

Desenvolvimento Web II

Aula 08 - jQuery (Parte 2)

Apresentação

Dando continuidade ao assunto visto na aula anterior, veremos, nesta aula, funcionalidades mais avançadas do jQuery. Em particular, você aprenderá como acessar e modificar o estilo CSS de sua página usando a API do jQuery. Além disso, você verá como executar efeitos visuais interessantes sobre os elementos HTML. Por fim, iremos aprender a usar alguns métodos bem úteis da API do AJAX.

Bons estudos!



Vídeo 01 - Apresentação

Objetivos

- Aplicar os métodos da API do jQuery para consultar ou modificar propriedades CSS.
- Diferenciar os efeitos disponíveis na API do jQuery.
- Usar os métodos de AJAX do jQuery.

Métodos – CSS

Na tabela a seguir, você pode conferir os principais métodos disponibilizados pela API do jQuery para recuperar ou definir propriedades CSS dos elementos HTML.



Vídeo 02 - Slide Down

Método	Descrição
<code>\$(seletor).css(nome, valor)</code>	Define o valor da propriedade informada nos elementos selecionados.
<code>\$(seletor).css({propriedades})</code>	Define o valor das propriedades informadas nos elementos selecionados.
<code>\$(seletor).css(nome)</code>	Recupera o valor da propriedade informada no elemento selecionado.

Tabela 1 - Principais métodos utilizados para recuperar e definir propriedades CSS.

Veja um exemplo de uso dos métodos apresentados:

Utilize o arquivo `logo_imd.png` disponível [aqui](#) e o adicione no seu projeto. Lembre de colocar o arquivo de imagem na pasta `WebContent` assim como o HTML do exemplo abaixo.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4   <head>
5     <title>jQuery - CSS</title>
6     <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
7   </head>
8
9   <body>
10    <script type="text/javascript">
11
12      $(document).ready(function(){
13        $("#d2").css({"border-width":"2px", "border-style":"dotted",
14          "border-color":"red", "text-align":"center"});
15      });
16
17      function recuperarValor() {
18
19        var altura = $("#d1").css("height");
20        $("#t1").val(altura);
21
22        var largura = $("#d1").css("width");
23        $("#t2").val(largura);
24
25      }
26
27      function definirValor() {
28
29        var altura = $("#t1").val();
30        $("#d1").css("height", altura);
31
32        var largura = $("#t2").val();
33        $("#d1").css("width", largura);
34
35      }
36    </script>
37
38    <div id="d1" style="background-color: black; text-align: center;">
39      
40    </div>
41
42    <br>
43
44    <div id="d2">
45      Altura: <input id="t1" type="text">
46      <br>
47      Largura: <input id="t2" type="text">
48    </div>
49
50    <br>
51
```

```
52 <div style="text-align: center;">
53   <input type="button"
54     value="Recuperar Valor" onclick="recuperarValor()">
55   <input
56     type="button" value="Definir Valor" onclick="definirValor()">
57 </div>
58
59 </body>
60 </html>
```

Notas sobre o código apresentado:

- Nas linhas 12 e 13, após a página ser completamente carregada, utilizamos o método 'css' que recebe um array de propriedades utilizadas para definir uma borda vermelha pontilhada no elemento 'd2'.
- Na função 'recuperarValor' (linhas 16 a 24), utilizamos o método 'css' informando o nome da propriedade CSS que desejamos saber o valor, em seguida usamos esse valor para atualizarmos os campos de texto 't1' e 't2'.
- Na função 'definirValor' (linhas 26 a 34), utilizamos o método 'css' da API do jQuery informando o nome e o valor da propriedade que desejamos definir ou atualizar (recuperados dos campos texto 't1' e 't2').

Após a execução desse código, a página apresentada na Figura 1 é exibida.

Figura 01 - Exemplo de uso de métodos CSS



Atividade 01

1. Implemente o exemplo mostrado e relate se você encontrou dificuldade.
2. Experimente redimensionar a janela do navegador e pressione o botão "Recuperar Valor". Relate o comportamento apresentado dos componentes da tela e justifique-o com base no código do exemplo.
3. Crie um novo botão que, ao ser clicado, troca as cores de fonte das caixas de texto para vermelho.

Métodos – Efeitos

Nesta seção, veremos como utilizar o jQuery para produzir efeitos visuais, por exemplo, fazer um elemento aparecer ou desaparecer, ficar parcialmente transparente etc. Veja na tabela a seguir o resumo dos principais métodos do jQuery usado para criar esses efeitos.



Vídeo 03 - Fade

Método	Descrição
<code>\$(seletor).hide()</code>	Torna os elementos selecionados não visíveis.
<code>\$(seletor).show()</code>	Torna os elementos selecionados visíveis.
<code>\$(seletor).toggle()</code>	Alterna a visibilidade dos elementos selecionados: se estiverem visíveis ficarão não visíveis e se estiverem não visíveis ficarão visíveis.
<code>\$(seletor).slideUp()</code>	Diminui gradualmente a altura dos elementos selecionados até zero.
<code>\$(seletor).slideDown()</code>	Aumenta gradualmente a altura dos elementos selecionados até a altura original.
<code>\$(seletor).slideToggle()</code>	Análogo ao 'toggle'; no entanto, alternando entre os efeitos gerados pelos métodos 'slideUp' e 'slideDown'.
<code>\$(seletor).fadeOut()</code>	Aumenta gradualmente a transparência dos elementos selecionados até eles desaparecerem.
<code>\$(seletor).fadeIn()</code>	Aumenta gradualmente a opacidade dos elementos selecionados até eles voltarem ao nível de opacidade original.
<code>\$(seletor).fadeTo()</code>	Define o grau de transparência dos elementos informados.
<code>\$(seletor).animate()</code>	Define uma sequência de animações a ser executada nos elementos selecionados.

Tabela 2 - Principais métodos do jQuery para se produzir efeitos visuais.

O código abaixo exemplifica o método 'slideToggle':

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <title>Teste</title>
6 <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
7 </head>
8
9 <body>
10 <script type="text/javascript">
11     $(document).ready(function() {
12
13         $("#b1").click(function() {
14             $("#logo").slideToggle();
15         });
16
17     });
18 </script>
19
20 
21 <br>
22 <input id="b1" type="button" value="Mostrar/Esconder Imagem">
23
24 </body>
25 </html>
26
```

