

A BIBLIOTECA JQUERY

JavaScript
jQuery

JQUERY

- jQuery é um ficheiro JavaScript que se pode incluir em páginas web.
- Providencia uma forma fácil de aceder aos elementos da página usando seletores CSS-style podendo executar tarefas com esses elementos através de métodos disponibilizados (*loop* de elementos, adicionar/remover elementos, manipulação de eventos, fade in/out de elementos, manipulação de pedidos Ajax).
- Manipula processos de compatibilidade de forma transparente

FICHEIRO / VERSÕES JQUERY

- Os ficheiro jQuery pode ser descarregado do sítio <http://jquery.org>
- Existem dois ficheiros `jQuery.1.x.x.js` e `jQuery1.x.x.min.js` (o mesmo código mas sem espaços)
- Versões:
 - jQuery 1.x.x – mantem a compatibilidade com IE6, 7 e 8
 - jQuery 2.x.x – retira a compatibilidade com browsers antigos

INCLUSÃO DA BIBLIOTECA JQUERY

■ Através do ficheiro jQuery descarregado

```
■ <head>  
    <title>Titulo</title>  
    <script src='jquery-1.11.3-min.js'></script>  
    <script src='my_script.js'></script>  
</head>
```

...

ou melhor

```
■ ...  
    </section>  
    <script src='jquery-1.11.3-min.js'></script>  
    <script src='my_script.js'></script>  
</html>
```

INCLUSÃO DA BIBLIOTECA JQUERY

- Ou via Google CDN (*Content Distribution Network*)

- ```
<head>
 <title>Titulo</title>
 <script src='<script src=
 'https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/
 jquery.min.js'></script>
 <script src='my_script.js'></script>
</head>
```

...

ou melhor

- ```
...
</section>
<script src='https://ajax.googleapis.com/ajax/libs/
jquery/2.1.3/jquery.min.js'></script>
<script src='my_script.js'></script>
</html>
```

SELETORES

jQuery

ANATOMIA DE UM SCRIPT JQUERY

- Função construtora de objetos *jQuery* é base de tudo.

■ `jQuery('li.hot')` ou `$('li.hot')`

Diagrama de anotação:

- Um bracket vermelho sob `jQuery` aponta para o texto "Função construtora".
- Um bracket vermelho sob `'li.hot'` aponta para o texto "seletor".
- Um bracket vermelho sob `$` aponta para o texto "alias da função construtora jQuery".

- Anatomia de uma instrução típica *jQuery*

■ `$('li.hot').addClass('complete')`

Diagrama de anotação:

- Um bracket vermelho sob `$('li.hot')` aponta para o texto "Objeto jQuery".
- Um bracket vermelho sob `.addClass` aponta para o texto "método".
- Um bracket vermelho sob `('complete')` aponta para o texto "parâmetros".
- Um bracket azul sob `$('li.hot')` aponta para o texto "seletor".
- Um bracket azul sob `.addClass('complete')` aponta para o texto "ação".

SELETORES

SELETORES DE BASE	
*	Todos os elementos
elemento	Todos com o nome indicado
#id	Elementos cujo atributo <i>id</i> seja o indicado
.class	Elementos cujo atributo <i>class</i> seja o indicado
seletor1, seletor2	Conjunto de elementos listados

HIERARQUIA	
ascendente descendente	Todos os elementos descendentes de outro elemento (ex: <i>li a</i>)
pai > filho	Todos os elementos que são filhos diretos de outro elemento
anterior + próximo	Elemento irmão imediatamente adjacente ao anterior
anterior ~ irmão	Elementos que são irmão do elemento anterior

SELETORES

FLTROS DE BASE	
:not (seletor)	Todos os elementos excepto o indicado - ex: div:not('#sumary')
:first	O primeiro elemento da seleção
:last	Último elemento da seleção
:even	Conjunto de elementos da seleção com índice par
:odd	Conjunto de elementos da seleção com índice ímpar
:eq (índice)	Elementos cujo índice é igual ao indicado no parâmetro
:gt (índice)	Elementos cujo índice é maior que o indicado no parâmetro
:lt (índice)	Elementos cujo índice é menor que o indicado no parâmetro
:header	Todos os elementos header <h1> ... <h6>
:animated	Todos os elementos que estão a ser animados
:focus	Todos os elementos que têm o foco.

