

Desenvolvimento Web II

Aula 07 - jQuery (Parte 1)

Apresentação

jQuery é uma biblioteca JavaScript gratuita e de código fonte aberto desenvolvida para simplificar a programação do lado do cliente de aplicações Web. Sua API contempla uma série de recursos como, por exemplo, funções para navegar e modificar elementos DOM, funções para criar animações e efeitos, funções para registrar e tratar eventos e funções para desenvolver aplicações AJAX. Nesta aula e na próxima, nosso foco será estudar a API de baixo nível do jQuery. Já em aula posterior, você conhecerá alguns elementos de alto nível da biblioteca jQuery (widgets e outros componentes).

Para adicionar o JQuery em uma página HTML ou JSP você precisa adicionar, preferencialmente dentro da tag “head”, uma tag “script” referenciando o arquivo javascript remoto ou no seu computador.

Bons estudos!



Vídeo 01 - Apresentação

Objetivos

- Selecionar elementos da página utilizando jQuery.
- Registrar e tratar eventos utilizando jQuery.
- Descrever o código HTML da página por meio do jQuery.

Utilizando jQuery

Assim como qualquer outra biblioteca JavaScript, para utilizar a biblioteca jQuery basta incluí-la na sua página. Para tal, é necessário utilizar a tag 'script' e definir o atributo 'src', o qual pode tanto apontar para o arquivo do jQuery no seu próprio servidor como também para qualquer outro endereço na web que contenha a biblioteca.

Para testar os comandos do JQuery você precisa os digitar dentro de algum código Javascript (aberto com a tag script anteriormente).

Exemplo: apontando para a biblioteca do próprio site do jQuery.



Vídeo 02 - Introdução

```
1 <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js">
2 </script>
```

Para referenciar a biblioteca no próprio servidor, faça o download do arquivo a partir da página oficial do jQuery e salve-o no seu servidor.

Seletores

A sintaxe do jQuery é bastante simples e se resume a selecionar um elemento ou um grupo de elementos e realizar uma ou mais operações. Neste tópico, nosso foco será apenas em como selecionar elementos. Nos tópicos seguintes, veremos como realizar operações nos elementos selecionados.

Por padrão, jQuery define implicitamente os variáveis '\$' e 'jQuery'. Veja isso nos seguintes exemplos, onde Seletor representa o elemento a ser selecionado e acao() a operação a ser realizada no Seletor:

```
1 // Ex. 1
2 var x = $(Seletor);
3 x.acao();
4 // Ex. 2
5 $(Seletor).acao();
6 // Ex. 3
7 var y = jQuery(Seletor);
8 y.acao();
9
10 // Ex. 4
11 jQuery(Seletor).acao();
```

Veja agora outros exemplos, utilizando as duas variáveis \$ e jQuery para selecionar e modificar o texto do elemento HTML cujo id é 'campoXYZ':

```
1 // Ex. 1
2 jQuery("#campoXYZ").text("Aprendendo jQuery");
3 // Ex. 2
4 $("#campoXYZ").text("Aprendendo jQuery");
```

Notas

- A função "text;" utilizada acima serve para modificar o texto contido no elemento selecionado (será vista com mais detalhes nas seções a seguir).
- Chamamos de seletor o texto entre parênteses utilizado para especificar os elementos a serem selecionados.
- Perceba a similaridade desse seletor com o comando JavaScript document.getElementById().

Seletores Básicos

Pois bem, vamos definir melhor o que são os seletores usados no jQuery. Você irá notar que alguns seletores representam elementos específicos e únicos da página, já outros representam um conjunto de elementos.

