em tecnologias existentes como o AJAX (*Asynchronous JavaScript And XML*) que é um conjunto de recursos para o desenvolvimento de aplicações dinâmicas baseadas em intensa interação de usuários.

Por este motivo, existem diversas bibliotecas ou *frameworks* – conjunto de recursos para resolver um problema, que ajudam o desenvolvedor a criar, com qualidade profissional, páginas e sites de forma rápida e padronizada.



A Biblioteca jQuery

A jQuery é uma biblioteca de funções JavaScript que trabalha diretamente com o HTML, manipulando componentes na página, a fim de proporcionar mais dinamismo. Ela é composta por funções que facilitam o desenvolvimento e a aplicação de recursos, sem a necessidade de muita programação para realizar as tarefas. Seu principal objetivo é otimizar, de maneira fácil, a interpretação dos *scripts* utilizados em uma página.



Entre suas principais funcionalidades pode-se destacar:

- Abstrai a incompatibilidade de recursos entre navegadores, ou seja, possui mecanismos internos que aplicam os efeitos ou processamento desejado de forma transparente, independente do navegador.
- Elimina a redundância de código, minimizando a escrita.
- Através de *plug-ins*, reaproveita o código de maneira inteligente. Existem centenas desse recurso disponíveis para utilização.
- Trabalha com as tecnologias DOM e AJAX perfeitamente.
- Utiliza os recursos do CSS 1, 2 e 3 de forma segura e compatível.

Obtendo a jQuery

Existem diversas versões da biblioteca jQuery, entretanto é aconselhável que seja utilizada sempre a versão mais recente, atualmente a 3.2.1. A jQuery tem, além da sua versão completa (jquery.js), uma versão menor conhecida como jQuery minificada (jquery.min.js), entretanto, a sua funcionalidade é a mesma, variando apenas o tamanho do arquivo, pois a versão completa é inteligível para humanos, enquanto a outra não.

Para utilizar a biblioteca há duas formas, uma é realizando o *download* dos arquivos que a compõe e fazendo sua inclusão no arquivo HTML por meio da tag *script*. Desta forma, todos os arquivos ficam junto com suas páginas ou site. Entretanto há uma desvantagem, pois, quando houver alguma atualização, não refletirá imediatamente na sua aplicação, fazendo com que os novos recursos e correções de erros não estejam presentes para você.

Para incluir a biblioteca em seu projeto utilize a instrução

```
1. <script type="text/javascript" src="caminho_do_arquivo/jquery.js"></scri
```

A outra forma é fazendo a inclusão diretamente do site do Google, que mantém sempre a versão atualizada. Você pode fazer a inclusão no arquivo HTML da seguinte forma:

```
1. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.m
```

Apesar de manter atualizada a biblioteca quando a utiliza desta maneira, há uma desvantagem, pois, se o site do Google ficar fora do ar ou houver algum atraso no carregamento por lentidão ou outro problema, os recursos não estarão disponíveis.

Portanto, você deve decidir qual o modo que irá utilizar em seus projetos.

ATENÇÃO!!!

- Para nossos exemplos será adotada a importação direta do Google.
- Os comentários dos códigos estão incluídos diretamente nos arquivos, a fim de melhorar o entendimento.
- Ao final deste tópico, há um vídeo mostrando todos os exemplos apresentados, em execução.

Utilizando a jQuery

A jQuery é uma ferramenta excelente para manipular os objetos DOM que é uma interface de programação para documentos HTML, XML e SVG, permitindo manipular elementos gráficos na tela do navegador. Os objetos DOM são inicializados automaticamente no carregamento de uma página.

