MODULE 5: JQUERY

INTRODUÇÃO

FACULDADE ENGENHARIA
Departamento de Informática

O QUE É O JQUERY?

O jQuery é uma biblioteca Javascript pequena, rápida e cheia de funcionalidades

Não é uma linguagem de programação em si

Executa tarefas como manipulação e percursão do documento HTML, gestão de eventos, animações, torna o Ajax mais simples com uma API fácil de usar através de uma multiplicidade de navegadores

Com combinação de versatilidade e extensibilidade, o jQuery alterou a maneira de escrever Javascript

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

O QUE É O JQUERY?

O jQuery também fornece funcionalidades para os programadores criarem *plugins* no topo da biblioteca Javascript

Isto permite aos programadores criares abstrações de interação de baixo-nível e animações, efeitos avançados e *widgets* temáticos de alto-nível

Esta abordagem modular à biblioteca jQuery permite a criação de poderosas páginas e aplicações *Web*

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

HISTÓRIA DO JQUERY

- O jQuery foi originalmente lançado em Janeiro de 2006 na conferência BarCamp NYC por John Resig
- Foi influenciado pela biblioteca cssQuery, de Dean Edwards
- O jQuery tem também uma história interessante de licenciamento
  - Originalmente sob a CC BY-SA 2.5, foi novamente licenciado para a licença MIT em 2006
  - No final de 2006 encontrava-se duplamente licenciado sob a licença GPL e MIT
  - Como levou a alguma confusão, em 2012 a licença GPL foi abandonada e apenas ficou a MIT

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

HISTÓRICO DE LANÇAMENTOS JQUERY

1.0	August 26, 2006	First stable release
1.1	January 14, 2007	
1.2	September 10, 2007	
1.3	January 14, 2009	Sizzle Selector Engine introduced into core
1.4	January 14, 2010	
1.5	January 31, 2011	Deferred callback management, ajax module rewrite
1.6	May 3, 2011	Significant performance improvements to the attr() and val() functions
1.7	November 3, 2011	New Event APIs: .on() and .off(), while the old APIs are still supported.
1.8	August 9, 2012	Sizzle Selector Engine rewritten, improved animations and $(html, props) flexibility.
1.9	January 15, 2013	Removal of deprecated interfaces and code cleanup
1.10	May 24, 2013	Incorporated bug fixes and differences reported from both the 1.9 and 2.0 beta cycles
1.11	January 24, 2014	
1.12	January 8, 2016	
2.0	April 18, 2013	Dropped IE 6–8 support for performance improvements and reduction in filesize
2.1	January 24, 2014	
2.2	January 8, 2016	
3.0	June 9, 2016	Promises/A+ support for Deferreds, $.ajax and $.when, .data() HTML5-compatible
3.1	July 7, 2016	

COMO O JQUERY FUNCIONA

jQuery: básicos

O atributo *src* no elemento <*script*> deve apontar para uma cópia do ficheiro jQuery

Transfira uma cópia do jQuery de http://jQuery.org e coloque-a na mesma pasta da sua aplicação web

Ou incluir apenas uma hiperligação CDN (Content Delivery Network)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

```
1
2  <!doctype html>
3  <html>
4  <head>
5  <meta charset="utf-8" />
6  <title>Demo</title>
7  </head>
8  <body>
9  <a href="http://jquery.com/">jQuery</a>
10 <script src="jquery.js"></script>
11 <script>
12 // Your code goes here.
13 </script>
14 </body>
15 </html>
16
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

- O JQuery depende do DOM estar pronto para o acesso
- Não queremos utilizá-lo antes da página estar completamente carregada
- Uma vez finalizado, podemos aceder ao jQuery para aceder às partes do documento (de várias maneiras) e manipulá-las (também de diversas maneiras)

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

A sintaxe jQuery está feita à medida para **selecionar** elementos HTML e posteriormente efetuar uma **ação** sobre os elementos selecionados

A sintaxe básica é: **$(*selector*).*action*()**

Um caracter $ para definir/aceder ao jQuery

Um *(selector)* para consultar ou encontrar elementos HTML

Uma ação() a ser executada sobre os elementos

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Para executar o código logo após o documento estar pronto para ser manipulado, o jQuery tem uma declaração: chamada evento *.ready()* :