SELETORES

FILTROS DE CONTEÚDO

:contains ('texto')	Todos os elementos que contêm o texto indicado
:empty	Todos os elementos que não têm descendentes
:parent	Todos os elementos que têm elementos descendentes (pode ser texto ou elementos)
:has (seletor)	Elementos que contêm pelo menos um elemento que corresponde ao seletor indicado (ex: <i>div:has(p)</i> – todos os div's que contêm um elemento p)

FILTROS DE VISIBILIDADE

:hidden	Todos os elementos que estão escondidos
:visible	Todos os elementos que consomem espaço no <i>layout</i> da página

SELETORES

FILTROS DE 'FILHOS'	
:nth-child (expr)	A seleção é feita considerando o primeiro índice com o valor 1.
:first-child	Primeiro elemento da seleção
:last-child	Último elemento da seleção
:only-child	Onde existe apenas um filho com elemento (ex: <i>div p:only-child</i>)

SELETORES

FILTROS DE ATRIBUTOS	
[atributo]	Elementos que correspondem ao atributo indicado.
[atributo='valor']	Elementos que correspondem ao atributo com o valor indicado.
[atributo!='valor']	Elementos que não correspondem ao atributo com o valor indicado.
[atributo^='valor']	Elementos cujo atributo começam com o valor indicado.
[atributo\$='valor']	Elementos cujo atributo terminam com o valor indicado
[atributo*='valor']	Elementos cujo valor do atributo aparece em qualquer parte
[atributo ='valor']	Elemento cujo valor do atributo é igual à <i>string</i> fornecida ou começando com a <i>string</i> seguida de hífen
[atributo~='valor']	Elementos cujo valor do atributo aparece separados por espaços
[atributo] [atributo2]	Elementos que correspondem aos seletores indicados

SELETORES

FORMS	
:input	Elementos de input.
:text	Elementos de input de texto.
:password	Elementos de input de password.
:radio	Elementos de <i>radio button</i> .
:checkbox	Elementos de <i>check box</i> .
:submit	Elementos de botões de <i>submit</i> .
:image	Elementos de imagem
:reset	Elementos de botão <i>reset</i>
:button	Elementos de botão < <i>button</i> >
:file	Elementos de input de ficheiros
:selected	Todos os elementos seleccionados nas <i>drop-list</i>
:enabled	Todos os elementos ativos no formulário
:disabled	Todos os elementos desativados no formulário (usando a propriedade CSS <i>disabled</i>)
:checked	Todos os elementos <i>radio</i> e <i>checkbox</i> seleccionados

MÉTODOS

jQuery

MÉTODOS – FILTRAGEM DE CONTEÚDO

OBTER/ALTERAR CONTEÚDO DOS ELEMENTOS

<code>.html ([html])</code>	Devolve / atualiza informação HTML da seleção <i>jQuery</i>
<code>.text ([texto])</code>	Devolve / atualiza o conteúdo textual da seleção <i>jQuery</i>
<code>.replaceWith ([html])</code>	Substitui os elementos indicados na seleção <i>jQuery</i> pelo conteúdo indicado
<code>.remove ()</code>	Remove os elementos indicados da seleção <i>jQuery</i> da árvore DOM

OBTER/ALTERAR CONTEÚDO DOS ATRIBUTOS

<code>.attr (atributo [, valor])</code>	Devolve ou atualiza o valor de um atributo da seleção <i>jQuery</i>
<code>.removeAttr (atributo)</code>	Remove um atributo e respetivo valor da seleção <i>jQuery</i>
<code>.addClass (valor)</code>	Adiciona um novo valor ao atributo <i>class</i> da seleção <i>jQuery</i>
<code>.removeClass (valor)</code>	Remove o valor especificado do atributo <i>class</i> da seleção <i>jQuery</i>
<code>.css (propriedade [, valor] {objeto literal notation})</code>	Devolve / atualiza propriedades CSS aplicadas à da seleção <i>jQuery</i>