No exemplo 1 a seguir vamos ver como utilizar a jQuery, mostrando uma mensagem ao usuário no momento do carregamento da página:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Primeiro exemplo utilizando a jQuery</title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função que será executada quando a página for carrega
12.        $(document).ready(function(){
13.            // Mostra uma caixa de mensagem no navegador
14.            alert("Olá, seja bem vindo a utilização da jQuery
15.        })// fim da função
16.    </script>
17. </head>
18. <body>
19. </body>
20. </html>
```

Você pode perceber no código anterior foi utilizado o atributo *document*. A linha 12 tem a seguinte tradução:

Ao carregar o documento HTML, leia-o e execute o que está dentro da *function()*.

Sintaxe da jQuery

A jQuery traz um recurso sintático bastante fácil de compreender e é tratado como uma função descrita pelo símbolo `$()`. Isto é um atalho para a função `jQuery()` e, quando utilizada, informa ao interpretador do JavaScript que deverá ser acessado algum recurso da biblioteca jQuery. Todas as funções são executadas de acordo com os parâmetros que forem informados, ou seja, quando identificamos um objeto utilizando o identificador “id”, qualquer elemento que possuir o identificador será utilizado pela função jQuery. Sempre lembrando que cada elemento deve possuir um “id” único. Vejamos no exemplo 2 a utilização do “id” com jQuery.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função que será executada quando a página for carrega
12.        $(document).ready(function(){
13.            // A hashtag (#) diz que o objeto está sendo ide
14.            $("#mensagem").text("Eu sou uma mensagem!");
15.        })// fim da função
16.    </script>
17. </head>
18. <body>
19.     <!-- tag h1 com o (id) definido com o nome "mensagem" -->
20.     <h1 id="mensagem"></h1>
21. </body>
22. </html>
```

Também pode ser utilizada a *function()* como parâmetro da função jQuery “\$()”, veja o próximo exemplo que mostra, ao invés de um seletor, uma classe HTML:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função que será executada quando a página for carrega
12.        $(function(){
13.            // o ponto (.) informa que o objeto está identif
14.            $(".mensagem").text("Eu sou uma mensagem, agora
15.        })// fim da função
16.    </script>
17. </head>
18. <body>
19.     <h1 class="mensagem"></h1>
20. </body>
21. </html>
```

A jQuery identifica os elementos ou objetos, manipuláveis pelo JavaScript, em uma página utilizando sua função “\$()”. Os objetos, quando não identificados com um “id” usando o símbolo “#”, ou por uma “class” por meio do símbolo “.”, serão sempre identificados pelo seu *nome*. Vamos ver um exemplo para ficar bem claro:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função jQuery que será executada quando a página for
12.        $(function(){
13.            // Utilizando o nome do objeto (parágrafo) e não
14.            $("p").text("Eu sou uma mensagem, agora com uma
15.        })// fim da função
16.    </script>
17. </head>
18. <body>
19.     <p></p>
20.     <p></p>
21.     <p></p>
22.     <p></p>
23. </body>
24. </html>
```

Você pode perceber que foram colocados quatro parágrafos vazios, mas a jQuery identificou o elemento “p” (linha 14) e atribui o valor a todos eles.

No próximo exemplo é utilizado uma expansão dos objetos, ou seja, é possível identificar mais de um objeto ao mesmo tempo e que receberá o mesmo valor. Para isso, basta colocar entre vírgulas quais objetos se deseja selecionar. Vejamos:


```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função que será executada quando a página for carrega
12.        $(function(){
13.            // Utilizando o nome de múltiplos objetos
14.            $("p, h1, h2").text("Eu sou uma mensagem para di
15.        })// fim da função
16.
17.    </script>
18. </head>
19. <body>
20.     <h1></h1>
21.     <h2></h2>
22.     <p></p>
23. </body>
24. </html>
```

Manipulando Eventos

Apenas controlar identificadores, classes e objetos não é o suficiente para deixar uma página dinâmica. A jQuery fornece condições de manipularmos os eventos de uma página, ou seja, podemos controlar o que ocorre quando um botão é clicado “*onclick()*”, o foco de um objeto é perdido “*focusout*” ou então ele ganha o foco “*hover*”, etc.

Segundo Alvarez (2011), "Os eventos são um dos elementos mais importantes no desenvolvimento de aplicações web enriquecidas do lado do cliente (*Front-End*), posto que servem para realizar ações na página à medida que o usuário realiza coisas com a página. Ou seja, são a base para criar a interação com o usuário, algo tão importante nas páginas que usam jQuery.". Portanto, é fundamental dominar o conceito sobre eventos, pois é através deles que conseguimos implementar funcionalidades que geram as características de uma página dinâmica.