Por exemplo, você pode, por meio do seletor a seguir, selecionar todos os elementos da página HTML:

```
1 $("*")
```

Por outro lado, se quisermos selecionar apenas os elementos de um determinado tipo, basta passar o nome do tipo entre aspas, como em:

```
1 $("tipo_do_elemento")
```

Veja isto aplicado a um exemplo concreto, onde queremos selecionar todos os elementos do tipo 'div' (tag <div> do HTML):

```
1 $("div")
```

Caso você queira selecionar um elemento específico, poderá fazê-lo por meio de seu id, passado entre aspas e iniciado pelo caractere '#':

```
1 $("#id_do_elemento")
```

Você pode também selecionar elementos pela sua classe. Para isso, dentro do seletor, utilize o caractere ponto (.) seguido pelo nome da classe (atributo 'class' da tag HTML). Por exemplo, para selecionar todos os elementos cuja classe é 'classeA', basta utilizar o seletor '.classeA', como mostrado a seguir:

```
1 $(".classeA")
```

Observe que é possível misturar os exemplos acima para compor um só seletor. Por exemplo, para selecionar todos os parágrafos (tag <p>) cuja classe seja 'classeA' (<p class='classeA'>), utilizamos o seguinte:

```
1 $("p.classeA").text("Combinando seletores!");
```

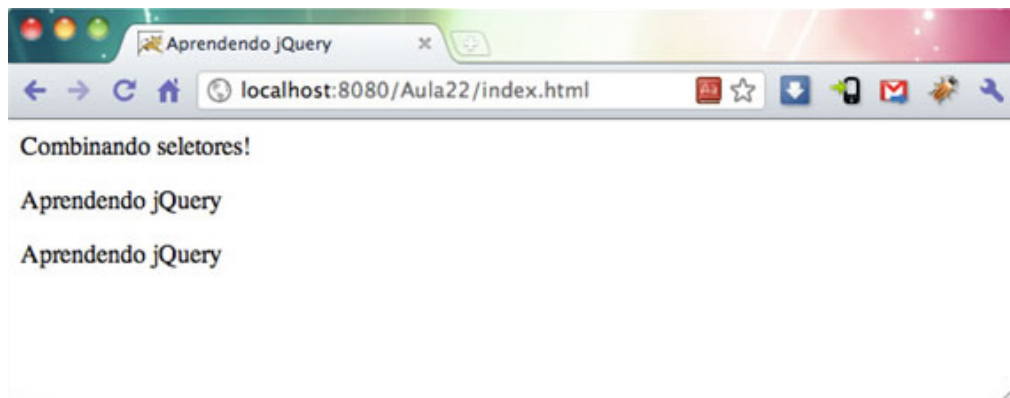
Veja, agora, um exemplo de código HTML completo que faz uso do jQuery para mudar o texto mostrado na tela:

```
1 <html>
2   <head>
3     <title>Aprendendo jQuery</title>
4     <script type="text/javascript" src="http://code.jquery.com/jquery-1.4.2.js"></script>
5     <script type="text/javascript">
6       $(document).ready(
7         function()
8         {
9           $("p.classeA").text("Combinando seletores!");
10        });
11    </script>
12  </head>
13
14  <body>
15    <p class="classeA">Aprendendo jQuery</p>
16    <p class="classeB">Aprendendo jQuery</p>
17    <p class="classeC">Aprendendo jQuery</p>
18  </body>
19 </html>
```

Algumas observações sobre esse código:

- A função `$(document).ready()` (linha 7) informa ao navegador que, após carregar toda página HTML, a função passada como parâmetro deve ser invocada. Note que essa função produz o mesmo resultado que especificar o atributo 'onload' da tag 'body' (veja nas aulas anteriores).
- A função 'text()' (linha 10) define o texto do (s) elemento (s) selecionado (s).

Ao acessar esse código por um navegador web, você observará o seguinte resultado:



Atividade 01

1. Adicione um botão no exemplo mostrado e escreva um código para que o texto do parágrafo de classe "classeB" seja modificado para "Alterado pelo clique do botão".
2. Adicione ao exemplo um segundo botão com o texto "Clique aqui" e faça com que ele, ao ser clicado, troque o seu próprio texto para "Já clicou".

Seletores baseados em atributos

Nesta seção, veremos como selecionar elementos de acordo com os atributos. Por exemplo, será possível selecionar todas as imagens (tag) cujo atributo 'src' terminem com ".jpg".