MÉTODOS – FILTRAGEM DE CONTEÚDO

OBTER/ALTERAR ELEMENTOS

<code>.before (html)</code>	Insere conteúdo antes dos elementos da seleção <i>jQuery</i>
<code>.after (html)</code>	Insere conteúdo depois dos elementos da seleção <i>jQuery</i>
<code>.prepend (html)</code>	Insere conteúdo dentro dos elementos da seleção <i>jQuery</i> logo após a <i>tag</i> de abertura
<code>.append (html)</code>	Insere conteúdo dentro dos elementos da seleção <i>jQuery</i> antes da <i>tag</i> de fecho
<code>.remove ()</code>	Remove todos elementos da seleção <i>jQuery</i>
<code>.clone ()</code>	Cria uma cópia da seleção <i>jQuery</i> (incluindo todos os nós descendentes)
<code>.unwrap ()</code>	Remove todos os elementos ascendentes à seleção <i>jQuery</i>
<code>.detach ()</code>	Mesmo que <code>.remove()</code> mas deixando um cópia em memória
<code>.empty ()</code>	Remove nós descendentes da seleção <i>jQuery</i>
<code>.add ()</code>	Permite adicionar novos elementos a uma seleção <i>jQuery</i> de forma a aplicar uma ação específica a todos eles.

MÉTODOS – PROCURAR ELEMENTOS

PROCURAR ELEMENTOS GENÉRICOS

<code>.find (seletor)</code>	Todos os elementos que dentro da seleção <i>jQuery</i> correspondem ao seletor indicado
<code>.closest (seletor)</code>	Ascendente próximo (não obrigatoriamente o ascendente direto) da seleção <i>jQuery</i> que corresponde ao seletor indicado
<code>.parent ()</code>	Ascendente direto da seleção <i>jQuery</i>
<code>.parents ()</code>	Todos os ascendentes da seleção <i>jQuery</i>
<code>.children ()</code>	Todos os descendentes da seleção <i>jQuery</i>
<code>.siblings ()</code>	Todos os elementos irmãos da seleção <i>jQuery</i>
<code>.next ()</code>	Próximo elemento irmão da seleção <i>jQuery</i>
<code>.nextAll ()</code>	Todos os elementos irmãos seguintes da seleção <i>jQuery</i>
<code>.prev ()</code>	Elemento irmão anterior à seleção <i>jQuery</i>
<code>.prevAll ()</code>	Todos os elementos irmãos anteriores à seleção <i>jQuery</i>

MÉTODOS – PROCURAR ELEMENTOS

PROCURAR / TESTAR

.filter (seletor)	Encontra elementos que correspondam ao seletor indicado dentro da seleção <i>jQuery</i>
.not (seletor)	Encontra elementos que não correspondam ao seletor indicado
.has (seletor)	Encontra elementos que correspondam ao seletor indicado dentro da seleção <i>jQuery</i>
.is (condição)	Verifica se o elemento selecionado corresponde à condição (devolve Boolean)
:contains (texto)	Seleciona os elementos que contêm o texto especificado

ORDEM NA PROCURA

.eq (índice)	Elemento que possui o índice indicado
.lt (índice)	Elementos cujo índice é menor que o indicado
.gt(índice)	Elementos cujo índice é maior que o indicado

MÉTODOS – DIMENSÃO

DIMENSIONAR

<code>.height ()</code>	Altura da caixa (sem margem, <i>border</i> e <i>padding</i>)
<code>.width ()</code>	Largura da caixa (sem margem, <i>border</i> e <i>padding</i>)
<code>.innerHeight ()</code>	Altura da caixa com <i>padding</i>
<code>.innerWidth ()</code>	Largura da caixa com <i>padding</i>
<code>.outerHeight ()</code>	Altura da caixa com <i>padding</i> e <i>border</i>
<code>.outerWidth ()</code>	Largura da caixa com <i>padding</i> e <i>border</i>
<code>\$(document).height ()</code>	Altura do documento HTML
<code>\$(document).width ()</code>	Largura do documento HTML
<code>\$(window).height ()</code>	Altura da janela do browser
<code>\$(window).width ()</code>	Largura da janela do browser

MÉTODOS – POSIÇÃO

POSICIONAR

.offset ()	Obtém/atualiza as coordenadas do elemento relativamente ao canto superior/esquerdo do objeto documento
.position ()	Obtém/atualiza as coordenadas do elemento relativamente ao canto superior/esquerdo do elemento ascendente
.scrollLeft ()	Obtém o posição da barra de deslocamento horizontal para o primeiro elemento da seleção <i>jQuery</i> , ou atualiza a posição da barra de deslocamento horizontal para os elementos da seleção
.scrollTop ()	Obtém o posição da barra de deslocamento vertical para o primeiro elemento da seleção <i>jQuery</i> , ou atualiza a posição da barra de deslocamento vertical para os elementos da seleção

