

Programação para Web

jQuery

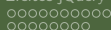
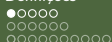
Ivo Calado

Instituto Federal de Educação, Ciência e Tecnologia de Alagoas

22 de Fevereiro de 2016

Roteiro

- 1 Definições
- 2 Efeitos jQuery
- 3 jQuery e HTML



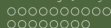
O que é *jQuery*?

- É uma biblioteca JavaScript para construção de páginas Web
- Tem como objetivo tornar facilitar a utilização de JavaScript nas páginas
- Reduz o esforço para realização de ações comuns do JavaScript

O que é necessário conhecer para utilizar *jQuery*?

- HTML
- CSS
- JavaScript





O que *jQuery* possibilita?

O *jQuery* possui suporte a manipulação dos seguintes recursos:

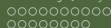
- Manipulação HTML/DOM
- Manipulação de CSS
- Tratamento de eventos HTML
- Efeitos e animações
- AJAX

Além disso, é possível encontrar *plugins* para diversas outras tarefas



Por que utilizar *jQuery*?

- É considerado o framework JavaScript mais popular
 - Google
 - Microsoft
 - IBM
 - Netflix
- É compatível com praticamente todos os navegadores modernos
 - Com exceção do IE 6!



Como adicionar jQuery?

- Via Download de jquery.com

```
<head>  
  <script src="jquery-1.11.3.min.js"></script>  
</head>
```

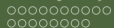
- Utilização de CDNs

```
<script src="https://ajax.googleapis.com/ajax/  
  libs/jquery/1.11.3/jquery.min.js"></script>
```

ou

```
<script src="http://ajax.aspnetcdn.com/ajax/  
  jquery/jquery-1.11.3.min.js"></script>
```





Observação

Qual a importância de se utilizar CDNs?





Observação

Qual a importância de se utilizar CDNs?

- Reaproveitamento do cache
- CDNs servem o arquivo através de servidores mais próximos fisicamente do cliente

Princípios básicos da Sintaxe jQuery (ex-01)

A sintaxe básica do jQuery é **\$(seletor).acao()** ou **jquery(seletor).acao()**

- **\$** ou **jquery**, especificam o acesso ao jQuery
- **seletor** especifica o elemento(s) que serão afetados
- **action()** especifica a ação a ser tomada

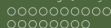
`$(this).hide()` – oculta o elemento atual

`$("p").hide()` – oculta todos os elementos p

`$(".test").hide()` – oculta todos os elementos com a classe 'test'

`$("#test").hide()` – oculta o elemento com o id 'test'





Hello World (ex-01)

```
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<button>Click me</button>
<p>This is another paragraph.</p>
<script>
    $("button").click(function(){
        $("p").hide();
    });
</script>
</body>
```





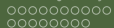
Seletores

jQuery fornece uma ampla gama de possibilidades de busca de seletores. Exemplo:

- `$("p")`: seleciona todos elementos "p"
- `$(".c1")`: seleciona todos os elementos com a classe "c1"
- `$("#id1")`: seleciona o elemento com o id "id1"
- `$("p,h1")`: seleciona todos elementos "p" e "h1"

Uma lista completa pode ser visualizada em:

<http://www.w3schools.com/jquery/tryselect.asp>



O evento “*Ready Event*” (ex-02)

O que acontece se o código que faz seleção a elementos aparece antes dos elementos propriamente ditos?

O evento “Ready Event” (ex-02)

O que acontece se o código que faz seleção a elementos aparece antes dos elementos propriamente ditos?

Os eventos não serão disparados pois os elementos não são encontrados

Causas:

- Problemas no carregamento da páginas
- Definição do código antes do elemento

Solução: utilizar o evento *ready event*!

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

Exercícios

Baixar arquivo de <https://goo.gl/gq5CBS>. Para cada questão, utilizar exemplo correspondente.

- 1 Ocultar todos os elementos “p”
- 2 Ocultar o seletor com o id “test”
- 3 Ocultar todos os elementos com a classe “test”
- 4 Ocultar todos os elementos do documento
- 5 Ocultar todos os elementos com o atributo **href**
- 6 Ocultar as linhas ímpares da tabela



Atividade para casa

Estudar as demais formas de se selecionar elementos a partir de
http:

`//www.w3schools.com/jquery/jquery_ref_selectors.asp`

O que são eventos? I

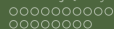
São ações disparadas pela página Web em resposta as ações do usuário não página. Exemplos:

Eventos do mouse:

- click
- dblclick
- mouseenter
- mouseleave

Eventos do teclado:

- keypress
- keydown



O que são eventos? II

- keyup

Eventos de formulário:

- submit
- change
- focus
- blur

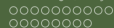
Eventos do documento:

- load
- resize
- scroll
- unload

Evento click

- Atribui tratador de evento a ação de clique de usuário
- Pode ser atribuído a qualquer elemento (botão ou não)