Para uma referência completa dos seletores baseados em atributos, dirija-se a: <http://api.jquery.com/category/selectors/attribute-selectors/>. Acesso em: 04 de março de 2015

A sintaxe para trabalharmos com atributos é informar dentro de chaves ([...]) o atributo, o operador (de igualdade, por exemplo) e o valor:

1	<code>\$([atributo operador valor])</code>
---	--

Veja, a seguir, alguns exemplos comentados:

```

1 / Modificando o valor de todos os elementos que possuam o atributo 'type'
2 // com valor igual a 'text'
3 $("[type='text']").val("test");
4 // Modificando o valor de todos os elementos que possuam o atributo 'name'
5 // com valor igual a 'teste1'
6
7 $("[name='teste1']").val("novo valor ...");

```

Para montar essas instruções, você poderá fazer uso dos seguintes operadores:

Operador	Significado
=	O valor informado precisa ser igual ao valor do atributo.
!=	O valor informado precisa ser diferente do valor do atributo.
^=	O valor do atributo precisa começar ou ser igual ao valor informado.
\$=	O valor do atributo precisa terminar ou ser igual ao valor informado.
~=	O valor do atributo precisa conter o valor informado.

Veja um exemplo concreto, que seleciona todos os elementos cujo atributo 'name' é igual 'campoDeTestes2':

```

1 $("[name='campoDeTestes2']")

```

Novamente, os seletores recém apresentados também podem ser utilizados de maneira composta com os demais seletores. Veja isso no exemplo a seguir, onde:

- A primeira linha recupera e armazena em "value" o valor do elemento cujo atributo 'name' seja igual a 'campoDeTestes2'.
- A segunda linha define o valor de todos os elementos do tipo 'input', cujo atributo 'name' seja igual a 'campoDeTestes2', como sendo igual ao valor da variável value.

```

1 var value = $("[name='campoDeTestes2']").val();
2 $("input[type='text'][name='campoDeTestes2']").val(value);

```


Eventos

Nesta seção, veremos como utilizar o jQuery para registrar e tratar determinados tipos de eventos. Veremos como tratar os eventos mais relevantes, mas, para uma referência completa, leia a documentação relativa ao registro e tratamento de eventos do jQuery (<http://api.jquery.com/category/events/>).



Vídeo 03 - Funções e Eventos

A tabela a seguir apresenta os principais métodos utilizados para se associar uma função (passada como parâmetro) a um evento.

Método	Descrição
<code>\$(document).ready(função)</code>	Define uma função a ser chamada quando a página estiver pronta, ou seja, depois de ter sido completamente carregada pelo navegador.
<code>\$(seletor).click(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) selecionado (s) receber (em) um clique.
<code>\$(seletor).dblclick(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) selecionado (s) receber (em) um clique duplo.
<code>\$(seletor).change(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) selecionado (s) for (em) modificado (s).
<code>\$(seletor).focus(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) informado (s) ganha (rem) foco.

Método	Descrição
<code>\$(seletor).focusin(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) informado (s) ou algum elemento dentro do elemento informado ganhar foco.
<code>\$(seletor).focusout(função)</code>	Define uma função a ser chamada quando o (s) elemento (s) informado (s) ou algum elemento dentro do elemento informado perder foco.

Tabela 1 - Principais métodos para associar funções a eventos.

Para exemplificar o uso dos métodos listados acima, vejamos os códigos abaixo que transmitem os dados de livros:

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4
5 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
6   "http://www.w3.org/TR/html4/loose.dtd">
7
8 <html>
9   <head>
10    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
11    <title>JSON</title>
12  </head>
13  <body onload="onLoad()">
14
15    <script type="text/javascript">
16
17      var jsonObj = null;
18      var livro = null;
19
20      function onLoad(){
21
22        var ajaxReq = new XMLHttpRequest();
23        var url = "/Aula07/livros.json";
24
25        ajaxReq.onreadystatechange = function() {
26          if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
27            jsonObj = eval('(' + ajaxReq.responseText + ')');
28            carregarLista();
29          }
30        };
31
32        ajaxReq.open("GET", url, true);
33        ajaxReq.send();
34
35      }
36
37      function carregarLista(){
38        var livros = jsonObj.livros.livro;
39        for ( var i = 0; i < livros.length; i++) {
40          var livro = livros[i];
41          var campoLivro = document.getElementById("livros");
42          campoLivro.options[campoLivro.length] = new Option(livro.nome, "" + i);
43        }
44      }
45
46      function carregarLivro(indice){
47
48        var livros = jsonObj.livros.livro;
49        var livro = livros[indice];
50
51        document.getElementById("nome").innerHTML = livro.nome;
```