Uma página web, mesmo que estática, ainda assim apresenta um mínimo de interação, pois todo usuário executa, no mínimo, um clique na página. Pois bem, um clique é um evento, entretanto, há muito mais a ser explorado. A tabela a seguir apresenta todos os eventos atualizados disponíveis na jQuery.

Lista de Eventos jQuery	
Método / Propriedade	Descrição
<code>blur()</code> (https://www.w3schools.com/jquery/event_blur.asp)	Evento disparado quando um determinado elemento perde o foco.
<code>change()</code> (https://www.w3schools.com/jquery/event_change.asp)	Evento disparado quando o valor de um controle é alterado
<code>click()</code> (https://www.w3schools.com/jquery/event_click.asp)	Evento disparado quando um determinado controle recebe um clique do mouse ou, quando selecionado, um <i>enter</i> .
<code>dblclick()</code> (https://www.w3schools.com/jquery/event_dblclick.asp)	Evento disparado quando um determinado controle recebe um duplo clique do mouse.
<code>event.currentTarget</code> (https://www.w3schools.com/jquery/event_currenttarget.asp)	Identifica o destino atual para o evento.
<code>event.data</code> (https://www.w3schools.com/jquery/event_data.asp)	Contém dados opcionais passados para um método de evento quando o manipulador de execução atual está vinculado.
<code>event.delegateTarget</code> (https://www.w3schools.com/jquery/event_delegatetarget.asp)	Retorna o elemento onde o manipulador de eventos jQuery atualmente chamado foi anexado.
<code>event.isDefaultPrevented()</code> (https://www.w3schools.com/jquery/event_isdefaultprevented.asp)	Retorna se <code>event.preventDefault()</code> foi chamado para o objeto de evento.

event.isImmediatePropagationStopped() (https://www.w3schools.com/jquery/event_isimmediatepropagationstopped.a sp)	Retorna se event.stopImmediatePropagation() foi chamado para o objeto de evento.
event.isPropagationStopped() (https://www.w3schools.com/jquery/event_ispropagationstopped.asp)	Retorna se event.stopPropagation() foi chamado para o objeto de evento.
event.namespace (https://www.w3schools.com/jquery/event_namespace.asp)	Retorna o <i>namespace</i> especificado quando o evento foi disparado.
event.pageX (https://www.w3schools.com/jquery/event_pagex.asp)	Retorna a posição do mouse relativa à borda esquerda do documento.
event.pageY (https://www.w3schools.com/jquery/event_pagey.asp)	Retorna a posição do mouse em relação à borda superior do documento.
event.preventDefault() (https://www.w3schools.com/jquery/event_preventdefault.asp)	Impede a ação padrão do evento
event.relatedTarget (https://www.w3schools.com/jquery/event_relatedtarget.asp)	Retorna qual elemento está sobre o movimento do mouse.
event.result (https://www.w3schools.com/jquery/event_result.asp)	Contém o último / anterior valor retornado por um manipulador de eventos acionado pelo evento especificado.
event.stopImmediatePropagation() (https://www.w3schools.com/jquery/event_stopimmediatepropagation.asp)	Impede que outros manipuladores de eventos sejam chamados.
event.stopPropagation() (https://www.w3schools.com/jquery/event_stoppropagation.asp)	Impede que os manipuladores pai sejam notificados do evento.
event.target (https://www.w3schools.com/jquery/event_target.asp)	Retorna qual elemento DOM acionou o evento.