```
1  $( document ).ready(function() {
2  // Your code here.
3  });
4
```

Reparemos na sintaxe

A maior parte dos comendos jQuery são precedidos do prefixo $

Esta é uma função da biblioteca jQuery que filtra o resto dos dados e invoca a função apropriada baseada no comando

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Para assegurar que o código o corre após o navegador acabar de carregar o documento, muitos programadores Javascript envolvem o seu código numa função *onload* :

```
1
2  window.onload = function() {
3  alert( "welcome" );
4  }
5
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Por exemplo, dentro do evento ***ready***, podemos adicionar um manipulador de clique à hiperligação

```
1
2
3  $( document ).ready(function() {
4  $( "a" ).click(function( event ) {
5  alert( "Thanks for visiting!" );
6  });
7  });
8
9
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Para clique e para a maior parte dos outros eventos, pode prevenir o comportamento padrão chamando o manipulador *event.preventDefault()*:

```
1
2
3  $( document ).ready(function() {
4  $( "a" ).click(function( event ) {
5  alert( "As you can see, the link no longer took you to jquery.com" );
6  event.preventDefault();
7  });
8  });
9
10
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Podemos executar uma série de funcionalidades utilizando uma sintaxe simples e consistente

No caso da função ready(), estamos a associar uma função que contém o resto do nosso código associado ao evento ready

O evento ready dispara quando o DOM está pronto e a função é chamada

Podemos colocar o código apropriado na função, incluindo atribuições com retorno de chamada (callback) a eventos, etc.

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: básicos

Estamos a configurar o nosso documento e vamos esperar que os eventos ocorram:

Ex: Associar um retorno de chamada (callback) a um botão

Ex: Atribuir um estilo a algum texto específico

Ex: Adicionar um texto a um elemento

De modo a fazer o anteriormente descrito, temos que ser capazes de selecionar elementos e itens apropriados no nosso documento

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Selectors

Selecionar por nome de etiqueta (tag):

`$("tagname")` - Retorna um array com a etiqueta que coincide com a *tagname*

Selecionar por ID:

`$("#theid")` - Retorna o elemento com o id igual a *theid*

Selecionar por classe CSS:

`$(".className")` - Retorna um array com os elementos da classe *.className*

Selecionar por par/ímpar:

`$("element:odd")` - Retorna um array de itens que coincidem

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Selectors

Selecionar por índice:

```
$("element:eq(2)")
$("element:lt(4)")
$("element:gt(1)")
```

Retorna elementos especificados pelo índice (eq = igual, lt = menor que, gt = maior que)

Podemos também utilizar seletores para encontrar elementos interiores de forma intuitiva:

`$("outerElement innerElement")` – Pode ser bastante útil se tivermos vários elementos do mesmo tipo mas se apenas quisermos modificar os elementos de um específico `$("element#id")` – Permite-nos encontrar um elemento específico com um id específico

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Selectors

Seletor	Exemplo	Descrição
*	$("*")	All elements
#id	$("#fred")	The element with id="fred"
.class	$(".ethel")	All elements with class "ethel"
HTML element type	$("p")	All <p> elements
Parent descendant	$("div p")	All <p> elements that are descendants of a

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Modificar os elementos selecionados

Uma vez selecionado o elemento, podemos fazer o que quisermos:

Alterar o CSS do elemento(s):

```
$(selector).css()
```

Métodos para alterar a aparên cia dos elemento(s)

```
$(selector).hide()
$(selector).show()
```

Outros:

```
$(selector).append()
$(selector).attr()
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Modificar os elementos selecionados

Uma vez selecionado o elemento, podemos fazer o que quisermos:

Métodos para lidar com eventos e gestão de eventos