```

52 document.getElementById("isbn").innerHTML = livro.isbn;
53
54 document.getElementById("editora").innerHTML = livro.editora;
55 document.getElementById("edicao").innerHTML = livro.edicao;
56 document.getElementById("preco").innerHTML = livro.preco;
57
58 var autoresStr = "";
59 var autores = livro.autores.autor;
60 for ( var i = 0; i < autores.length; i++) {
61     if( i > 0 ){
62         autoresStr += " | ";
63     }
64     autoresStr += autores[i];
65 }
66 document.getElementById("autores").innerHTML = autoresStr;
67 document.getElementById("sinopse").innerHTML = livro.sinopse;
68
69 var atributos = document.getElementById("divLivroSelecioneado").attributes;
70 atributos.getNamedItem("style").nodeValue="";
71 }
72
73
74 </script>
75 <br>
76 <select id="livros" size="2" onclick="carregarLivro(this.value)">
77 </select>
78 <br>
79
80 <div id="divLivroSelecioneado" style="visibility: hidden;"><br>
81     <table border="1" width="60%">
82         <tr>
83             <td width="20%">Nome:</td>
84             <td id="nome"></td>
85         </tr>
86         <tr>
87             <td>ISBN:</td>
88             <td id="isbn"></td>
89         </tr>
90         <tr>
91             <td>Editora:</td>
92             <td id="editora"></td>
93         </tr>
94         <tr>
95             <td>Edição:</td>
96             <td id="edicao"></td>
97         </tr>
98         <tr>
99             <td>Autor(es):</td>
100             <td id="autores"></td>
101         </tr>
102         <tr>

```

```
103     <td>Preço:</td>
104     <td id="preco"></td>
105 </tr>
106 <tr>
107     <td colspan="2">Sinopse:</td>
108 </tr>
109 <tr height="100">
110     <td colspan="2" id="sinopse"></td>
111 </tr>
112 </table>
113 </div>
114
115 </body>
116 </html>
```

Para utilizar o jQuery, podemos criar um novo arquivo JSP tanto com código HTML com o Javascript. Repare que na linha que declara a variável *url* (var url = "/Aula07/livros.json"); você deve trocar a string Aula07 pela que representa o seu projeto. O arquivo se chamará json_jQuery.jsp e terá o seguinte conteúdo:

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4
5 <html>
6   <head>
7     <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
8     <title>JSON+jQuery</title>
9   </head>
10  <body>
11
12    <script type="text/javascript">
13
14      $(document).ready(onLoad);
15
16      var jsonObj = null;
17      var livro = null;
18
19      function onLoad() {
20
21        $("#livros").click(carregarLivro);
22
23        var ajaxReq = new XMLHttpRequest();
24        var url = "/Aula07/livros.json";
25
26        ajaxReq.onreadystatechange = function() {
27          if (ajaxReq.readyState == 4 && ajaxReq.status == 200) {
28            jsonObj = eval('(' + ajaxReq.responseText + ')');
29            carregarLista();
30          }
31        };
32
33        ajaxReq.open("GET", url, true);
34        ajaxReq.send();
35
36      }
37
38      function carregarLista() {
39
40        var livros = jsonObj.livros.livro;
41        console.log(livros);
42        for ( var i = 0; i < livros.length; i++) {
43          var livro = livros[i];
44          console.log(livro);
45          var campoLivro = document.getElementById("livros");
46          alert(livro);
47          campoLivro.options[campoLivro.length] = new Option(livro.nome, "" + i);
48        }
49      }
50
51      //function carregarLivro(indice) {