event.timeStamp (https://www.w3schools.com/jquery/event_timestamp.asp)	Retorna o número de milissegundos desde 1º de janeiro de 1970, quando o evento é acionado.
event.type (https://www.w3schools.com/jquery/event_type.asp)	Retorna qual tipo de evento foi acionado.
event.which (https://www.w3schools.com/jquery/event_which.asp)	Retorna qual tecla do teclado ou botão do mouse foi pressionado para o evento.
focus() (https://www.w3schools.com/jquery/event_focus.asp)	Evento disparado quando um determinado controle recebe o foco. É inverso ao evento blur().
focusin() (https://www.w3schools.com/jquery/event_focusin.asp)	Anexa um manipulador de eventos ao evento focusin().
focusout() (https://www.w3schools.com/jquery/event_focusout.asp)	Anexa um manipulador de eventos ao evento focusout().
hover() (https://www.w3schools.com/jquery/event_hover.asp)	Anexa dois manipuladores de eventos ao evento hover, que é quando o cursor do mouse está sobre um controle.
keydown() (https://www.w3schools.com/jquery/event_keydown.asp)	Ocorre quando uma tecla está pressionada.
keypress() (https://www.w3schools.com/jquery/event_keypress.asp)	Evento que ocorre quando a tecla está pressionada.
keyup() (https://www.w3schools.com/jquery/event_keyup.asp)	Evento que ocorre quando a tecla retorna a sua posição original no teclado.
mousedown() (https://www.w3schools.com/jquery/event_mousedown.asp)	Este evento ocorre quando o botão do mouse (qualquer um) está pressionado.

<code>mouseenter()</code> (https://www.w3schools.com/jquery/event_mouseenter.asp)	Evento disparado quando o ponteiro do mouse é apontado para um elemento.
<code>mouseleave()</code> (https://www.w3schools.com/jquery/event_mouseleave.asp)	Idêntico ao <code>mouseout()</code> .
<code>mousemove()</code> (https://www.w3schools.com/jquery/event_mousemove.asp)	Evento disparado quando o mouse é movimentado.
<code>mouseout()</code> (https://www.w3schools.com/jquery/event_mouseout.asp)	Evento disparado quando o cursor é removido de sobre um elemento.
<code>mouseover()</code> (https://www.w3schools.com/jquery/event_mouseover.asp)	Similar ao <code>mouseenter()</code> .
<code>mouseup()</code> (https://www.w3schools.com/jquery/event_mouseup.asp)	Evento disparado quando o botão do mouse é solto.
<code>off()</code> (https://www.w3schools.com/jquery/event_off.asp)	Remove os manipuladores de eventos anexados.
<code>on()</code> (https://www.w3schools.com/jquery/event_on.asp)	Anexa os manipuladores de eventos anexados.
<code>one()</code> (https://www.w3schools.com/jquery/event_one.asp)	Adiciona um ou mais manipuladores de eventos aos elementos selecionados. Esse manipulador só pode ser acionado uma vez por element.
<code>\$.proxy()</code> (https://www.w3schools.com/jquery/event_proxy.asp)	Toma uma função existente e retorna uma nova com um contexto específico.
<code>ready()</code> (https://www.w3schools.com/jquery/event_ready.asp)	Especifica uma função a ser executada quando o DOM está totalmente carregado.
<code>resize()</code> (https://www.w3schools.com/jquery/event_resize.asp)	Anexa / ativa o evento de redimensionamento.
<code>scroll()</code> (https://www.w3schools.com/jquery/event_scroll.asp)	Anexa / ativa o evento de rolagem.

<code>select()</code> (https://www.w3schools.com/jquery/event_select.asp)	Evento disparado quando um determinado controle é selecionado, ou seja, controles que permitem a entrada de dados.
<code>submit()</code> (https://www.w3schools.com/jquery/event_submit.asp)	Evento disparado quando um determinado controle é acionado dentro de um formulário.
<code>trigger()</code> (https://www.w3schools.com/jquery/event_trigger.asp)	Ativa todos os eventos vinculados aos elementos selecionados.
<code>triggerHandler()</code> (https://www.w3schools.com/jquery/event_triggerhandler.asp)	Dispara todas as funções ligadas a um evento especificado para os elementos selecionados.
Fonte: Adaptado de W3Schools (https://www.w3schools.com/jquery/jquery_ref_events.asp)	

Como você pode perceber, existem muitos eventos que podem ser manipulados para criarmos efeitos e interatividade em uma página Web.