```
$(selector).bind()
$(selector).click()
$(selector).focus()
$(selector).mouseover()
```

…

Existem muitos métodos DOM para atualizar as propriedades de um elemento

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Modificar os elementos selecionados

- Tal como na maior parte das vezes, existem várias maneiras de fazer as coisas com o jQuery
- Por vezes, uma abordagem pode ser melhor que outra, mas na maior parte dos casos apenas são diferentes
- Não devemos assumir que a maneira apresentada é a única possível
- Ou necessariamente a melhor maneira

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: iterar através das linhas de uma tabela e adicionar um botão a cada linha

Queremos também que o clique do botão altere a classe da linha

Solução:

1. Primeiro temos que descobrir como iterar sobre as linhas
2. Depois devemos adicionar um novo botão a cada linha
3. No final devemos adicionar um gestor do evento de clique para cada botão fazer a tarefa desejada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: iterar através das linhas de uma tabela e adicionar um botão a cada linha

1. O jQuery tem o iterador *each()*
   Itera sobre cada elemento coincidente de um seletor, executando uma função de retorno (callback) para cada um
   A função de retorno recebe dois argumentos, o elemento atual e o índice atual (começando em 0)
   A diferença com este iterador é a maneira como é executado, uma chamada de função para cada elemento
   Podemos utilizar um seletor para obter as linhas de uma tabela e depois utilizar o *each()* para aceder a cada uma delas

SEBASTIAO@DI.UBI.PT
FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: iterar através das linhas de uma tabela e adicionar um botão a cada linha

2. Podemos utilizar a função *append()*
   Isto permite-nos adicionar texto/HTML a um elemento
   Podemos adicionar um input button à linha atual
3. Existem várias maneiras de fazer isto
   Podemos codificar estaticamente o atributo *onclick* para uma função de retorno que irá alterar a classe CSS
   Podemos aceder ao botão utilizando o jQuery imediatamente após a sua adição e utilizar a função *click()* para associar a uma função

SEBASTIAO@DI.UBI.PT
FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: iterar através das linhas de uma tabela e adicionar um botão a cada linha

```
<script type="text/javascript">
$(document).ready(function() {
$("table#firstTable tr").each(function(ind, el)
{
newE = "<input type='button' value='Toggle " + ind + "' ";
newE = newE + "onclick='process(" + ind + ")'/>";
$(this).append("<td>" + newE + "</td>");
}
);
});

function process(i)
{  $("table#firstTable tr:eq(" + i + ")").toggleClass("evenRow"); }
</script>
```

```
<table id = "firstTable" class = "tableClass" border = "1">
<tr><td>This is row 0</td></tr>
<tr><td>This is row 1</td></tr>
<tr><td>This is row 2</td></tr>
<tr><td>This is row 3</td></tr>
<tr><td>This is row 4</td></tr>
<tr><td>This is row 5</td></tr>
<tr><td>This is row 6</td></tr>
</table>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: ajustar o tamanho da letra do nosso documento à medida que aumentamos ou diminuímos o tamanho da janela

Solução:

1. Precisamos de detetar a largura do documento
2. Precisamos de calcular o tamanho da letra baseado nessa largura
3. Precisamos de assinalar um evento de redimensionamento e chamar a função para atualizar o tamanho

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: ajustar o tamanho da letra do nosso documento à medida que aumentamos ou diminuímos o tamanho da janela

1. Podemos utilizar a função *width()* para descobrir a largura do documento
2. Existem várias maneiras de fazer isto
   - Depende como estamos a formatar a letra
   - O CSS permite muitas métricas diferentes (*pt*, *px*, *em*, %)
   - Para escalar será provavelmente melhor utilizar *em* ou %
   - Se quisermos um tamanho fixo, utilizamos *pt* ou *px*
   - Em qualquer caso, calculamos o novo tamanho de letra baseado no tamanho relativo ao original

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: ajustar o tamanho da letra do nosso documento à medida que aumentamos ou diminuímos o tamanho da janela