Notas

1. Na linha 11, definimos as operações que serão realizadas após a página ser completamente carregada.
2. Na linha 13, definimos o código a ser executado quando o botão cujo id é 'b1' for clicado.
3. Na linha 14, através do método 'slideToggle' da API de AJAX do jQuery, geramos uma animação que esconde a logo do IMD se ela estiver exibida e mostra o logo do IMD se ele estiver escondido.

Atividade 02

1. Implemente o exemplo mostrado e relate se você encontrou dificuldade.
2. Crie um novo botão que, quando clicado, rode uma função JavaScript executando todas as animações do exemplo.

Métodos – AJAX

Nesta seção, veremos alguns dos métodos que a API do jQuery disponibiliza para facilitar e aumentar a produtividade do desenvolvimento utilizando AJAX. A tabela a seguir faz esse resumo. Para a referência completa, dirija-se ao seguinte endereço: <<http://api.jquery.com/category/ajax/>>



Vídeo 04 - get JSON

Método	Descrição
<code>\$.ajax(opções)</code>	Recupera dados remotos de acordo com as opções informadas como parâmetro.
<code>\$.get(url, dados, callback, type)</code>	Recupera dados remotos utilizando HTTP GET.
<code>\$.post(url, dados, callback, type)</code>	Recupera dados remotos utilizando HTTP POST.

Método	Descrição
<code>\$.getJSON(url, dados, callback)</code>	Recupera dados em formato JSON utilizando HTTP GET.

Tabela 3 - Principais métodos da API para facilitar desenvolvimento AJAX.

Veja a seguir um exemplo utilizando o método 'load':