```

```

52 function carregarLivro() {
53
54     var indice = $("#livros").val();
55     var livros = jsonObj.livros.livro;
56     var livro = livros[indice];
57
58     $("#nome").html(livro.nome);
59     $("#isbn").html(livro.isbn);
60
61     $("#editora").html(livro.editora);
62     $("#edicao").html(livro.edicao);
63     $("#preco").html(livro.preco);
64
65     var autoresStr = "";
66     var autores = livro.autores.autor;
67     for ( var i = 0; i < autores.length; i++) {
68         if (i > 0) {
69             autoresStr += " | ";
70         }
71         autoresStr += autores[i];
72     }
73     $("#autores").html(autoresStr);
74     $("#sinopse").html(livro.sinopse);
75
76     var atributos = document.getElementById("divLivroSelecionado").attributes;
77     atributos.getItemNamedItem("style").nodeValue = "";
78 }
79 </script>
80 <br>
81 <select id="livros" size="2">
82 </select>
83 <br>
84
85 <div id="divLivroSelecionado" style="visibility: hidden;"><br>
86     <table border="1" width="60%">
87         <tr>
88             <td width="20%">Nome:</td>
89             <td id="nome"></td>
90         </tr>
91         <tr>
92             <td>ISBN:</td>
93             <td id="isbn"></td>
94         </tr>
95         <tr>
96             <td>Editora:</td>
97             <td id="editora"></td>
98         </tr>
99         <tr>
100             <td>Edição:</td>
101             <td id="edicao"></td>
102         </tr>

```

```
103     <tr>
104         <td>Autor(es):</td>
105         <td id="autores"></td>
106     </tr>
107     <tr>
108         <td>Preço:</td>
109         <td id="preco"></td>
110     </tr>
111     <tr>
112         <td colspan="2">Sinopse:</td>
113     </tr>
114     <tr height="100">
115         <td colspan="2" id="sinopse"></td>
116     </tr>
117 </table>
118 </div>
119
120 </body>
121 </html>
```

Os livros que serão carregados vem de um arquivo JSON que deve ser criado no seu projeto com o nome “livros.json” com o seguinte conteúdo:


```

1 {
2   "livros": {
3     "livro": [
4       {
5         "nome": "A Cabana",
6         "isbn": "8599296361",
7         "editora": "SEXTANTE FICÇÃO",
8         "edicao": "1ª Edição - 2008 ",
9         "autores": { "autor": "YOUNG, WILLIAM P." },
10        "preco": "12,45",
11        "sinopse": "A filha mais nova de Mackenzie Allen Philip foi raptada durante as férias em família",
12      },
13      {
14        "nome": "A Ilha Sob o Mar",
15        "isbn": "8528614441",
16        "editora": "BERTRAND BRASIL",
17        "edicao": "1ª Edição - 2010",
18        "autores": {
19          "autor": [
20            "ALLENDE, ISABEL",
21            "SEGUNDO AUTOR"
22          ]
23        },
24        "preco": "12,45",
25        "sinopse": "O romance narra a vida de Zarité, a escrava que foi vendida aos nove anos de idade",
26      }
27    ]
28  }
29 }

```

Notas

1. Observe que na linha 9 do código modificado, não utilizamos o atributo 'onload' da tag 'body': ao invés de definir o atributo 'onload', na linha 14 do código modificado, utilizamos jQuery para definir uma função a ser chamada quando a pagina estiver completamente carregada.
2. Na linha 21 do código modificado, utilizamos jQuery para registrar uma função a ser chamada, caso a lista de livros seja clicada.

3. Na linha 21, experimente trocar o método 'click' por 'change' e observe a seguinte diferença: ao mudar o livro selecionado utilizando o teclado (apertando a tecla 'subir' ou 'descer') ao invés do mouse, utilizando o método 'click' nada ocorrerá. Já utilizando o método 'change', a tabela de detalhes do livro será atualizada.
4. A partir da linha 54, podemos ver exemplos de seleção de elementos pelo id.
5. Os métodos 'val' e 'html' (linhas 54 e 58, respectivamente) utilizados no exemplo modificado serão explicados em detalhe nas próximas seções desta aula.
6. O arquivo modificado ainda não está utilizando jQuery em sua plenitude. À medida que formos aprendendo novos recursos, iremos atualizando o exemplo acima.

Métodos – HTML

Neste momento da aula, veremos os métodos da API do jQuery cuja finalidade é inserir, apagar ou modificar a estrutura HTML da página:



Vídeo 04 - Funções e Eventos pt.2

Método	Descrição
<code>\$(seletor).text(conteúdo)</code>	Recupera ou define o texto dos elementos.
<code>\$(seletor).val(conteúdo)</code>	Recupera ou define o valor dos elementos de um formulário (Ex.: campos do tipo input, select e radio).