MÉTODOS – EFEITOS / ANIMAÇÃO

ANIMAÇÕES BÁSICAS

<code>.show ()</code>	Mostra os os elementos selecionados
<code>.hide ()</code>	Esconde os elementos selecionados
<code>.toggle ()</code>	Alterna entre mostrar ou esconder elementos selecionados

FADING

<code>.fadeIn (tempo)</code>	Faz aparecer os elementos selecionados
<code>.fadeOut (tempo)</code>	Desvanece os elemento selecionados tornando-os transparentes
<code>.fadeTo (t, op)</code>	Altera a opacidade dos elementos selecionados
<code>.fadeToggle ()</code>	Alterna entre mostrar ou esconder elementos selecionados mudando a sua opacidade (o contrário do estado atual)

MÉTODOS – EFEITOS / ANIMAÇÃO

SLIDING

<code>.slideDown()</code>	Mostra os elementos selecionados através de um deslizamento para baixo
<code>.slideUp ()</code>	Mostra os elementos selecionados através de um deslizamento para cima
<code>.slideToggle ()</code>	Esconde ou mostra os elementos selecionados com um deslizamento na posição oposta à definida no momento

PERSONALIZAÇÃO

<code>.delay ()</code>	Para a execução dos subsequentes elementos presentes na <i>query</i> de animação
<code>.stop ()</code>	Para a animação se ela estiver em execução
<code>.animate ()</code>	Cria animações personalizadas Sintaxe: <code>.animate({styles}[, speed][, easing][, complete]);</code>

MÉTODOS – EVENTOS

DOCUMENT / FILE

.ready ()	Método que verifica se uma página está pronta a ser processada, analisando se os browsers suportam eventos <i>DOMContentLoaded</i> .
.load ()	Define a ação a ser realizada depois de da página e seus recursos terem sido carregados (no <i>jQuery</i> foi substituído pelo método <i>.on()</i>);

INTERAÇÃO COM O UTILIZADOR

.on ()	Define o evento a aplicar ao à seleção <i>jQuery</i>
----------------	--

OBTER/ALTERAR VALOR DO FORM

.val ()	Utilizado em primeiro lugar com os elementos <i><input></i> , <i><select></i> e <i><textarea></i> . Permite obter e alterar os valores dos elementos indicados na seleção <i>jQuery</i> .
.isNumeric ()	Verifica se o valor representa um valor numérico e devolve um valor Boolean

A PÁGINA HTML

```
<html>
  <head>
    <title>Exemplo básico de JQuery</title>
    <link rel="stylesheet" href="css/estilos.css">
  </head>
  <body>
    <div id="page">
      <h1 id="header">Lista</h1>
      <h2>Compra de Mantimentos</h2>
      <ul>
        <li id="one" class="hot"><em>Figos</em> frescos</li>
        <li id="two" class="hot">Pinhões</li>
        <li id="tree" class="hot">Mel</li>
        <li id="four">Vinagre balsâmico</li>
      </ul>
    </div>
    <script src="js/jquery-1.11.3.js"></script>
    <script src="js/exemplo-basico.js"></script>
  </body>
</html>
```