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/I
2 <html>
3 <head>
4 <title>Teste</title>
5 <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
6 </head>
7
8 <body>
9 <script type="text/javascript">
10     $(document).ready(function() {
11
12         $("#b1").click(function() {
13             $("#ta1").load("info.txt");
14         });
15
16     });
17 </script>
18
19 <textarea id="ta1" rows="5" cols="50"></textarea>
20 <br>
21 <input id="b1" type="button" value="recuperar dados">
22
23 </body>
24 </html>
25

```

Notas

- Na linha 10, definimos as operações que serão realizadas após a página ser completamente carregada.
- Na linha 12, definimos o código a ser executado quando o botão cujo id é 'b1' for clicado.
- Na linha 13, através do método 'load' da API de AJAX do jQuery, buscamos do servidor os dados contidos no arquivo "info.txt" e preenchemos o campo de texto 'ta1'. Crie esse arquivo "info.txt" na pasta WebContent, assim como o HTML, e coloque algum conteúdo nele para que ele seja exibido no textarea do HTML.

Veja agora um exemplo utilizando os métodos 'get' e 'post' (arquivo ajax2.html):

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4   <head>
5     <title>Ajax2</title>
6     <script type="text/javascript" src="http://code.jquery.com/jquery-2.1.3.js"></script>
7   </head>
8
9   <body>
10    <script type="text/javascript">
11      $(document).ready(function() {
12
13        $("#b1").click(function() {
14          var url = "/UrlParaAlgumServlet";
15          $.get( url,
16            {nome: $("#nome").val(), sobrenome: $("#sobrenome").val()},
17            function(data, status, xhr){
18              $("#ta1").val(data);
19            }
20          );
21
22          $("#b2").click(function() {
23            var url = "/UrlParaAlgumServlet";
24            $.post( url,
25              {nome: $("#nome").val(), sobrenome: $("#sobrenome").val()},
26              function(data, status, xhr){
27                $("#ta1").val(data);
28              }
29            );
30
31          });
32        </script>
33
34        <textarea id="ta1" rows="5" cols="50"></textarea>
35        <br>
36        Nome:<input id="nome" type="text" value="John">
37        Sobrenome:<input id="sobrenome" type="text" value="Lennon">
38
39        <br>
40        <input id="b1" type="button" value="recuperar dados utilizando HTTP GET">
41        <br>
42        <input id="b2" type="button" value="recuperar dados utilizando HTTP POST">
43
44      </body>
45    </html>
```

Notas

1. No exemplo anterior usamos a url `"/UrlParaAlgumServlet"` como destino de nossas requisições. Esse servlet precisa existir e processar requisições GET e POST para que o exemplo funcione. Se desejar pode criar você mesmo.

Dando continuidade a conversão do código visto em aulas anteriores, iremos prosseguir incluindo os novos recursos ensinados (arquivo `json_jQuery2.jsp`):

```

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 2</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       $("#livros").click(carregarLivro);
21       $.getJSON("livros.json.jsp", function(dados){
22         jsonObj = dados;
23         carregarLista();
24       });
25     }
26
27     function carregarLista() {
28       var livros = jsonObj.livros.livro;
29       for ( var i = 0; i < livros.length; i++) {
30         var livro = livros[i];
31         var campoLivro = document.getElementById("livros");
32         campoLivro.options[campoLivro.length] = new Option(livro.nome, ""
33           + i);
34       }
35     }
36
37     function carregarLivro() {
38
39       var indice = $("#livros").val();
40       var livros = jsonObj.livros.livro;
41       var livro = livros[indice];
42
43       $("#nome").html(livro.nome);
44       $("#isbn").html(livro.isbn);
45
46       $("#editora").html(livro.editora);
47       $("#edicao").html(livro.edicao);
48       $("#preco").html(livro.preco);
49
50       var autoresStr = "";
51       var autores = livro.autores.autor;

```

```

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

```

```
103 </body>
104 </html>
```

Notas

- Linha 21: utilizamos a função 'getJSON' da API do jQuery para recuperar dados no formato JSON do servidor. A assinatura da função é:
 - `getJSON(url, [dados], [callback(dados, textStatus)]);`
 - o segundo e terceiro parâmetros são opcionais.
- Linha 60: utilizamos o método 'attr' para definir o valor do atributo 'style' do elemento 'divLivroSelecioneado'.

Veja a seguir o arquivo JSON carregado do servidor (arquivo livros.json.jsp):