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



Evento dblclick

- Efeito semelhante ao evento click porém requer o clique duplo do usuário

```
$(document).ready(function(){  
    $("p").dblclick(function(){  
        $(this).hide();  
    });  
});
```

Eventos `mouseenter` e `mouseleave`

- O evento *mouseenter* é disparado quando o usuário passa o mouse sobre o elemento

```
$("#p1").mouseenter(function() {  
    alert("You entered p1!");  
});
```

- O evento *mouseleave* é disparado quando o usuário move o mouse para fora do elemento

```
$("#p1").mouseleave(function() {  
    alert("Bye! You now leave p1!");  
});
```

- Testar eventos *mousedown* e *mouseup*



Evento *hover*

- Combina os eventos *mouseenter* e *mouseleave*
- Utiliza duas funções, onde a primeira é disparada quando o mouse é colocada sobre o elemento (*mouseenter*) e a segunda função é chamada quando o mouse é retirado do elemento

```
$("#p1").hover(function(){  
    alert("Entrou em p1!");  
},  
function(){  
    alert("Saiu de p1!");  
});
```



Como é possível atribuir uma classe css a um dado elemento?

```

<style>.c1 { color: red; font-size: 30px; }</style>
<script>
$(document).ready(function(){
    $("#p1").hover(function(){
        $("p").addClass("c1")
    },
    function(){
        $("p").removeClass("c1")
    });
});
</script></head><body>
<span id="p1">This is a paragraph.</span><br>
<p>teste!</p>
</body>

```

Evento *focus* e *blur*

- Evento *focus*

```
$("#input").focus(function(){  
    $(this).css("background-color", "#cccccc");  
});
```

- Evento *blur*

```
$("#input").blur(function(){  
    $(this).css({"background-color", "#ffffff", "color"  
        : "blue"});  
});
```



O método *on*

- Com o método *on* é possível atrelar mais de um tratador de eventos a um elemento por vez

```

$("p").on({
  mouseenter: function() {
    $(this).css("background-color", "lightgray");
  },
  mouseleave: function() {
    $(this).css("background-color", "lightblue");
  },
  click: function() {
    $(this).css("background-color", "yellow");
  }
});

```



Exercícios

Baixar arquivo de <https://goo.gl/KewJiz>. Para cada questão, utilizar exemplo correspondente.

- 1 Alterar o código para que os elementos “p” sejam ocultados ao clique
- 2 Alterar o código para que os elementos “p” sejam ocultados ao clique duplo
- 3 Alterar o código para que quando o mouse passar sobre o elemento “span” ele seja ocultado
- 4 Alterar o código para que o “input” seja ocultado quando o usuário digitar algum caracter
- 5 Utilizar o método “on” para atribuir um tratador ao evento do clique do mouse a todos os elementos “p”

Atividade para casa

Estudar os demais eventos do jQuery a partir de
http://www.w3schools.com/jquery/jquery_ref_events.asp

Eventos *hide* e *show*

- O efeito *hide* tem como objetivo ocultar o elemento

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

- O efeito *show* tem como objetivo tornar visível um elemento previamente ocultado

```
$("#show").click(function(){  
    $("p").show();  
});
```

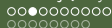
Velocidade para eventos *hide* e *show*

- É possível definir a velocidade para o desaparecimento ou exibição dos elemento

```
$( "button" ). click ( function () {  
    $( "p#1" ). hide ( 1000 );  
    $( "p#2" ). hide ( "slow" );  
    $( "p#3" ). hide ( "fast" );  
});
```

Possíveis valores:

- “slow”
- “fast”
- tempo em milissegundos



Hide, Show, Toggle, Fade e Slide

Efeito *toggle*

- Com o evento *toggle* é possível implementar a troca contínua entre os efeitos *hide* e *show*

```
$(document).ready(function(){
    $("button").click(function(){
        $("p#id1").toggle(1000);
        $("p#id2").toggle("fast");
        $("p#id3").toggle("slow");
        $("p#id4").toggle();
    });
});
```



Efeitos fade

- jQuery fornece a possibilidade de adição de efeitos de fade (desvanecimento) aos elementos
- Efeito de fade é o efeito em que o elemento é removido ou adicionado ao mesmo tempo em que sua transparência é modificada
- São definidos 4 métodos diferentes de fade
 - fadeIn
 - fadeOut
 - fadeToggle
 - fadeTo



Hide, Show, Toggle, Fade e Slide

fadeIn

- Utilizado para adicionar elementos da página

Sintaxe:

```
$(selector).fadeIn(speed, callback);
```

Exemplo:

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow"); // slow ou fast  
    $("#div3").fadeIn(3000);  
});
```



fadeOut

- Utilizado para tornar visível elementos da página

Sintaxe:

```
$( selector ).fadeOut( speed , callback );
```

Exemplo:

```
$( "button" ).click( function () {  
    $( "#div1" ).fadeOut();  
    $( "#div2" ).fadeOut( "slow" );  
    $( "#div3" ).fadeOut( 3000 );  
});
```

fadeToggle

- Equivalente ao toggle: chaveamento entre os efeitos de fadeIn e fadeOut

Sintaxe:

```
$( selector ).fadeToggle( speed , callback );
```

Exemplo:

```
$( "button" ).click( function () {
    $( "#div1" ).fadeToggle();
    $( "#div2" ).fadeToggle( "slow" );
    $( "#div3" ).fadeToggle( 3000 );
} );
```

fadeTo

- Semelhante aos demais métodos de *fading*, porém ao invés de tornar os elementos visíveis/invisíveis modifica o nível de opacidade/transparência dos elementos

Sintaxe:

```
$( selector ).fadeTo( speed , opacity , callback );
```

Exemplo:

```
$( "button" ).click( function() {  
    $( "#div1" ).fadeTo( "slow", 0.15 );  
    $( "#div2" ).fadeTo( "slow", 0.4 );  
    $( "#div3" ).fadeTo( "slow", 0.7 );  
});
```

Exercícios

Baixar arquivo de <https://goo.gl/nSpQh8>. Para cada questão, utilizar exemplo correspondente.

- 1 A partir do jQuery, remova o elemento div utilizando desvanecimento
- 2 Repita o processo do exercício anterior porém utilizando a velocidade “slow”
- 3 Adicione o elemento div utilizando desvanecimento (a velocidade da ação deve ser de 2000 milissegundos)
- 4 Modifique a opacidade do elemento div para 0.2 (utilize a velocidade “slow” para realização da ação)
- 5 Adicione o efeito de chaveamento para adição e remoção do elemento div com desvanecimento (a velocidade da ação deve ser de 1000 milissegundos)

Métodos de *sliding*

- jQuery fornece três métodos para realização de slide de elementos (adição e remoção de elementos via deslizamento)
 - *slideDown*: *exibe o elemento*
 - *slideUp*: *esconde o elemento*
 - *slideToggle*: *chaveamento entre exibição e ocultação*

Sintaxe:

```
$( selector ).slideDown( speed , callback );
```

Exemplo:

```
$( "#flip" ).click( function () {  
    $( "#panel" ).slideUp();  
});
```



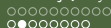
O método *animate*

- A partir da utilização do método *animate* é possível criar animações personalizadas

O método *animate*

- A partir da utilização do método *animate* é possível criar animações personalizadas

```
...
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({left: '250px'});
    });
});
</script>
...
<button>Iniciar animação</button>
<div style="background:blue;height:100px;width:100px;
position:absolute;"></div>
```



Animações: sintaxe

Sintaxe:

```
$( selector ).animate( { params }, speed , callback );
```

- Os parâmetros definem propriedades CSS a serem animadas
- O segundo parâmetro (opcional) define a velocidade com o efeito será ajustado. Valores possíveis: “slow”, “fast” e o tempo em milissegundos
- O terceiro parâmetro (opcional) define uma callback a ser chamada ao final do processo de animação



Animações: sintaxe

Sintaxe:

```
$( selector ).animate( { params }, speed , callback );
```

- Os parâmetros definem propriedades CSS a serem animadas
- O segundo parâmetro (opcional) define a velocidade com o efeito será ajustado. Valores possíveis: “slow”, “fast” e o tempo em milissegundos
- O terceiro parâmetro (opcional) define uma callback a ser chamada ao final do processo de animação

Importante!

Por padrão, todos os elementos HTML tem a posição *static* e não podem ser movidos. Para manipular a posição, deve-se primeiro definir a propriedade *position* para *relative*, *fixed* ou *absolute*.

Manipulando múltiplas propriedades

- É possível manipular múltiplas propriedades CSS em uma única chamada

```
$( "button" ). click ( function () {  
    $( "div" ). animate ( {  
        left: '250px',  
        opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    } );  
});
```



Dúvida?!

É possível manipular todas as propriedades CSS via o método *animate*?

- Sim, porém as propriedades devem ser escritas no formato camel-cased. Ou seja, a propriedade *padding-left* deve ser escrita como **paddingLeft**
- Para que as propriedades ligadas a cores tenham efeito deve-se importar o plugin jQuery UI

```
<link rel="stylesheet" href="http://code.jquery.com/  
  ui/1.11.4/themes/smoothness/jquery-ui.css">  
<script src="http://code.jquery.com/ui/1.11.4/jquery-  
  ui.js"></script>
```



Ver mais opções do plugin em <https://jqueryui.com/>



Utilizando valores relativos

- Ao realizar animações é possível definir valores relativos aos atuais
- Para tal, basta ao definir as propriedades utilizar os operadores `+=` ou `-=`