3. O jQuery tem a função *resize()* que tem uma função de retorno par ao evento de redimensionamento

   Simplesmente colocamos o nosso código nesta função e vai ser executada de cada vez que a janela for redimensionada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: ajustar o tamanho da letra do nosso documento à medida que aumentamos ou diminuímos o tamanho da janela

```
<script type="text/javascript">
$(document).ready(function() {
var wid = $(document).width();
var fsize = wid/50;
$("body").css("font-size", fsize + "px");

$(window).resize(function() {
wid = $(document).width();
fsize = wid/50;
$("body").css("font-size", fsize + "px");
}
);
});
</script>
```

```
<html>
<body>
Here is some information in the document.
The font size of this text will change
automatically as the document size is
increased and decreased.
Pretty cool!
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: queremos mostrar ao utilizador uma lista de itens e ordená-las pela ordem com que são escolhidas

Os itens vão ser colocados numa nova lista indicando a ordem

Solução:

1. Iterar através da lista, associando um gestor do evento clique
2. O gestor irá remover o item da lista não ordenada e vai adicioná-lo à lista ordenada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: queremos mostrar ao utilizador uma lista de itens e ordená-las pela ordem com que são escolhidas

1. Podemos utilizar novamente a função *each()* para iterar todos os elementos de <li> da uma lista não ordenada

   Para cada um associamos uma função de retorno para o evento de clique *click()*

   Para um estilo adicional podemos também utilizar a função *hover()* para atualizar o estilo quando o rato está sobre o elemento
2. Para o elemento <li> atual, simplesmente removemos da lista não ordenada e adicionamos à lista ordenada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

Exemplo: queremos mostrar ao utilizador uma lista de itens e ordená-las pela ordem com que são escolhidas

```
<script type="text/javascript">
$(document).ready(function() {
$("ul li").each(function(ind, el)

$(this).addClass("evenRow");
$(this).hov
functic
{ $(this).removeClass("evenRov
$(this).addClass("oddRow"
functic
{ $(this).removeClass("oddRov
$(this).addClass("evenRow")
```

```
tdsize = $("tabl
$("table#theTable tr").append("<td><ol></ol></td>");
$(this).remove();
$("ol").append(el);
oldsize = $("ul li").length;
if (oldsize == 0)
{
$("ul").remove();
$("table#theTable tr td:eq(0)").remove();
}});
});
});
</script>
```

```
<ul>
<li>Going to Kennywood</li>
<li>Going to Sandcastle</li>
<li>Biking on Rails to Trails</li>
<li>Hiking the Laurel Highlands</li>
<li>Rock Climbing</li>
<li>Concerts in the Park</li>
<li>Concerts in the Coliseum</li>
</ul>
```

COMO O JQUERY FUNCIONA

jQuery: Eventos

- Todas as ações de uma página web que reagem são chamadas eventos
- Um evento representa o momento exato quando alguma ação ocorre
- Exemplos:
  - Mover o rato sobre um elemento
  - Selecionar um radiobutton
  - Clicar num elemento
  - O termo “despoleta/despoletar" é frequentemente utilizado com eventos

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Eventos

Eventos DOM comuns

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Eventos

Em jQuery, a maioria dos eventos DOM tem um método equivalente jQuery

Para associar um evento de clique a todos os parágrafos de uma página:

```
$("p").click(function(){
  // action goes here!!
});
```

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

Quando um evento clique despoleta num elemento <p>, esconde o elemento <p> atual:

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Adicionar ou remover uma classe HTML

Outra tarefa comum é adicionar ou remover uma classe. Primeiro adicionamos alguma informação do estilo no <head> c como:

```
<style>
a.test {
font-weight: bold;
}
</style>
```

```
$( "a" ).addClass( "test" );
```

Depois, invocamos o .addClass() ao código:

Todos a negritos

```
$( "a" ).removeClass( "test" );
```

Para remover uma classe existente, usar .removeClass():

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Efeitos especiais

O jQuery também fornece efeitos especiais que ajudam a destacar as aplicações web; por exemplo, se criarmos um handler de:

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

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

Então a hiperligação vai desaparecer suavemente quando clicada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Efeitos especiais

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