A seguir vamos ver um exemplo de uma página que possui uma caixa de texto e, ao clicar na caixa, será mostrada uma mensagem orientando o preenchimento. Quando for dado um clique fora da caixa, será mostrada outra mensagem, em uma caixa de mensagem, informando que saiu da caixa de texto.

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
5.
6.         <!-- Descrição de uma folha de estilo -->
7.         <style type="text/css">
8.
9.             /* Cria um retângulo verde claro na parte superior da tela */
10.            #div_controle_mouse{
11.                width: 600px;
12.                height: 100px;
13.                background: #5F9EA0;
14.                position: absolute;
15.                top: 15%;
16.                left: 30%;
17.                margin-top: -50px;
18.                margin-left: -50px;
19.            }
20.
21.            /* Cria um estilo para o retângulo azul abaixo na tela */
22.            #div_outros_eventos_mouse{
23.                width: 600px;
24.                height: 100px;
25.                background: blue;
26.                position: absolute;
27.                top: 2%;
28.                left: 30%;
29.                margin-top: 200px;
30.                margin-left: -50px;
31.            }
32.
33.            /* Cria um estilo para a div do formulário */
34.            #div_formulario{
35.                width: 580px;
36.                height: 120px;
37.                border: solid 2px;
38.                padding-top: 20px;
39.                padding-left: 20px;
```

```
40.         position: absolute;
41.         top: 25%;
42.         left: 30%;
43.         margin-top: 200px;
44.         margin-left: -50px;
45.     }
46.
47.     /* Apresenta a mensagem gerada pelo jQuery na parte de cima
48.     #mensagens{
49.         text-align: center;
50.         font-size: 30px;
51.     }
52. </style>
53.
54. <!-- Criação do bloco script para o tratamento dos eventos -->
55. <script type="text/javascript">
56.     $(document).ready(function () {
57.
58.         // mouseover()
59.         $("#div_controle_mouse").mouseover(function () {
60.             alert("O cursor foi posicionado sobre a div - mouse
61.         });
62.
63.         // mouseout()
64.         $("#div_controle_mouse").mouseout(function () {
65.             $("#mensagens").text("O cursor saiu da div - mouseou
66.         });
67.
68.         // mousemove()
69.         $("#div_controle_mouse").mousemove(function () {
70.             $("#mensagens").text("O mouse está dentro dentro da
71.         });
72.
73.         // mousedown()
74.         $("#div_controle_mouse").mousedown(function () {
75.             alert("Um botão do mouse foi pressionado - mousedown
76.         });
77.
78.     /*
```



```
79.         Outros eventos do mouse
80.         */
81.
82.         // click()
83.         $("#div_outros_eventos_mouse").click(function () {
84.             $("#mensagens").text("Você clicou uma vez no retângulo");
85.         });
86.
87.         // dblclick()
88.         $("#div_outros_eventos_mouse").dblclick(function () {
89.             $("#mensagens").text("Você clicou duas vezes no retângulo");
90.         });
91.
92.         // hover()
93.         $("#div_outros_eventos_mouse").hover(
94.             function () {
95.                 $(this).css("background", "orange");
96.                 $("#mensagens").text("Você entrou no retângulo");
97.             },
98.             function () {
99.                 $(this).css("background", "blue");
100.                $("#mensagens").text("Você saiu do retângulo");
101.            }
102.        );
103.
104.        /*
105.        Tratando eventos em um formulário
106.        */
107.
108.        // blur()
109.        $("#nome").blur(function () {
110.            $("#mensagens").text("O input perdeu o foco.");
111.        });
112.
113.        // change()
114.        $("#sexo").change(function () {
115.            $("#mensagens").text("O valor do sexo foi modificado");
116.        });
117.
```