SELEÇÃO DE ELEMENTOS

Um elemento

`$('ul')`



Índice

Elemento

0

ul

Vários elementos

`$('li')`



Índice

Elemento

0

li#one.hot

1

li#two.hot

2

li#three.hot

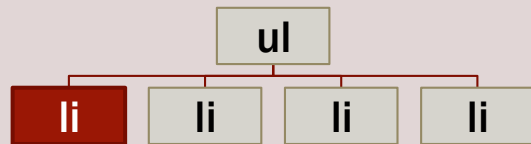
3

li#four

SELEÇÃO DE ELEMENTOS

Um elemento por id

`$('#one')`



Índice

Elemento

0

li#one.hot

Vários elementos por class

`$('.hot')`



Índice

Elemento

0

li#one.hot

1

li#two.hot

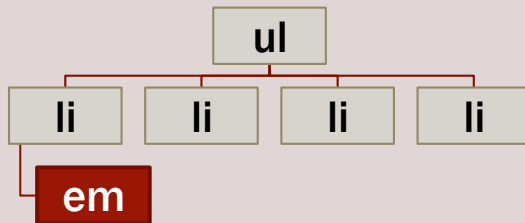
2

li#three.hot

SELEÇÃO DE ELEMENTOS

Um elemento por descrição completa

`$('div#page ul li#one em')`



Índice	Elemento
0	em

Vários elementos por descrição

`$('div#page ul li.hot')`



Índice	Elemento
0	li#one.hot
1	li#two.hot
2	li#three.hot

OBTER CONTEÚDO DOS ELEMENTOS

Obtenção o conteúdo HTML dos elementos internos

```
var listhtml = $('ul').html ( );
```

```
<li id="one" class="hot">  
    <em>Figos</em> frescos</li>  
<li id="two" class="hot">Pinhões</li>  
<li id="tree" class="hot">Mel</li>  
<li id="four">Vinagre balsâmico</li>
```

```
var lihtml = $('li').html ( );
```

```
<em>Figos</em> frescos
```

Obtenção o conteúdo textual dos elementos internos

```
var listtext = $('ul').text()
```

```
Figos frescos  
Pinhões  
Mel  
Vinagre balsâmico
```

```
var litext = $('li').text ( );
```

```
Figos frescos Pinhões Mel Vinagre balsâmico
```

MODIFICAÇÃO DO CONTEÚDO DOS ELEMENTOS

Modificação do conteúdo HTML ou textual dos elementos internos

```
$('#one').html('<em>Castanhas</em>');

$('li.hot').html(function() {
    return '<em> + $(this).text() + '</em>;
});

$('li:contains("pinhões")').text('amendoa');

$('li#one').replaceWith('<p>texto</p>');
```

VERIFICAÇÃO SE A PÁGINA ESTÁ PRONTA

- É importante garantir que a página esteja carregada e a árvore DOM criada para ser possível selecionar elementos. Isto pode ser realizado da seguinte forma:

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

- ou

- ```
$(function() {  
    // código  
});
```

EVENTOS EM JQUERY

- Os eventos são ativados pelo método `.on()`

- Ex de utilização:

```
$( 'li' ).on( 'click', function() {  
    $(this).addClass( 'complete' );  
} );
```

Grupo	Eventos
UI	focus, blur, change
TECLADO	input, keydown, keyup, keypress
RATO	click, dblclick, mouseup, mousedown, mouseover, mousemove, mouseout, hover
FORM	submit, select, change
DOCUMENT	ready, load, unload
BROWSER	error, resize, scroll

O OBJECTO EVENT EM JQUERY

- Na manipulação de eventos a respetiva função recebe um **objecto event**:

```
$('.li').on('click', function(e){  
    eventtype = e.type;  
});
```

Propriedade	Descrição
<code>type</code>	Tipo de evento (ex., <i>click</i> , <i>mouseover</i>)
<code>which</code>	Botão ou tecla que foi pressionada
<code>data</code>	Um objeto passado à função de ação do evento com informação extra
<code>target</code>	Elemento DOM que iniciou o evento
<code>pageX</code>	Posição X do rato tendo em conta o limite esquerdo do <i>viewport</i>
<code>pageY</code>	Posição Y do rato tendo em conta o limite superior do <i>viewport</i>
<code>timeStamp</code>	Número de milisegundos passados desde 1 jan 1970 até ser disparado o evento

Métodos	Descrição
<code>.preventDefault()</code>	Evita o funcionamento por defeito (ex., do <i>submit</i> do <i>form</i>)
<code>.stopPropagation()</code>	Para a propagação do evento pelos elementos ascendentes