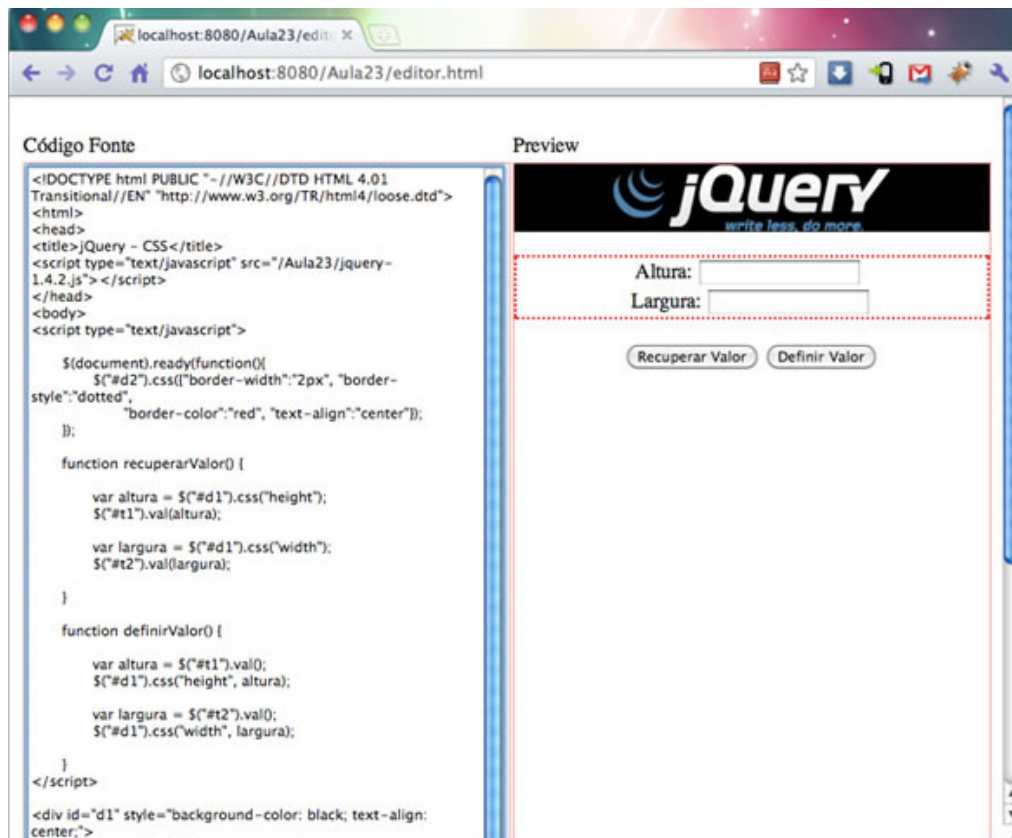

```
1 <%response.setContentType("application/x-javascript; charset=utf-8");%>
2 {
3     "livros":{
4         "livro":[
5             {
6                 "nome":"A Cabana",
7                 "isbn":8599296361,
8                 "editora":"SEXTANTE FICÇÃO",
9                 "edicao":"1ª Edição - 2008",
10                "autores":{
11                    "autor":["YOUNG, WILLIAM P."]
12                },
13                "preco":"12,45",
14                "sinopse":"A filha mais nova de Mackenzie Allen Philip foi raptada durante as férias em fa
15            },
16            {
17                "nome":"A Ilha Sob o Mar",
18                "isbn":8528614441,
19                "editora":"BERTRAND BRASIL",
20                "edicao":"1ª Edição - 2010 ",
21                "autores":{
22                    "autor":[
23                        "ALLENDE, ISABEL",
24                        "SEGUNDO AUTOR"
25                    ]
26                },
27                "preco":"12,45",
28                "sinopse":"O romance narra a vida de Zarité, a escrava que foi vendida aos nove anos de
29            }
30        ]
31    }
32 }
33 }
```

Atividade 03

Atividade Desafio

Utilizando as técnicas aprendidas até o momento, crie uma página HTML que tenha um campo do tipo 'textarea' para entrada de código HTML do lado esquerdo e tenha uma 'div' e que mostre o *preview* do código colocado no 'textarea' do lado direito. Veja isso na Figura 2. O *preview* deverá ser atualizado sempre que o campo 'textarea' perder o foco.

Figura 02 - Editor/Preview



Resumo

Nesta aula, aprendemos como acessar e modificar o estilo (com uso de CSS) de páginas HTML, através do uso da API do jQuery. Essa ferramenta permite a implementação de efeitos visuais bem interessantes, relacionados à mudança de estilo da sua página, mesmo depois da página ter sido carregada. Além disso, você aprendeu como fazer efeitos visuais mais avançados nos elementos HTML, tais como: esconder, aparecer, alterar transparência etc. Por fim, você aprendeu a usar alguns métodos bem úteis da API do AJAX. Em particular, métodos relacionados a requisições assíncronas AJAX.

Autoavaliação

1. Indique como utilizar os métodos da API do jQuery para consultar ou modificar propriedades CSS.
2. Indique o que está disponível na API do jQuery em termos de efeitos visuais.
3. Indique como usar os métodos de AJAX do jQuery para realizar a comunicação via JSON.

Referências

DOCUMENTAÇÃO Alternativa do JQuery. Disponível em: <<http://visualjquery.com/>>. Acesso em: 31 ago. 2012.

JQUERY. Disponível em: <<http://jquery.com/>>. Acesso em: 31 ago. 2012.

TUTORIAL: por onde começar com JQuery. Disponível em: <http://i18n.2kminterativa.com.br/jquery/jquery-getting-started-pt_br.html>. Acesso em: 31 ago. 2012.

TUTORIAIS de JQuery. Disponível em: <<http://docs.jquery.com/Tutorials>>. Acesso em: 31 ago. 2012.