



Universidade Federal de Uberlândia
Faculdade de Computação

Programação para Internet

Módulo 5

Introdução à Biblioteca jQuery

Prof. Dr. Daniel A. Furtado

jQuery – Introdução

- *jQuery* é uma **biblioteca JavaScript**;
- Com *jQuery*, muitas tarefas podem ser realizadas de maneira mais simples (quando comparado ao JavaScript puro). Um de seus fundamentos é: *fazer mais, codificando menos*;
- *jQuery* disponibiliza recursos para manipulação do documento HTML/DOM, estilos CSS, eventos, efeitos de animação, operações assíncronas (AJAX), dentre outros;
- A biblioteca é largamente utilizada no desenvolvimento de websites;
- jQuery facilita a separação do código JavaScript do HTML, o que pode ser uma boa prática e geralmente torna a manutenção do website mais fácil;
- Veja os exemplos anexos:
 - jQuery-Exemplo01.html
 - jQuery-Exemplo02.html

jQuery – Introdução

- Para utilizar jQuery em um website, há duas opções:
 1. Baixar o arquivo da biblioteca no site jquery.com e incluir uma referência para o mesmo dentro do arquivo HTML; ou
 2. Incluir uma referência diretamente de uma rede CDN (*Content Delivery Network*), como a do Google ou da Microsoft;
- Na primeira opção, é possível baixar a versão compactada da biblioteca (**jquery.min.js**, ideal para o website já em funcionamento), ou a versão de desenvolvimento (**jquery.js**), que não é compactada e possui o código legível e adequado para a realização de testes e depuração;

Dica: Utilizar uma referência direta para uma rede CDN é a opção mais prática e também pode ser mais eficiente, pois o navegador já pode ter o arquivo salvo em *cache* devido ao seu uso frequente por outros sites.

jQuery – Introdução

Opção 1: referenciando o arquivo da biblioteca baixado do site *jquery.com*:

```
<!DOCTYPE html>
<html>
<head>
    <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
</body></html>
```

Opção 2: Utilizando a biblioteca diretamente da rede CDN do Google (procurar link no endereço <https://developers.google.com/speed/libraries>)

```
<!DOCTYPE html>
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
<body>
</body></html>
```

jQuery – Introdução

Assim como no código JavaScript puro, o código que utiliza jQuery também deve ser inserido entre as tags `<script>` e `</script>`

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>

<body>

<script>
    // código jQuery
</script>

</body>
</html>
```

Sintaxe jQuery

Usar a biblioteca **jQuery** envolve:

1. Selecionar um elemento HTML
2. Executar uma ação sobre o elemento

Sintaxe básica:

`$ (seletor) . acao ()`

onde:

- O símbolo **\$** indica acesso à biblioteca jQuery (*na verdade “\$” é apenas um “alias” para o nome da função principal “jQuery” da biblioteca*)
- **seletor**: utilizado para buscar ou indicar o elemento HTML sobre o qual a ação será aplicada. O retorno será um **objeto jQuery** correspondente.
- **ação**: operação jQuery a ser aplicada no elemento selecionado;

Exemplo:

`$ ("p") . hide () ;`

Dica: É possível substituir o símbolo **\$** por **jQuery**

Sintaxe jQuery

Outra forma utilizada com frequência é:

`$ (seletor) .evento (função)`

A **função** será executada sobre o **elemento selecionado** quando ocorrer o **evento** indicado;

A função a ser executada pode ser definida sem um nome explícito:

```
$ (seletor) .evento (function () {  
    // código da função  
});
```

Exemplo:

```
$ ("button") .click (function () {  
    $ ("p") .hide ();  
});
```

Evento *Ready* do objeto *Document*

- O código jQuery a seguir é comumente utilizado para executar operações na página assim que o documento HTML terminar de ser carregado pelo navegador (e a estrutura de objetos DOM já tiver sido montada);
- Repare que isto é feito associando uma função anônima ao evento **ready** do objeto **document**. Ela será a primeira função a ser executada quando a página for carregada:

