



Programação para Web

Professor Juliano Paulo Menzen

JavaScript - jQuery

- Biblioteca JavaScript cross-browser desenvolvida para simplificar a programação cliente side;
- Criada por John Resig em 2006;
- Segundo a Wikipedia é utilizada por 77% dos 10 mil sites mais visitados no mundo;
- <http://jquery.com/>



JavaScript - Funcionalidades

- Resolução de incompatibilidade entre os navegadores;
- Redução de código;
- Reutilização de código;
- Altamente expansível e com diversas funcionalidades:
 - jQuery UI;
 - jQuery Mobile;
- Trabalha com Ajax e Dom;
- Permite a implementação de recursos CSS1, CSS2 e CSS3;

JavaScript – Versões

- Versão 1.x
- Versão 2.x
 - Mesmas características da versão 1.x, mas sem suporte a navegadores antigos (ie 6, 7 e 8);
- Possui versão full (development) e versão comprimida (produção);

JavaScript – Como incorporar

- Se o arquivo estiver no mesmo local da página;

```
<html>
    <head>
        <script src="jquery-2.1.1.js"></script>
    </head>
    <body>
    </body>
</html>
```

- Utilizando a referência do Google

```
<html>
    <head>
        <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    </head>
    <body>
    </body>
</html>
```

JavaScript - Exemplo

- Código em Javascript puro para atribuir o valor “5” em um elemento e o exemplo em jQuery;

```
document.getElementById( 'Teste' ).value = 5;
```

O mesmo código em jQuery.

```
$( '#Teste' ).val( 5 );
```

JavaScript - jQuery

- O jQuery permite o emprego de seletores para os elementos da mesma forma que o CSS;
- Podemos ter os seguintes tipos de seletores:
 - Seletor por ID;
 - Seletor por Classe;
 - Seletor por Elemento;
 - Agrupamento de seletores;

JavaScript – Seletor por ID

ID Selector (“#id”)

Also in: [Selectors > Basic](#)

Selects a single element with the given id attribute.

- **Exemplo:**
 - **Html:**
 - `<td id="celula"></td>`
 - **JavaScript:**
 - `document.getElementById("célula")`
 - **Jquery:**
 - `$("#celula")`

JavaScript – Seletor por Classe

Class Selector (".class")

Also in: [Selectors > Basic](#)

Selects all elements with the given class.

- **Exemplo:**
 - **Html:**
 - `<td class="celula"></td>`
 - **JavaScript:**
 - `document.getElementsByClassName("célula")`
 - **Jquery:**
 - `$(".celula")`

JavaScript – Seletor por Elemento

Element Selector (“element”)

Also in: [Selectors > Basic](#)

Selects all elements with the given tag name.

- Exemplo:
 - Html:
 - `<td></td>`
 - JavaScript:
 - `document.getElementsByTagName("td")`
 - JQuery:
 - `$("td")`

JavaScript – Seletor Multiplo

Multiple Attribute Selector `[name="value"][name2="value2"]`

Also in: [Selectors > Attribute](#)

Matches elements that match all of the specified attribute filters.

- **Exemplo:**
 - **Html:**
 - `<td></td>`
 - `<p><p>`
 - **Jquery:**
 - `$ ("td", "p")`

JavaScript – Outros seletores

- <http://api.jquery.com/category/selectors/>

JavaScript - Eventos

- O Jquery amplia o numero de eventos dos elementos;
- Antes devemos entender alguns conceitos:

JavaScript - Eventos

- Funções Anônimas
 - Não possuem um nome especificado;
 - A função é atribuída como se fosse um objeto qualquer;
 - Permite proteger variáveis contra o mal uso;
 - Exemplo:

```
var obj = (function() {  
    // declarações a serem protegidas  
    var a, b, c;  
    // retornando um objeto construído a partir de a, b e c  
    return obj;  
})();
```

JavaScript - Eventos

- Evento `$(document).ready();`
 - Executado quando todos os elementos html forem carregados;
 - É diferente do “*load*” do objeto “*window*”, pois este é executado quando tudo é carregado, inclusive imagens, histórico, etc, ...
 - Os eventos devem ser processados dentro do evento “*ready*”;

JavaScript - Eventos

- Evento `$(document).ready();`

- Exemplo:

```
$(document).ready(function() {  
    $('body').text('document esta carregado');  
  
    //Outras funções são adicionadas aqui...  
});
```


JavaScript - Eventos

Eventos de mouse	
click()	Evento gerado quando se produz um click em um elemento da página.
dblclick()	Evento gerado quando se produz um duplo clique sobre um elemento.
hover()	Executada quando o mouse entra e sai de cima de um elemento.
mousedown()	Executado quando o usuário pressiona o botão do mouse.
mouseup()	Executado quando o usuário solta o botão do mouse.
mouseenter()	Executado quando o mouse for situado sobre o elemento.