```
$("button").click(function(){
    $("#mydiv1").fadeIn();
    $("#mydiv2").fadeIn(2000);
});
```

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

```
$("button").click(function(){
    $("#mydiv1").fadeOut();
    $("#mydiv2").fadeOut("fast");
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Efeitos especiais

Method	Description
animate()	Runs a custom animation on the selected elements
clearQueue()	Removes all remaining queued functions from the selected elements
delay()	Sets a delay for all queued functions on the selected elements
dequeue()	Removes the next function from the queue, and then executes the function
fadeIn()	Fades in the selected elements
fadeOut()	Fades out the selected elements
fadeTo()	Fades in/out the selected elements to a given opacity
fadeToggle()	Toggles between the fadeIn() and fadeOut() methods
finish()	Stops, removes and completes all queued animations for the selected elements
hide()	Hides the selected elements
queue()	Shows the queued functions on the selected elements
show()	Shows the selected elements
slideDown()	Slides-down (shows) the selected elements
slideToggle()	Toggles between the slideUp() and slideDown() methods
slideUp()	Slides-up (hides) the selected elements
stop()	Stops the currently running animation for the selected elements
toggle()	Toggles between the hide() and show() methods

COMO O JQUERY FUNCIONA

jQuery: funções de retorno (callbacks) e funções

Ao contrário de muitas linguagens de programação, o JavaScript permite passar funções como argumentos para serem utilizadas posteriormente

Uma *callback* é uma função que é passada como argumento a outra função e é executada quando a função mãe finalizou a sua execução

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno (callbacks) e funções

As callbacks são especiais porque aguardam pacientemente até as suas funções mãe acabem a execução

Entretanto, o browser pode estar a executar outras funções com todo o tipo de tarefas

Para utilizar callbacks, é importante saber como passar as funções às funções mãe

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno sem argumentos

Se uma função callback não tiver argumentos, podemos passá-la assim:

```
1 $.get( "myhtmlpage.html", myCallBack );
```

Quando o $.get() finaliza a obtenção da página myhtmlpage.html, vai executar a função myCallBack()

Nota: o segunda parâmetro aqui é simplesmente o nome da função sem aspas nem parênteses)

```
$("button").click(function(){
    $("p").hide(1000);
    alert("The paragraph is now hidden");
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno sem argumentos

O exemplo abaixo não tem parâmetro de callback, sendo que a caixa de alerta vai ser mostrada antes do efeito de ocultar estar completo :

```
$("button").click(function(){
    $("p").hide(1000);
    alert("The paragraph is now hidden");
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno com argumentos

Executar callbacks com argumentos pode ser complexo

Este exemplo de código ***não*** funciona:

```
1 $.get( "myhtmlpage.html", myCallBack( param1, param2 ) );
```

A razão para falhar é que o código executa *myCallBack*(param1, param2) imediatamente e depois passa o *valor de retorno* de *myCallBack*() como segunda parâmetro no $.get()

Na realidade, queremos passar a função *myCallBack*(), e não o retorno de *myCallBack*(param1, param2) (que pode ser ou não uma função)

Então, como passar a função *myCallBack*() *e* incluir os argumentos de entrada?

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno com argumentos

Para deferir a execução de *myCallBack*() com os seus parâmetros, podemos utilizar uma função anónima como wrapper

Note-se o uso de *function*(): a função anónima faz exatamente uma tarefa: chama a *myCallBack*(), com os valores de param1 e param2:

```
1 | $.get( "myhtmlpage.html", function() { myCallBack( param1, param2 ); });
```

Quando o $.get() finaliza a página myhtmlpage.html, vai executar a função anónima, que por sua vez executa a *myCallBack*(param1, param2)

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: funções de retorno com argumentos