```
$(document).ready(function () {  
    // código jQuery/JavaScript  
});
```

- Entretanto, a partir da versão 3.0 do jQuery, o código a seguir já é suficiente:

```
$(function () {  
    // código jQuery/JavaScript  
});
```


Exemplo 1A – jQuery do servidor do Google

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>

$(document).ready(function() {
    $("h1").click(function() {
        $("#pObjetivos").slideToggle(500);
    });
});

</script>
</head>
<body style="width: 50%; margin: 0 auto">
<h1>Objetivos do Curso (Clique aqui para mostrar/ocultar)</h1>
<p id="pObjetivos">
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
</p>

<h1>Cronograma de Aulas</h1>

</body>
</html>
```

Exemplo 1B – jQuery de Arquivo Externo Local

Baixe o arquivo **jquery-3.3.1.min.js** (versão *compressed/production*) disponível no site **jquery.com** e salve-o em uma subpasta de nome **js**;

```
<!DOCTYPE html>
<html>
<head>
<script src="js/jquery-3.3.1.min.js"></script>
<script>

$(document).ready(function() {
    $("h1").click(function() {
        $("#pObjetivos").slideToggle(500);
    });
});

</script>
</head>
<body style="width: 50%; margin: 0 auto">

<h1>Objetivos do Curso (Clique aqui para mostrar/ocultar)</h1>
<p id="pObjetivos">
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
</p>

<h1>Cronograma de Aulas</h1>
</body></html>
```

Exemplo 1C – Funções jQuery em arquivo externo

Arquivo HTML

```
<!DOCTYPE html>
<html>
<head>
<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/script1.js"></script>
</head>
<body>

<h1>Objetivos do Curso (Clique aqui para mostrar/ocultar)</h1>
<p id="pObjetivos">
Objetivos objetivos objetivos objetivos objetivos <br>
Objetivos objetivos objetivos objetivos objetivos <br>
</p>

</body>
</html>
```

Arquivo script1.js

```
$(document).ready(function() {
    $("h1").click(function() {
        $("#pObjetivos").slideToggle(500);
    });
});
```

Principais formas de seleção (*seletores*)

■ Seleção pelo nome da *tag* HTML

- Exemplo: `$ ("p")` – seleciona todos os elementos `<p>` do documento;

■ Seleção de um elemento pelo seu **ID**

- Deve-se colocar o caractere “#” antes do ID;
- Exemplo: `$ ("#par1")` – seleciona o elemento HTML que possui o atributo ID igual a “par1”;

■ Seleção pela **classe** CSS utilizada pelo elemento

- Utiliza-se o caractere `.` seguido do nome da classe CSS que o elemento a ser selecionado faz referência;
- Exemplo: `$ (".imgGaleria")` – seleciona todos os elementos que fazem uso da classe CSS *imgGaleria*;

Seleção Utilizando *'this'*

- O exemplo a seguir associa o efeito ***fadeOut*** (*desaparece suavemente*) ao evento ***click*** para todas as imagens que utilizam a classe ***imgGaleria***. Entretanto, quando o usuário clicar sobre uma imagem em particular, apenas aquela imagem desaparecerá;
- Observe que a palavra ***this*** na linha destacada seleciona o objeto em particular que disparou o evento (a imagem clicada);

```
<!DOCTYPE html>
<html><head>
<script>

    $(document).ready(function() {
        $(".imgGaleria").click(function() {
            $(this).fadeOut();
        });
    });

</script>
<style> .imgGaleria { width: 400px; height: 300px; } </style>

</head>
<body>
<br>




</body></html>
```

Exercício: Acrescente um botão neste exemplo para que as imagens sejam reexibidas, todas de uma vez, quando o mesmo for pressionado. Utilize o efeito ***fadeIn*** para reexibí-las.

Veja Anexos/Exemplo-jQuery-05.html

jQuery – Introdução

jQuery facilita a separação do código JavaScript do HTML, o que pode ser uma boa prática e geralmente torna a manutenção do website mais fácil.