Método	Descrição
<code>\$(seletor).html(conteúdo)</code>	Recupera ou define o conteúdo interno dos elementos (Ex.: o código HTML contido dentro uma 'div').
<code>\$(seletor).append(conteúdo)</code>	Adiciona o conteúdo informado ao fim do conteúdo interno dos elementos.
<code>\$(seletor).prepend(conteúdo)</code>	Adiciona o conteúdo informado ao início do conteúdo interno dos elementos.
<code>\$(seletor).after(conteúdo)</code>	Adiciona o conteúdo informado após os elementos.
<code>\$(seletor).before(conteúdo)</code>	Adiciona o conteúdo informado após os elementos.

O exemplo a seguir mostra como utilizar cada uma das ações HTML apresentadas nesta seção. Este exemplo é composto de alguns elementos HTML, os quais, após pressionar o botão 'Teste', sofrerão modificações.

Código da página (arquivo `acoesHtml.html`):

```
1 <html>
2
3 <head>
4   <script src="http://code.jquery.com/jquery-2.1.3.js"></script>
5 </head>
6
7 <body>
8
9   <script type="text/javascript">
10     $(document).ready(function() {
11
12       $("#b1").click(function() {
13         $('[type="text"]').val("novo valor!");
14         $('[type="text"]').before("Rotulo:");
15         $('[type="text"]').after("<br><br>");
16         $("#d1").prepend("Aula 22 - ");
17         $("#d1").append(" - parte 1");
18         $("#d2").html("Programacao Web - jQuery - 2010");
19       });
20
21     });
22   </script>
23   <div id="d1"
24     style="border-width: 1px; border-style: dotted; border-color: red;">
25   Aprendendo jQuery</div>
26
27   <br>
28   <input type="text" value="aprendendo jQuery 1">
29   <input id="b1" type="button" value="Teste">
30   <br>
31   <br>
32   <div id="d2"
33     style="border-width: 1px; border-style: dotted; border-color: red;">rodape
34   </div>
35
36 </body>
37 </html>
```

Ao executar esse código no navegador, as páginas apresentadas nas Figuras 1 e 2 serão exibidas.

Figura 01 - Antes de pressionar o botão

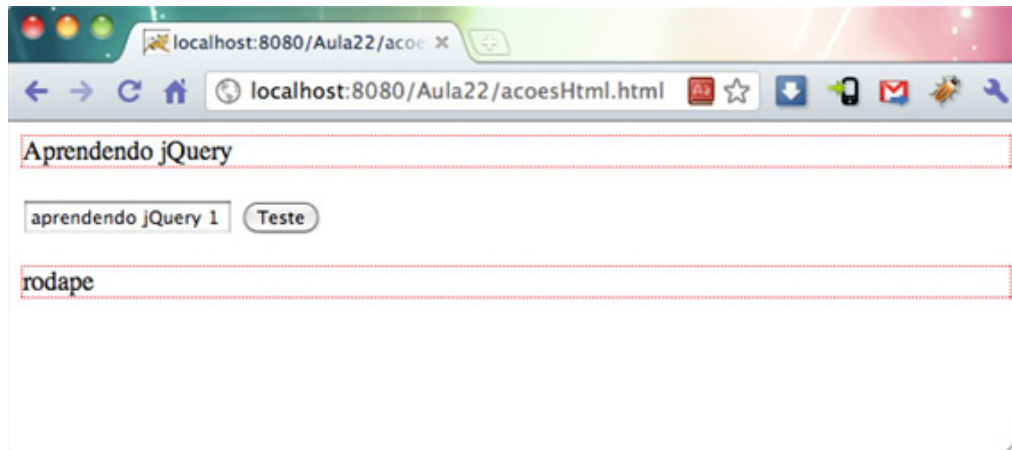
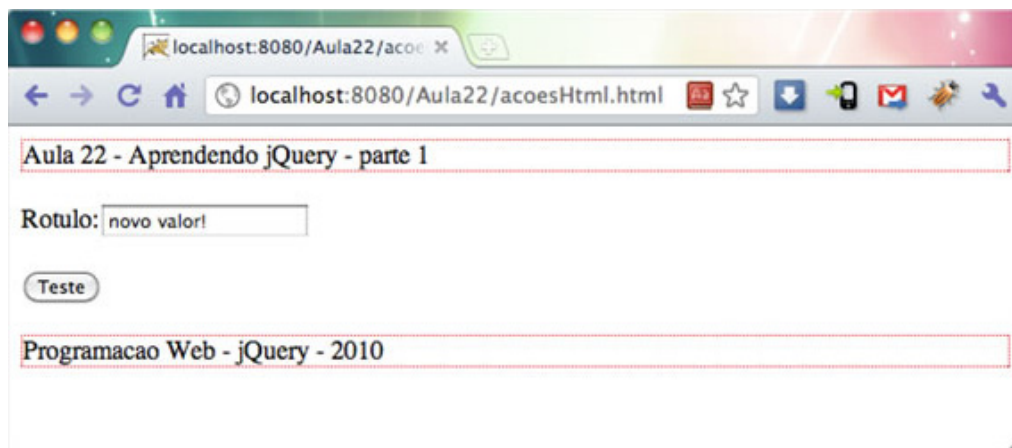