O exemplo abaixo tem um parâmetro de callback que é uma função que vai ser executada após o efeito de ocultar estar completo

```
$("button").click(function(){
    $("p").hide("slow", function(){
        alert("The paragraph is now hidden");
    });
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: encadeamento (chaining)

Existe uma técnica que se chama encadeamento, que permite-nos executar múltiplos comandos jQuery, um após o outro, no(s) mesmo(s) elemento(s)

Desta maneira, não necessitamos de encontrar os mesmos elementos várias vezes

Para encadear uma ação, podemos simplesmente acrescentar outra ação à anterior com .

O exemplo seguinte encadeia os métodos *css*(), *slideUp*() e *slideDown*(). O elemento "`p1`" primeiro fica vermelho, depois desliza para cima e no final para baixo :

```
$("#p1").css("color", "red").slideUp(2000).slideDown(2000);
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: métodos HTML/CSS

Method	Description
addClass()	Adds one or more class names to selected elements
after()	Inserts content after selected elements
append()	Inserts content at the end of selected elements
appendTo()	Inserts HTML elements at the end of selected elements
attr()	Sets or returns attributes/values of selected elements
before()	Inserts content before selected elements
clone()	Makes a copy of selected elements
css()	Sets or returns one or more style properties for selected elements
detach()	Removes selected elements (keeps data and events)
empty()	Removes all child nodes and content from selected elements
hasClass()	Checks if any of the selected elements have a specified class name
height()	Sets or returns the height of selected elements
html()	Sets or returns the content of selected elements
innerHeight()	Returns the height of an element (includes padding, but not border)
innerWidth()	Returns the width of an element (includes padding, but not border)
insertAfter()	Inserts HTML elements after selected elements
insertBefore()	Inserts HTML elements before selected elements
offset()	Sets or returns the offset coordinates for selected elements (relative to the document)
offsetParent()	Returns the first positioned parent element

Method	Description
outerHeight()	Returns the height of an element (includes padding and border)
outerWidth()	Returns the width of an element (includes padding and border)
position()	Returns the position (relative to the parent element) of an element
prepend()	Inserts content at the beginning of selected elements
prependTo()	Inserts HTML elements at the beginning of selected elements
prop()	Sets or returns properties/values of selected elements
remove()	Removes the selected elements (including data and events)
removeAttr()	Removes one or more attributes from selected elements
removeClass()	Removes one or more classes from selected elements
removeProp()	Removes a property set by the prop() method
replaceAll()	Replaces selected elements with new HTML elements
replaceWith()	Replaces selected elements with new content
scrollLeft()	Sets or returns the horizontal scrollbar position of selected elements
scrollTop()	Sets or returns the vertical scrollbar position of selected elements
text()	Sets or returns the text content of selected elements
toggleClass()	Toggles between adding/removing one or more classes from selected elements
unwrap()	Removes the parent element of the selected elements
val()	Sets or returns the value attribute of the selected elements (for form elements)
width()	Sets or returns the width of selected elements
wrap()	Wraps HTML element(s) around each selected element
wrapAll()	Wraps HTML element(s) around all selected elements
wrapInner()	Wraps HTML element(s) around the content of each selected element

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Percorrer

Method	Description
add()	Adds elements to the set of matched elements
addBack()	Adds the previous set of elements to the current set
andSelf()	Deprecated in version 1.8. An alias for addBack()
children()	Returns all direct children of the selected element
closest()	Returns the first ancestor of the selected element
contents()	Returns all direct children of the selected element (including text and comment nodes)
each()	Executes a function for each matched element
end()	Ends the most recent filtering operation in the current chain, and return the set of matched elements to its previous state
eq()	Returns an element with a specific index number of the selected elements
filter()	Reduce the set of matched elements to those that match the selector or pass the function's test
find()	Returns descendant elements of the selected element
first()	Returns the first element of the selected elements
has()	Returns all elements that have one or more elements inside of them
is()	Checks the set of matched elements against a selector/element/jQuery object, and return true if at least one of these elements matches the given arguments