```
18.         // focus()
19.         $("#nome").focus(function () {
20.             $("#mensagens").text("0 input recebeu o foco.");
21.         });
22.
23.         // select()
24.         $("#nome").select(function () {
25.             $("#mensagens").text("Texto selecionado.");
26.         });
27.
28.         // submit()
29.         $("#formulario").submit(function () {
30.             alert("0 formulario foi enviado.");
31.         });
32.
33.     });
34. </script>
35. </head>
36. <body>
37.     <!-- Cria uma tag span para mostrar as mensagens -->
38.     <div id="mensagens"></div>
39.
40.     <!-- Criação de uma tag div para gerar o retângulo-->
41.     <div id="div_controle_mouse"></div>
42.
43.     <!-- Criação de uma tag div para testar os eventos do mouse-->
44.     <div id="div_outros_eventos_mouse"></div>
45.
46.     <!-- Criação de uma tag div para colocar o formulário dentro -->
47.     <div id="div_formulario">
48.         <form id="formulario">
49.             Digite seu nome:
50.             <input type="text" id="nome"/><br><br>
51.             Selecione o sexo:
52.             <select id="sexo">
53.                 <option value="0" selected disabled>Selecione o Sexo
54.                 <option value="1">Masculino</option>
55.                 <option value="2">Feminino</option>
56.             </select><br><br>
```

```
57.         <input type="submit" value="Enviar" />
58.     </form>
59. </div>
60. </body>
61. </html>
```

No próximo exemplo, criaremos um link e, quando clicarmos nele, mostrará uma mensagem em uma tag *span*.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Indica que será incluído um script no código -->
10.    <script type="text/javascript">
11.        // Função que será executada quando o objeto (a) for cli
12.        $(function(){
13.            // Utilizando o nome do objeto (a)
14.            $("a").click(function(){
15.                $("#mensagem").text("Você clicou no link
16.                });
17.            })// fim da função
18.    </script>
19. </head>
20. <body>
21.     <h1>Mais um exemplo de eventos em jQuery</h1>
22.     <!-- link vazio -->
23.
24.     <a href="#">Clique Aqui</a>
25.     <span id="mensagem"></span>
26. </body>
27. </html>
```