Figura 02 - Após pressionar o botão



Notas

1. A linha 13 utiliza a função 'val' para modificar o conteúdo do campo texto selecionado.

Observe que essa mesma função, quando invocada sem argumentos, ao invés de definir o valor do elemento, retorna o valor corrente dele. Para verificar, experimente substituir o código da linha 12 por:

```
1 alert($("#[type='text']").val());
```

1. As linhas 14 e 15 utilizam os métodos 'before' e 'after', respectivamente, para inserir texto antes e depois do campo texto.
2. As linhas 16 e 17 utilizam os métodos 'prepend' e 'append' para modificar o conteúdo interno do elemento cujo id é 'd1', adicionando, respectivamente, textos no início e no final do texto original.
3. A linha 18 utiliza o método 'html' para definir o conteúdo interno do elemento cujo id é 'd2'.

Note que esse método, assim como o método 'val', quando utilizado sem argumentos, ao invés de definir o valor do elemento, retorna o valor corrente do mesmo. Nesse caso, retorna o conteúdo interno do elemento.

Atividade 02

1. Crie uma página contendo três elementos, um abaixo do outro: um campo texto (TextArea); um botão de Copiar; um elemento div com texto inicial vazio. Sempre que o botão Copiar for clicado, o texto digitado no TextArea deve ser transferido para o final do texto contido dentro do div.
2. Adicione um novo botão ao exemplo anterior que, ao ser clicado, cria, automaticamente, um novo botão logo após o botão Copiar, de texto "Clone". O botão criado, ao ser clicado, deve possuir o mesmo comportamento de o botão Copiar.
3. Nos exemplos das aulas 17 e 18 sobre JavaScript, modifique todos os usos da função getElementById pelos seletores apropriados de jQuery.
4. Altere agora os exemplos das aulas 17 e 18 para que utilizem as funções eval e text de jQuery, sempre que for ler ou alterar o conteúdo de um elemento HTML.

Resumo

Nesta aula, aprendemos como selecionar e modificar elementos da página (por meio dos seletores e dos métodos da API do jQuery), como também aprendemos a registrar e tratar eventos. Na próxima aula, continuaremos estudando a API da biblioteca jQuery.

Autoavaliação

1. Explique os passos necessários para selecionar elementos de uma página HTML utilizando jQuery.
2. Explique como podemos registrar e tratar eventos utilizando o jQuery.
3. Explique como é possível modificar o código HTML da página por meio do jQuery.

Referências

DOCUMENTAÇÃO Alternativa do JQuery. Disponível em: <<http://visualjquery.com/>>. Acesso em: 01 ago. 2012.

JQUERY. Disponível em: <<http://jquery.com/>>. Acesso em: 01 ago. 2012.

TUTORIAL: por onde começar com JQuery. Disponível em: <http://i18n.2kminterativa.com.br/jquery/jquery-getting-started-pt_br.html>. Acesso em: 01 ago. 2012.

TUTORIAIS de JQuery. Disponível em: <<http://docs.jquery.com/Tutorials>>. Acesso em: 01 ago. 2012.