Method	Description
last()	Returns the last element of the selected elements
map()	Passes each element in the matched set through a function, producing a new jQuery object containing the return values
next()	Returns the next sibling element of the selected element
nextAll()	Returns all next sibling elements of the selected element
nextUntil()	Returns all next sibling elements between two given arguments
not()	Remove elements from the set of matched elements
offsetParent()	Returns the first positioned parent element
parent()	Returns the direct parent element of the selected element
parents()	Returns all ancestor elements of the selected element
parentsUntil()	Returns all ancestor elements between two given arguments
prev()	Returns the previous sibling element of the selected element
prevAll()	Returns all previous sibling elements of the selected element
prevUntil()	Returns all previous sibling elements between two given arguments
siblings()	Returns all sibling elements of the selected element
slice()	Reduces the set of matched elements to a subset specified by a range of indices

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMO O JQUERY FUNCIONA

jQuery: Propriedades

Método	Descrição
context	Removed in version 3.0. Contains the original context passed to jQuery()
jquery	Contains the jQuery version number
jQuery.fx.interval	Change the animation firing rate in milliseconds
jQuery.fx.off	Globally disable/enable all animations
jQuery.support	A collection of properties representing different browser features or bugs (Intended for jQuery's internal use)
length	Contains the number of elements in the jQuery object

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

AJAX (Asynchronous JavaScript and XML) é uma tecnologia que se encarrega de transferir dados por detrás e disponibiliza-o numa página web sem carregar a página toda

Com os métodos jQuery AJAX podemos requisitar texto, HTML, XML, ou JSON de um servidor remoto, utilizando ambos o HTTP Get e HTTP Post

Podemos transferir os dados externos diretamente para o elemento HTML selecionado da nossa página

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

O jQuery tem diversas funções convenientes de AJAX

Ver: http://api.jquery.com/category/ajax/

Elas permitem fazer uso de todas as capacidades AJAX

Como se estivéssemos a utilizar um pedido XMLHttpRequest diretamente

Também existem alguns atalhos úteis

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

- O jQuery tem diversas funções convenientes de AJAX
  - Vamos abordar algumas
  - $.ajax()
    - A função mais geral ajax()
    - Tem opções para praticamente tudo o que se precisa
      - Cabeçalhos, callbacks, cache, etc
  - Contudo, em muitos casos, utilizamos uma variação específica da chamada AJAX

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

- O jQuery tem algumas variações pré-definidas que são muito úteis
  - .load()
    - Esta é uma função que é chamada de um elemento selecionado
    - Faz uma chamada AJAX para atualizar a informação do elemento com dados de um servidor
  - Ex: `$("#theTable tr").load('url');`
    - Vai obter o ficheiro do 'url' (que pode resultar de um script) e carrega-o para a linha da tabela selecionada
    - Podemos também colocar uma função de retorno, mas não é obrigatório

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

- O jQuery tem algumas variações pré-definidas que são muito úteis
  - .load()

```
$("button").click(function(){
    $("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){
        if(statusTxt == "success")
            alert("External content loaded successfully!");
        if(statusTxt == "error")
            alert("Error: " + xhr.status + ": " + xhr.statusText);
    });
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

- O jQuery tem algumas variações pré-definidas que são muito úteis
  - .get(), .post()
    - Faz pedidos AJAX da forma usual de passar parâmetros
    - Ambos podem indicar opções ao servidor
    - Muitas outras funções estão disponíveis para opções específicas

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

O jQuery tem algumas variações pré-definidas que são muito úteis

.get()