Exemplo de código JavaScript junto com HTML:

```
<!DOCTYPE html>
<html><body>

<p id="par1">Programação para Internet Programação para Internet </p>

<a onclick="document.getElementById('par1').style.display='none'">Ocultar parágrafo!</a>

</body></html>
```

Exemplo de “separação” do código HTML do JavaScript (com jQuery):

```
<!DOCTYPE html>
<html><body>
<script>

$(function () {
    $("#linkOcultar").click(function () {
        $("#par1").hide();
    });
});

</script>
<p id="par1">Programação para Internet Programação para Internet</p>
<a id="linkOcultar">Ocultar parágrafo!</a>
</body></html>
```

Dica: Esta separação também poderia ser feita utilizando JavaScript puro e o método *addEventListener*, conforme já apresentado anteriormente. Experimente implementar!

Outros exemplos de seletores jQuery

Sintaxe	Descrição
<code>\$ ("*")</code>	Seleciona todos os elementos da página HTML
<code>\$ (this)</code>	Seleciona o elemento HTML corrente
<code>\$ ("p.intro")</code>	Seleciona todos os elementos <p> com class="intro"
<code>\$ ("p:first")</code>	Seleciona o primeiro elemento <p>
<code>\$ ("p:last")</code>	Seleciona o último elemento <p>
<code>\$ ("ul li:first")</code>	Seleciona o primeiro da primeira lista
<code>\$ (" [href] ")</code>	Seleciona todos os elementos com um atributo href
<code>\$ ("a [target='_blank'] ")</code>	Seleciona todos os elementos <a> com o atributo target igual a "_blank"
<code>\$ ("a [target!='_blank'] ")</code>	Seleciona todos os elementos <a> com o atributo target diferente de "_blank"
<code>\$ ("tr:even")</code>	Seleciona todas as linhas pares das tabelas (incluindo a linha 0)
<code>\$ ("tr:odd")</code>	Seleciona todas as linhas ímpares das tabelas

Principais Métodos de Efeitos jQuery

- `$(seletor).hide(velocidade,callback);`
- `$(seletor).show(velocidade,callback);`
- `$(seletor).toggle(velocidade,callback);`
- `$(seletor).fadeIn(velocidade,callback);`
- `$(seletor).fadeOut(velocidade,callback);`
- `$(seletor).fadeToggle(velocidade,callback);`
- `$(seletor).fadeTo(velocidade,opacity,callback);`
- `$(seletor).slideDown(velocidade,callback);`
- `$(seletor).slideUp(velocidade,callback);`
- `$(seletor).slideToggle(velocidade,callback);`
- `$(seletor).delay(velocidade)`

Onde:

- **velocidade:** *velocidade do efeito, opcional ('slow', 'normal', 'fast', ou valor em milissegundos)*
- **callback:** *função a ser executada após o término do efeito (opcional)*
- **opacity:** *define o nível de transparência do objeto (valor entre 0 e 1, onde 1 é tot. opaco)*

Exercício: testar os efeitos modificando os exemplos apresentados anteriormente.

Concatenando Efeitos/Métodos

- É possível concatenar efeitos para execução de maneira sequencial;
- O exemplo a seguir faria com que a imagem *img1* desaparecesse, aparecesse novamente e, depois de 1 segundo, voltasse a desaparecer:

```
$("#img1").fadeOut(300).fadeIn(500).delay(1000).slideUp(200);
```

Concatenando Efeitos/Métodos - Exemplo

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>

$(document).ready(function () {
    $("button").click(function () {
        $("#div1").fadeIn(1000).delay(500).fadeOut(1000);
        $("#div2").fadeIn(1000).delay(500).fadeOut(1000);
        $("#div3").fadeIn(1000).delay(500).fadeOut(1000);
    });
});

</script>
</head>
<body>

<button>Clique aqui para iniciar efeito</button><br><br>

<div id="div1" style="width:100%;height:300px;display:none;background-color:red;"></div>
<div id="div2" style="width:100%;height:300px;display:none;background-color:green;"></div>
<div id="div3" style="width:100%;height:300px;display:none;background-color:blue;"></div>