JavaScript - Eventos

Eventos de mouse... continuação

<code>mouseleave()</code>	Executado quando o mouse sai de cima de um elemento.
<code>mousemove()</code>	Executado quando o mouse é movimentado sobre o elemento.
<code>mouseout()</code>	Executado quando o mouse sai do elemento.

JavaScript - Eventos

Eventos de teclado	
keydown()	Executado no momento em que uma tecla é pressionada, independente de se soltar ou manter a tecla pressionada.
keypress()	Executado quando uma tecla é pressionada. Executado uma vez se a tecla é pressionada e liberada. Executado diversas vezes se a tecla for pressionada e mantida.
keyup()	Executado no momento em que uma tecla é liberada após ser pressionada.

JavaScript - Eventos

Eventos combinados de teclado e mouse

<code>focusin()</code>	Evento executado quando um elemento ganha o foco. Este foco pode ser causado pelo clique do mouse ou pela tabulação do teclado.
<code>focusout()</code>	Este evento ocorre quando um elemento perde o foco, podendo ser perdido pelo uso do mouse ou tabulação do teclado.
<code>focus()</code>	Executado quando um elemento ganha o foco.

JavaScript - Eventos

- Exemplos:

```
<script>
```

```
$(document).ready(function() {
```

```
    $("#minhacamada").dblclick(function(e) {
```

```
        alert("evento de clique executado no elemento de id 'minhacamada'...");
```

```
    });
```

```
    $("#minhacamada").click(function(e) {
```

```
        alert("evento de duplo clique executado no elemento de id 'minhacamada'...");
```

```
    });
```

```
})
```

```
</script>
```

JavaScript - Efeitos

- O jQuery permite a aplicação de diversos efeitos aos elementos de uma página html;
- Dois efeitos bastante aplicados e conhecidos são os efeitos fade e slide;

JavaScript - Efeitos

- Efeito Fade
 - Altera gradualmente a opacidade de um elemento até o ponto esperado;
 - Exemplo:

JavaScript - Efeitos

```
<DOCTYPE html>
<html><head>
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.7.2.min.js"></script>
  <script type="text/javascript">
    $(function() {
      $("#minhaDiv").hide();
      $("#btnTeste").click(function() {
        //Sem parâmetros: o efeito é executado em 400ms
        $("#minhaDiv").fadeIn();
        //Parâmetro com a duração em ms do efeito
        $("#minhaDiv").fadeIn(1000);
        //Parâmetro slow: o efeito é executado em 600ms
        $("#minhaDiv").fadeIn("slow");
        //Parâmetro fast: o efeito é executado em 200ms
        $("#minhaDiv").fadeIn("fast");
        //Informada duração e função de callback
        $("#minhaDiv").fadeIn("fast",
          function() {
            alert("fade concluido");
          }
        );
      });
    });
  </script>
</head><body>
  <button id="btnTeste">Clique aqui</button>
  <div id="minhaDiv" style="height:100px; width:100px; background:green"/>
</body>
</html>
```


JavaScript - Efeitos

- Efeito Slide
 - Altera um elemento “rolando” o mesmo em uma direção até atingir um determinado ponto.
 - Exemplo:

JavaScript - Efeitos

```
<DOCTYPE html>
<html><head>
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.7.2.min.js"></script>
  <script type="text/javascript">
    $(function() {
      //conteúdo das listagens
      $("#btnTeste").click(function() {
        //Sem parâmetros: o efeito é executado em 400ms
        $("#minhaDiv").slideUp();
        //Parâmetro com a duração em ms do efeito
        $("#minhaDiv").slideUp(1000);
        //Parâmetro slow: o efeito é executado em 600ms
        $("#minhaDiv").slideUp("slow");
        //Parâmetro fast: o efeito é executado em 200ms
        $("#minhaDiv").slideUp("fast");
        //Informada duração e função de callback
        $("#minhaDiv").slideUp("fast",
          function(){
            alert("Slide concluido");
          }
        );
      });
    });
  </script>
</head><body>
  <button id="btnTeste">Clique aqui</button>
  <div id="minhaDiv" style="height:100px; width:100px; background:green"/>
</body>
</html>
```

JavaScript - jQueryUI



- Biblioteca JavaScript criada em 01/07/2007;
- Criada por um grupo de desenvolvimento, atualmente liderado por Richard D. Worth;
- Ter por objetivo fornecer uma biblioteca de elementos para o jQuery para a criação de elementos de interface Ricos;

JavaScript - jQueryUI



- Os elementos dividem-se em:
 - Componentes;
 - Widgets;
 - Efeitos;
 - Utilidades;