No próximo exemplo vamos mostrar um comparativo entre a utilização do JavaScript puro e a jQuery. Utilizaremos o mesmo exemplo anterior, adicionando a parte JavaScript.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title></title>
5.
6.     <!-- Inclusão da biblioteca jQuery -->
7.     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
8.
9.     <!-- Criação do JavaScript para, quando clicar no link, mostrar
10.    <script type="text/javascript">
11.        function mensagem(){
12.            alert("Olá, esta mensagem foi gerada pelo JavaSc
13.        }
14.    </script>
15.
16.    <!-- Indica que será incluído um script no código utilizando a j
17.    <script type="text/javascript">
18.        // Função que será executada quando o objeto (a) for cli
19.        $(function(){
20.            // Utilizando o nome do objeto (a)
21.            $("a").click(function(){
22.                alert("Você clicou no link utilizando jQ
23.            });
24.        })// fim da função
25.    </script>
26.
27. </head>
28. <body>
29.     <h1>Comparação javaScript <em>versus</em> jQuery</h1>
30.
31.     <!-- link vazio com JavaScript-->
32.     <a href="#" onclick="mensagem();">Clique Aqui (Eu sou JavaScript
33.     <br><br>
34.
35.     <!-- link vazio com jQuery-->
36.     <a href="#">Clique Aqui (Eu sou jQuery)</a>
37. </body>
38. </html>
```

Porque quando clicamos no primeiro link que está utilizando JavaScript puro são exibidas duas caixas de mensagem? Por que, quando realizamos o clique, a jQuery interpreta o objeto “a” e, portanto, executa as duas instruções.

Você pode perceber nitidamente a diferença entre as duas formas de apresentar a mensagem na tela. Quando o JavaScript é utilizado se faz necessário indicar dentro do link que haverá uma ação, ou seja, deverá ser executada uma função relacionada ao evento *onclick()*. Já, quando se utiliza a jQuery, o evento é definido dentro do bloco da função *jQuery()*.

Eventos Mais Utilizados

O código a seguir mostra os eventos mais comuns utilizados em páginas Web, abordando os eventos do *mouse* e de manipulação de *formulários*.

É importante deixar claro que os efeitos podem ser mesclados, conseguindo assim mais recursos para implementar funcionalidades nas páginas.

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title></title>
5.         <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
6.
7.         <!-- Descrição de uma folha de estilo -->
8.         <style type="text/css">
9.
10.             /* Cria um retângulo verde claro na parte superior da tela */
11.             #div_controle_mouse{
12.                 width: 600px;
13.                 height: 200px;
14.                 background: #5F9EA0;
15.                 position: absolute;
16.                 top: 15%;
17.                 left: 30%;
18.                 margin-top: -50px;
19.                 margin-left: -50px;
20.             }
21.
22.             /* Cria um estilo para o retângulo azul abaixo na tela */
23.             #div_outros_eventos_mouse{
24.                 width: 600px;
25.                 height: 200px;
26.                 background: blue;
27.                 position: absolute;
28.                 top: 15%;
29.                 left: 30%;
30.                 margin-top: 200px;
31.                 margin-left: -50px;
32.             }
33.
34.             /* Apresenta a mensagem gerada pelo jQuery na parte de cima
35.             #mensagens{
36.                 text-align: center;
37.                 font-size: 30px;
38.             }
39.         </style>
```

```
40.
41.     <!-- Criação do bloco script para o tratamento dos eventos -->
42.     <script type="text/javascript">
43.         $(document).ready(function () {
44.
45.             // mouseover()
46.             $("#div_controle_mouse").mouseover(function () {
47.                 alert("O cursor foi posicionado sobre a div - mouse
48.             });
49.
50.             // mouseout()
51.             $("#div_controle_mouse").mouseout(function () {
52.                 $("#mensagens").text("O cursor saiu da div - mouseou
53.             });
54.
55.             // mousemove()
56.             $("#div_controle_mouse").mousemove(function () {
57.                 $("#mensagens").text("O mouse está dentro dentro da
58.             });
59.
60.             // mousedown()
61.             $("#div_controle_mouse").mousedown(function () {
62.                 alert("Um botão do mouse foi pressionado - mousedown
63.             });
64.
65.             /*
66.             Outros eventos do mouse
67.             */
68.
69.             // click()
70.             $("#div_outros_eventos_mouse").click(function () {
71.                 $("#mensagens").text("Você clicou uma vez no retângu
72.             });
73.
74.             // dblclick()
75.             $("#div_outros_eventos_mouse").dblclick(function () {
76.                 $("#mensagens").text("Você clicou duas vezes no retâ
77.             });
78.
```



```
79.          // hover()
80.          $("#div_outros_eventos_mouse").hover(
81.              function () {
82.                  $(this).css("background", "orange");
83.                  $("#mensagens").text("Você entrou no retângulo");
84.              },
85.              function () {
86.                  $(this).css("background", "blue");
87.                  $("#mensagens").text("Você saiu do retângulo");
88.              }
89.          );
90.
91.      });
92.  </script>
93. </head>
94. <body>
95.     <!-- Cria uma tag span para mostrar as mensagens -->
96.     <div id="mensagens"></div>
97.
98.     <!-- Criação de uma tag div para gerar o retângulo-->
99.     <div id="div_controle_mouse"></div>
100.
101.     <!-- Criação de uma tag div para testar os eventos do mouse-->
102.     <div id="div_outros_eventos_mouse"></div>
103. </body>
104. </html>
```

Efeitos com jQuery

Além da manipulação dinâmica de objetos, a jQuery permite aplicar efeitos interessantes para criar animações. Vejamos no exemplo a seguir como criar um efeito *Fade-in* em uma “div”.

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title></title>
5.         <!-- Inclusão da biblioteca jQuery -->
6.         <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/
7.
8.         <script type="text/javascript">
9.             $(function () {
10.                 $("#btnTeste").click(function () {
11.
12.                     // Sem parâmetros: o efeito é executado em 400ms
13.                     $("#minhaDivUp").slideUp();
14.
15.                     // Parâmetro com a duração em ms do efeito
16.                     $("#minhaDivUp").slideUp(1000);
17.
18.                     // Parâmetro slow: o efeito é executado em 600ms
19.                     $("#minhaDivUp").slideUp("slow");
20.
21.                     // Parâmetro com função: o efeito é executado em 200ms

```