</body>
</html>
```

Anexos/jquery-Exemplo03.html

Outros Métodos de Eventos

- Além do método de evento **click**, a biblioteca jQuery disponibiliza várias outros métodos que também podem ser utilizados para associar ações a outros tipos de eventos;
- Alguns desses métodos são:

Mouse	Teclado	Formulário	Documento/Janela
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
hover			

- Observe que o nome desses métodos são semelhantes aos nomes dos atributos de eventos da linguagem HTML (*onclick*, *ondblclick*, *onmouseenter*, *onmouseleave*, etc.)

Outros Métodos de Eventos - Exemplos

Uso dos Métodos de Eventos	Descrição
<pre><code>\$("p").click(function () { \$(this).hide(); });</code></pre>	Oculto o parágrafo quando o usuário clica sobre ele.
<pre><code>\$("p").dblclick(function () { \$(this).hide(); });</code></pre>	Oculto o parágrafo quando o usuário clica duas vezes sobre ele.
<pre><code>\$("#p1").mouseenter(function () { alert("Mouse enter!"); });</code></pre>	Apresenta uma mensagem quando o ponteiro do mouse “entra” na região do parágrafo <i>p1</i>
<pre><code>\$("#p1").mouseleave(function () { alert("Mouse leave!"); });</code></pre>	Apresenta uma mensagem quando o ponteiro do mouse se afasta da região do parágrafo <i>p1</i>

Outros Métodos de Eventos - Exemplos

Uso dos Métodos de Eventos	Descrição
<pre><code>\$("#p1").mousedown(function () { alert("Mouse down!"); });</code></pre>	Apresenta uma mensagem no momento em que o botão do mouse é pressionado sobre o parágrafo <i>p1</i> .
<pre><code>\$("#p1").mouseup(function () { alert("Mouse up!"); });</code></pre>	Apresenta uma mensagem no momento em que o botão do mouse é liberado sobre o parágrafo <i>p1</i> .
<pre><code>\$("#p1").hover(function () { alert("You entered p1!"); }, function () { alert("Bye! You now leave p1!"); });</code></pre>	Possibilita associar duas funções de uma só vez: uma delas é executada quando o ponteiro do mouse “entra” na região; a outra é executada quando o ponteiro deixa a região.

Método de Evento *on*

- O método *on* também pode ser utilizado para vincular uma ou várias funções aos eventos;
- Os dois exemplos a seguir são equivalentes

```
$("p").click(function() {  
    alert('Você clicou neste parágrafo!');  
});
```

Utilizando o método *click*

```
$("p").on("click", function() {  
    alert('Você clicou neste parágrafo!');  
});
```

Utilizando o método *on*

Animações

Animações podem ser criadas por meio do método ***animate()***;

A sintaxe é:

```
$(seletor).animate({params}, veloc, callback);
```

onde:

params: definem as propriedades CSS a serem animadas (*não colocar o hífen no nome da propriedade e colocar em maiúscula a primeira letra seguinte*);

veloc: duração da animação ('fast', 'slow' ou um *valor* em milissegundos);

callback: função a ser executada após o término da animação.

Animações – Exemplo 6

- O exemplo a seguir insere uma animação para a imagem ‘**logoufu**’, que é iniciada após clique no botão ‘**Animar**’;
- A animação altera progressivamente as propriedades CSS da imagem (posição **left**, posição **top** e largura **width**), do valor corrente até os valores definidos na função **animate**;

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>

$(document).ready(function() {
    $("button").click(function() {
        $("#logoufu").animate({
            left: '1000px',
            top: '500px',
            width: '500px'
        }, 800);
    });
});

</script>
</head>
<body>

<button>Animar</button>


</body></html>
```

OBS 1: as propriedades **left** e **top** indicam a posição (coordenadas) do elemento com relação ao ancestral mais próximo com posição definida (neste caso, o próprio documento). Para terem efeito, devem ser utilizadas juntamente com **position: absolute**;

OBS 2: a função **animate** não é capaz de animar cores. Algumas propriedades como **background-color** não surtirão efeito.