```
$( "button" ). click ( function () {  
    $( "div" ). animate ( {  
        left : '250px',  
        height : '+=150px',  
        width : '+=150px'  
    } );  
});
```



Utilizando valores predefinidos

- jQuery especifica três valores predefinidos para as propriedades: **show**, **hide** e **toggle**

```
$( "button" ).click( function () {  
    $( "div" ).animate({  
        height: 'toggle'  
    });  
});
```

Chamadas consecutivas ao método *animate*

- Se múltiplas chamadas ao método **animate** são realizadas elas são enfileiradas
- Isto significa que uma chamada será executada após o término da anterior



Exemplo

```

("button").click(function(){
    var div = $("#div");
    div.animate({height: '300px', opacity: '0.4'}, "
        slow");
    div.animate({width: '300px', opacity: '0.8'}, "slow
        ");
    div.animate({height: '100px', opacity: '0.4'}, "
        slow");
    div.animate({width: '100px', opacity: '0.8'}, "slow
        ");
});

```

...

```

<div style="background:#98bf21;height:100px;width:100px
;position:absolute;"></div>

```

Definição

- jQuery possibilita a manipulação de maneira simples dos elementos de DOM

O que é mesmo DOM?

Definição

- jQuery possibilita a manipulação de maneira simples dos elementos de DOM

O que é mesmo DOM?

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."



Métodos text(), html() e val()

São fornecidos três métodos para recuperação e modificação de conteúdo via jQuery

- text() - Ajusta ou recupera o texto do elemento selecionado
- html() - Ajusta ou recupera o conteúdo do elemento selecionado (incluindo marcações HTML)
- val() - Ajusta ou recupera o conteúdo de campos de formulários

```
$("#btn1").click(function(){  
    alert("Text: " + $("#test").text());  
});  
$("#btn2").click(function(){  
    alert("HTML: " + $("#test").html());  
});
```

Uso do val()

```

<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>
</head>
<body>

```

```

<p>Name: <input type="text" id="test" value="Mickey
    Mouse"></p>

```

```

<button>Show Value</button>

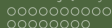
```

Recuperando valores de atributos

- Para recuperarmos valores de atributos basta utilizar o método `attr()`

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#w3s").attr("href"));
    });
});
</script>
</head>
<body>
```

```
<p><a href="http://www.w3schools.com" id="w3s">
W3Schools.com</a></p>
```



Modificando conteúdo de elementos

```
$("#btn1").click(function(){
    $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
});
```



Modificando atributos

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#w3s").attr("href", "http://www.w3schools.
            com/jquery");
    });
});
</script>
```

Funções de callbacks para os métodos text(), html() e val()

```
<script>
$(document).ready(function(){
    $("#btn2").click(function(){
        $("#test2").html(function(i, origText){
            return "Old html: " + origText + " New html
                : Hello <b>world!</b> (index: " + i + "
                )";
        });
    });
});
</script>
```



Função de callback para método attr

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#w3s").attr("href", function(i, origValue){
            return origValue + "/jquery";
        });
    });
});
</script>
```

Adicionando novos elementos ao HTML

São fornecidos 4 métodos para inserção de novos elementos na página

- **append()**: o novo elemento é inserido como último elemento filho do elemento selecionado
- **prepend()**: o novo elemento é inserido como primeiro elemento filho do elemento selecionado
- **after()**: insere o elemento APÓS o elemento selecionado
- **before()**: insere o elemento ANTES do elemento selecionado

Exemplos

```
$("#p").append("Some appended text.");  
$("#p").prepend("Some prepended text.");
```

```
var txt1 = "<p>Text.</p>";  
var txt2 = $("#<p></p>").text("Text.");  
var txt3 = document.createElement("p");  
txt3.innerHTML = "Text.";  
$("#p").append(txt1, txt2, txt3);
```

```
$("#img").after("Some text after");  
$("#img").before("Some text before");
```



Um pouco mais de exemplos...

```
$(document).ready(function(){
    $("#b1").click(function(){
        var $a = $("<a>", {id: "foo", href:"http://www
            .google.com", target: "blank", text: "
            Clique"});

        $("div").append($a);
    });
    $("#b2").click(function(){
        a = document.createElement("a")
        a.id="foo"
        a.href="http://www.google.com"
        a.target="blank"
        a.innerHTML = "Clique"
        $("div").append(a);
    });
});
```

Definição

São fornecidos dos métodos para remoção de elementos do HTML

- **remove()**: Remove os elementos selecionados (e seus elementos filhos)
- **empty()**: Remove os elementos filhos do elementos selecionado

```
$("#div1").remove();  
$("#div1").empty();  
$("p").remove(".test");  
$("p").remove(".test, .demo");
```