AJAX

JavaScript
jQuery

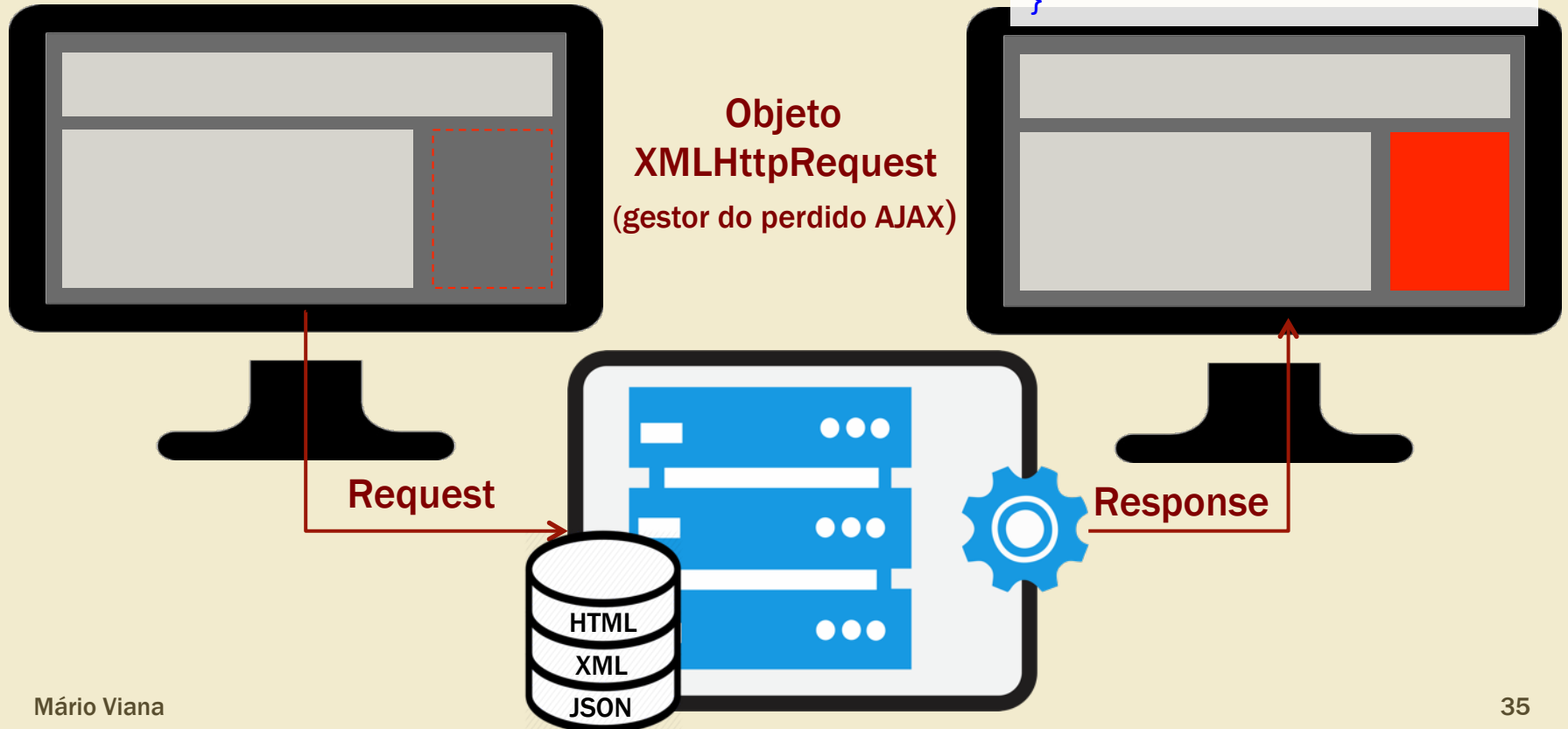
AJAX

- **AJAX é uma técnica que permite carregar dados para uma parte da página sem ser necessário refrescar o seu conteúdo total;**
- **Tipicamente os servidores devolvem ficheiros HTML, XML ou JSON (utilizados para obter informação parcelar da página);**
- ***jQuery* simplifica o processo de efetuar pedidos AJAX e de tratar os dados devolvidos;**
- **Esta técnica é utilizada, por exemplo, em:**
 - **Autocomplete; avisos rápidos; validação antes de completar a inserção de dados, preenchimento/modificação parcelar da página**

AJAX – COMO FUNCIONA?

```
var xhr = new XMLHttpRequest();  
xhr.open('GET', 'data/test.json', true);  
xhr.send('search=arduino');
```

```
xhr.onload = function(){  
  if(xhr.status === 200)  
    // código  
}
```



FORMATO DE DADOS - HTML

```
<div class="event">
  
  <p>
    <b>São Francisco, CA</b>
    <br/>1 de Maio de 2106
  </p>
</div>
<div class="event">
  
  <p>
    <b>Austin, TX</b>
    <br/>15 de Maio de 2106
  </p>
</div>
<div class="event">
  
  <p>
    <b>Nova York, NY</b>
    <br/>30 de Maio de 2106
  </p>
</div>
```

FORMATO DE DADOS - XML

```
<?xml version="1.0" encoding="UTF-8"?>
<eventos>
  <evento>
    <local>São Francisco, CA</local>
    <data>1 de Maio de 2016</data>
    <mapa>imagens/map-ca.png</mapa>
  </evento>
  <evento>
    <local>Austin, TX</local>
    <data>15 de Maio de 2016</data>
    <mapa>imagens/map-tx.png</mapa>
  </evento>
  <evento>
    <local>Nova York, NY</local>
    <data>30 de Maio de 2016</data>
    <mapa>imagens/map-ny.png</mapa>
  </evento>
</eventos>
```

FORMATO DE DADOS - JSON

```
{
  "eventos": [
    {
      "local": "São Francisco, CA",
      "data": "1 de Maio 2016",
      "mapa": "imagens/map-ca.png"
    },
    {
      "local": "Austin, TX",
      "data": "15 de Maio 2016",
      "mapa": "imagens/map-tx.png"
    },
    {
      "local": "Nova York, NY",
      "data": "30 de Maio 2016",
      "mapa": "imagens/map-ny.png"
    }
  ]
}
```

JQUERY E AJAX: PEDIDOS

Método	Descrição
<code>.load()</code>	Carrega fragmento HTML para dentro de um elemento. Sintaxe: <code>\$(seletor).load('url framentoHTML')</code>
<code>\$.get()</code>	Carrega dados utilizando o método HTTP GET. Utilizado para obter dados do servidor. Sintaxe: <code>\$.get(url [,dados] [,callback][,tipo])</code>
<code>\$.post()</code>	Carrega dados utilizando o método HTTP POST. Utilizado para enviar dados para atualização no servidor. Sintaxe: <code>\$.post(url [,dados] [,callback][,tipo])</code>
<code>\$.getJSON()</code>	Carrega dados JSON utilizando o método GET. Utilizado para dados JSON. Sintaxe: <code>\$.getJSON(url [,dados] [,callback])</code>
<code>\$.getScript()</code>	Carrega e executa dados JavaScript utilizando o método GET. Utilizado para dados JavaScript (p. ex. JSONP) Sintaxe: <code>\$.getScript(url [,callback])</code>
<code>\$.ajax()</code>	Método utilizado para executar qualquer pedido. O método anteriores utilizam este método no seu interior.

JQUERY E AJAX: RESPOSTAS

- Para os método `.load()` , o HTML devolvido é inserido no elemento da seleção *jQuery*.
- Para os restantes, é necessário especificar o que fazer manipulando os dados devolvidos utilizando o objeto *jqXHR*.

Propriedades jqXHR	Descrição
responseText	Dados devolvidos em formato texto
responseXML	Dados devolvidos em formato XML
status	Código de estado da resposta
statusText	Descrição do estado

Métodos jqXHR	Descrição
.done()	Código a executar caso o pedido foi bem sucedido
.fail()	Código a executar caso o pedido não foi bem sucedido
.always()	Código a executar independentemente do estado do pedido
.abort()	Para a comunicação

JQUERY E AJAX

DEFINIÇÕES DO MÉTODO \$.AJAX()

- O método \$.ajax() permite a partir de um conjunto de definições obter um grande controlo sobre o pedido Ajax.

Definição	Descrição
type	Pode assumir os valores GET e POST conforme o tipo de pedido
url	Endereço da página que deve ser satisfeita pelo pedido
data	Dados que foram enviados pelo servidor tendo em conta o pedido <i>Ajax</i>
success	Função que possui o código caso o pedido <i>Ajax</i> foi concluído com sucesso
error	Função que possui o código caso o pedido <i>Ajax</i> não teve sucesso
beforeSend	Função que será executada antes do pedido <i>Ajax</i> ter começado (p. ex: <i>icon loading</i>)
complete	Função que será executada depois do pedido ter sido satisfeito
timeout	Número de milissegundos a esperar antes de disparar o evento de erro.