Veja Anexos/jquery-Exemplo06.html

Animações – Exemplo 7

- O exemplo a seguir insere um efeito de animação que aumenta o tamanho da imagem quando o ponteiro do mouse é passado sobre a mesma; e outro efeito que volta ao tamanho normal quando o ponteiro do mouse se afasta da mesma
(*veja anexos/jquery-Exemplo01.html*)

```
<!DOCTYPE html>
<html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function() {

    $(".imgGaleria").hover(

        function() { // esta funcao eh executada quando o pont. do mouse entra na região da img.
            $(this).animate({
                width: '500px', // a largura da img. será aumentada gradualmente até 500 pixels
                height: '400px' // a altura da img. será aumentada gradualmente até 400 pixels
            });
        },

        function() { // esta funcao eh executada quando o pont. do mouse deixa a região da img.
            $(this).animate({
                width: '400px',
                height: '300px'
            });
        }
    );
});

</script>
<style>.imgGaleria { width: 400px; height: 300px; border-radius: 10px; }</style>
</head>
<body>
<br>



</body></html>
```

O Método *each*

- O método *each* possibilita associar uma ação individual para cada objeto de uma coleção de objetos;
- O método é comumente utilizado em conjunto com a variável *i*, que é atualizada automaticamente para corresponder à posição do objeto dentro da coleção.

```
<!DOCTYPE html>
<html><head>
<script>

    $(document).ready(function() {
        $(".imgGaleria").each(function(i) {
            $(this).delay(200*i).fadeIn();
        });
    });

</script><style> .imgGaleria { width: 400px; height: 300px; } </style>
</head>
<body>





</body></html>
```

Neste exemplo, uma ação com um *delay* diferente é associada a cada imagem da galeria. Observe que o tempo de espera do efeito (*delay*) é crescente, de acordo com a posição da imagem. Isso faz com que as imagens sejam exibidas em sequência, uma após a outra.

jQuery e JavaScript

- Pode-se inserir código JavaScript (JS) normalmente dentro de trechos de código jQuery (e vice-versa). Isso é obvio, pois jQuery é apenas uma biblioteca JS;
- No exemplo a seguir (*anexos/Exemplo-jQuery-07.html*) uma função JavaScript é chamada a partir de um bloco de código da biblioteca jQuery

```
<!DOCTYPE html>
<html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>

function validaIdade(obj)
{
    var idade = obj.value;
    if (idade < 0 || idade > 120)
        return false;

    return true;
}
$(document).ready(function() {
    $("input[type='number']").blur(function() { // o evento 'blur' ocorre
        if (! validaIdade(this))               // quando o objeto perde o foco
            alert('Idade invalida!');
    });
});
</script>

</head>
<body>

<h1>Quando deseja se aposentar? </h1>
<input type="text" placeholder="Seu Nome"><br>
<input type="number" placeholder="idade minima"><br>
<input type="number" placeholder="idade maxima"><br>
<input type="number" placeholder="idade ideal"><br>

</body></html>
```

Exercício 01

- Altere o arquivo anexo *jQuery-Exemplo03.html* para que o conteúdo do tópico “***Das Disciplinas***” seja ocultado e mostrado, alternadamente, quando o usuário clicar sobre “Das Disciplinas”.

Exercício 02

- Construa uma função JavaScript com a assinatura ***showMessage(msg)***. A função deverá exibir a mensagem 'msg' no centro da tela em uma caixa de mensagem elegante, com cor de fundo e borda;
- A caixa de mensagem deve ser exibida utilizando o efeito fadeIn da biblioteca jQuery;
- A mensagem deve ser exibida durante 3 segundos e então desaparecer automaticamente (com o efeito fadeOut);
- **Dicas:**
 - Crie um div no corpo do documento HTML para ser utilizado como caixa de mensagem;
 - Crie uma classe CSS para definir os estilos visuais da caixa de mensagem (cor de fundo, borda, cor da fonte, posição na tela, etc.);
 - Utilize um posicionamento absoluto para exibir a caixa de mensagens sempre próximo ao centro da tela (position: absolute; e as propriedades left e top)
 - Não utilize bootstrap;
 - Sugestão de formatação da caixa de mensagem:

Operacao realizada com sucesso!

Referências

- www.w3schools.com/jquery
- api.jquery.com
- www.wikipedia.org