JavaScript - jQueryUI

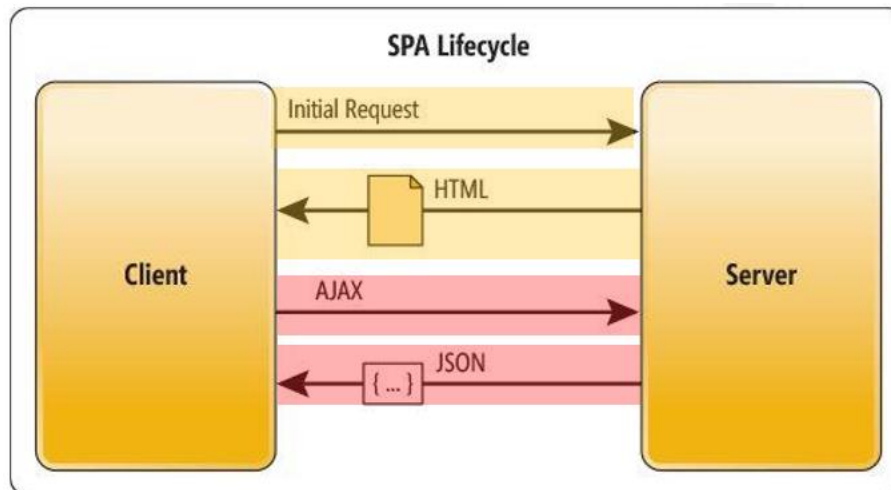
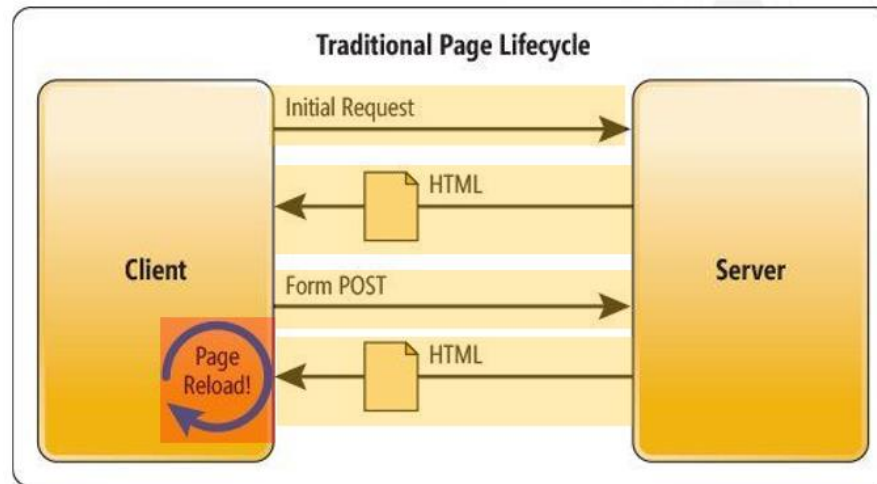


- Download e exemplos de uso da biblioteca:
- <http://jqueryui.com/>

JavaScript - Ajax

- Acrônimo para **JavaScript assíncrono + XML**;
- Não é uma tecnologia;
- Permite carregar elementos da tela de forma assíncrona;
- Faz uso por padrão de transferência de dados conhecido como Json;

JavaScript - Ajax



JavaScript - Ajax

- Json

- Sigla para Javascript Object Notation;

```
1 {  
2   "estado":"São Paulo",  
3   "cidade":"São Carlos",  
4   "bairro":"Nova Aurora",  
5   "rua":"Nova Esperança",  
6   "número":"14"  
7 }  
8
```

```
1 {  
2   "endereço":{  
3     "país":"Brasil",  
4     "estado":"São Paulo",  
5     "cidade":"São Paulo",  
6     "rua":"Brigadeiro Luiz Antônio"  
7   }  
8 }
```

```
1 {  
2   "países_atendidos": ["Brasil", "Estados Unidos", "Grã Bretanha", "África do Sul"]  
3 }
```


JavaScript - Ajax

- Principal emprego para o consumo de APIs em arquitetura backend;
- jQuery facilita na utilização do Ajax, porém pode ser utilizado diretamente no JavaScript ou através de outras bibliotecas;

JavaScript - Ajax

- Exemplo de uso:

```
<script src="jquery-3.7.0.js"></script>
<script>
    function ProcessarConsulta(){
        var tabela="";

        $.get("https://jsonplaceholder.typicode.com/posts", function(data, status){

            tabela = "<table>"
            $.each(data, function(i, item) {
                tabela += "<tr>"
                tabela += "<td>" + item.id + "</td>"
                tabela += "<td>" + item.title + "</td>"
                tabela += "</tr>"
            });
            tabela += "</table>"
            $("#div1").html(tabela);
        });
    }
</script>
```

JavaScript - Ajax

- Exemplo de uso:

```
<body>  
  <input type="button" value="Executar" onclick="ProcessarConsulta()">  
  <div id="div1">  
  
  </div>  
</body>
```

JavaScript - Ajax

- Dúvidas ???