```
$("button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

O jQuery tem algumas variações pré-definidas que são muito úteis

.post()

```
$("button").click(function(){
    $.post("demo_test_post.asp",
    {
        name: "Donald Duck",
        city: "Duckburg"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

Dados de retorno jQuery AJAX:

Aspetos interessantes em jQuery:

Por omissão (default) está ativado o Intelligent Guess, que não é uma teoria científica!

O jQuery analisa o cabeçalho/formato do documento retornado e faz um julgamento acerca do tipo

Automaticamente filtra baseado no palpite

Se os dados são JSON, cria um objeto Javascript object a partir dele

Se dos dados são XML, cria um documento XML que pode ser filtrado com os seletores jQuery

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

AJAX

Métodos jQuery AJAX

Method	Description
$.ajax()	Performs an async AJAX request
$.ajaxPrefilter()	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by $.ajax()
$.ajaxSetup()	Sets the default values for future AJAX requests
$.ajaxTransport()	Creates an object that handles the actual transmission of Ajax data
$.get()	Loads data from a server using an AJAX HTTP GET request
$.getJSON()	Loads JSON-encoded data from a server using a HTTP GET request
$.getScript()	Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request
$.param()	Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests)
$.post()	Loads data from a server using an AJAX HTTP POST request
ajaxComplete()	Specifies a function to run when the AJAX request completes

Method	Description
ajaxError()	Specifies a function to run when the AJAX request completes with an error
ajaxSend()	Specifies a function to run before the AJAX request is sent
ajaxStart()	Specifies a function to run when the first AJAX request begins
ajaxStop()	Specifies a function to run when all AJAX requests have completed
ajaxSuccess()	Specifies a function to run when an AJAX request completes successfully
load()	Loads data from a server and puts the returned data into the selected element
serialize()	Encodes a set of form elements as a string for submission
serializeArray()	Encodes a set of form elements as an array of names and values

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Obter um elemento com ID foo e adicionar algum HTML

```
$("#foo").append("<p>test</p>");

<html>
<body>
<div>jQuery</div>
<div id="foo">Exemplo</div>
</body>
</html>
```

```
<html>
<body>
<div>jQuery</div>
<div id="foo">Exemplo<p>test</p></div>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Mover parágrafos to element com o id “foo”

```
$(“p”).appendTo(“#foo”);

<html>
<body>
<div>jQuery
<p>moving</p>
<p>paragraphs</p>
</div>
<div id=”foo”>Exemplo</div>
</body>
</html>
```

```
<html>
<body>
<div>jQuery</div>
<div id=”foo”>Exemplo
<p>moving</p>
<p>paragraphs</p>
</div>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Obter/definir atributos

Obter

```
.attr('id')
.html()
.val()
.width()
.css("top")
```

Definir

```
.attr('id', 'foo')
.html("<p>hi</p>")
.val("new val")
.width(60)
.css("top", "80px")
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Obter/definir atributos

```
//Colocar o limite em preto 1px
$(...).css("border", "1px solid black");
//Definir várias propriedades css
$(...).css({
"background": "yellow",
"height": "400px"
});
//Definir todas hiperligações com o href indicando google.com
$("a").attr("href", "http://google.com");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Obter/definir atributos

```
//Substituir HTML com novo parágrafo
$(...).html("<p>I'm new</p>");
<div>whatever</div> turns into
<div><p>I'm new</p></div>
// Definir atributo checkboxes "checked" para checked
$(":checkbox").attr("checked","checked");
// Definir valor introduzido para 3
$(...).val("3");
//Obter o valor introduzido
$(...).val();
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Eventos

```
//Quando um botão é clicado, fazer alguma coisa
$("button").click(function(){
something();
});
//Definir um evento personalizado e despoletá-lo
$("button").bind("expand", function(){
something();
});
$("button:first").trigger("expand");
//Desassociar o evento personalizado
$("button").unbind("expand");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Delegação de Eventos

```
//Associar eventos ao documento
$("button").live('click', function(){
something();
});
// Associar delegação de eventos a elementos
$("form").delegate("button", "click", function(){
something();
});
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Animação/Efeitos

```
//Com cada clique, deslizar para cima / baixo o div
$(...).click(function(){
$("div:first").slideToggle();
});
//Animar os elementos até 300px de largura em .5 segundos
$(...).animate({ "width": "300px" }, 500);
//Tirar o foco dos elementos, desvanecendo-os para 30% opacidade em .5s
$(...).fadeTo(500, 0.3);
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Percorrer – obter células da tabela anteriores a #myCell

```
$("#myCell").prevAll()

<html>
<body>
<table><tr>
<td></td>
<td></td>
<td id="myCell"></td>
<td></td>
</tr></table>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EXEMPLOS

Percorrer – obter parágrafos seguintes à tabela

```
$("table").next().find("p");
```

```
<html>
<body>
<table></table>
<div>
<p>foo</p>
<span>bar</span>
</div>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática