

Introdução às tecnologias Web - ITW

Aula 8 – jQuery /

jQueryUI / Bootstrap Javascript / AJAX / JSON



Sumário

A biblioteca jQuery
AJAX, JSON



A biblioteca jQueryUI



Os componentes javascript da biblioteca Bootstrap



O que o jQuery

jQuery é uma biblioteca JavaScript multi-plataforma projetada para simplificar a programação (*scripting*) do lado do cliente de HTML.

A sintaxe do jQuery foi projetada para tornar mais fácil a navegação nos elementos de um documento. Exemplos:

- * selecionar elementos DOM
- * criar animações,
- * manipular eventos e
- * desenvolver aplicações Ajax.

Vantagens da utilização de jQuery

Separação entre o Javascript e o HTML

Ao invés de usar atributos HTML para identificar as funções para manipulação de eventos, o jQuery lida com eventos puramente em JavaScript.

Deste modo, as tags HTML e o código Javascript são completamente separados.

Em javascript:

```
<form id="target" action="http://192.168.160.36/FormEcho.aspx">
  <input type="submit" value="Enviar" class="btn btn-default" onclick="return validateForm()" />
</form>
```

```
function validateForm() {
  // TODO ...
}
```

Em jQuery

```
<form id="target" action="http://192.168.160.36/FormEcho.aspx">
  <input type="submit" value="Enviar" class="btn btn-default" />
</form>
```

```
$("#target").submit(function (event) {
  // TODO ...
});
```

Vantagens da utilização de jQuery

Elimina incompatibilidades entre navegadores:

Os motores de Javascript dos diferentes navegadores diferem ligeiramente, de modo que o código Javascript que funciona para um navegador pode não funcionar em outro.

O jQuery lida com todas essas inconsistências entre browsers e fornece uma interface consistente que funciona nos diferentes navegadores.

Extensível:

O jQuery é muito extensível – através a adição de novas livrarias ao projeto.

Novos eventos, elementos e métodos podem ser facilmente adicionados e depois reutilizados como um plugin.

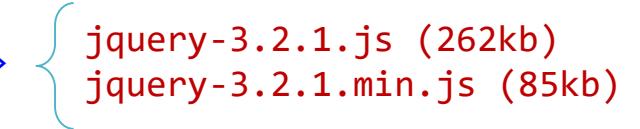
Utilização da biblioteca jQuery

A biblioteca jQuery é um único ficheiro JavaScript, contendo todas as suas funcionalidades: acesso aos elementos DOM, eventos, efeitos e funções comuns do Ajax.

Esse ficheiro pode ser incluído numa página web através da ligação a uma cópia local, desde que previamente descarregada, ou interligando-o a uma das muitas cópias disponíveis a partir de servidores públicos.

Ficheiro local:

```
<script src="Scripts/jquery-3.2.1.min.js"></script>
```



jquery-3.2.1.js (262kb)
jquery-3.2.1.min.js (85kb)

Ficheiro remoto:

```
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

- 
- Lista de livrarias suportadas pela google:
<https://developers.google.com/speed/libraries/>

Sintaxe jQuery

A sintaxe jQuery foi feita a pensar especialmente na seleção de elemento(s) HTML e na execução de alguma ação sobre o(s) mesmo(s).

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

Um sinal `$` para definir / aceder à biblioteca jQuery

um (`seletor`) para "consultar/encontrar" elementos HTML no documento

Uma `ação` jQuery () a ser executada no(s) elemento(s)

Seletores jQuery (1)

Os seletores jQuery são usados para “encontrar” (ou selecionar) elementos HTML baseados no nome, id, classes, tipos, atributos, valores de atributos e muito mais.

Exemplo:

```
<p>Este é um parágrafo.</p>
<button type="button"><i class="fa fa-edit"></i></button>
```

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

Neste exemplo, o seletor usado é apenas o nome do marcador html.
Quando qualquer **button** for carregado, todos os elementos html do tipo **<p>...</p>** serão escondidos **.hide()**.

Seletores jQuery (2)

O seletor jQuery **#id** usa o atributo id de uma tag HTML para localizar o elemento específico no documento.

Como sabemos, um id deve ser único dentro de uma página; assim, este seletor deve ser utilizado para encontrar um elemento único.

Exemplo:

```
<div id="errorMessage">O nome deve ter pelo menos três letras.</div>
<button type="button"><i class="fa fa-edit"></i></button>
```

```
<script>
$(document).ready(function () {
    $("button").click(function () {
        $("#errorMessage").show();
    });
});
</script>
```

Seletores jQuery (3)

O seletor de classe jQuery localiza elementos com uma classe específica.

Para encontrar elementos com uma classe específica, escreva ":"+ nome da classe.

Exemplo:

```
<div class="bannerTop"></div>
```

Neste caso, depois do documento estar carregado no browser, são alteradas propriedades CSS da classe bannerTop. Isso é feito através da chamada ao método/ação **.css()**

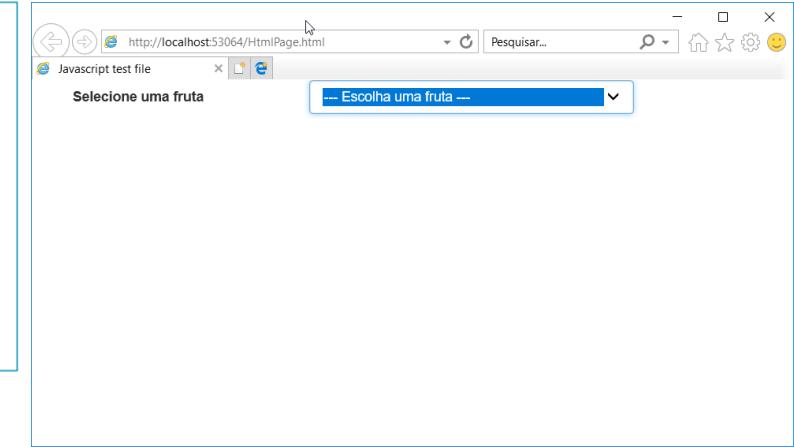
```
<script>
$(document).ready(function () {
    $(".bannerTop").css({"background-color": "yellow",
        "font-size": "200%"});
});
</script>
```

Seletores jQuery (4)

O seletor `option:selected` permite selecionar a opção selecionada numa caixa de seleção

```
<label for="fruta" class="col-md-4 form-control-static">Selecione uma fruta</label>
<div class="col-md-6">
  <select id="fruta" class="form-control">
    <option>--- Escolha uma fruta ---</option>
    <option value="1">Banana</option>
    <option value="2">Maçã</option>
    <option value="3">Pera</option>
  </select>
</div>
```

```
$(document).ready(function () {
  $("#fruta").change(function () {
    var retVal = $("#fruta option:selected").val() +
      " - " +
      $("#fruta option:selected").text();
    alert(retVal)
  });
});
```



Este evento é ativado sempre que o utilizador altera a seleção (evento `.change()`).
 O valor selecionado é recolhido através do método `.val()`;
 O texto da opção é recolhido através do método `.text()`

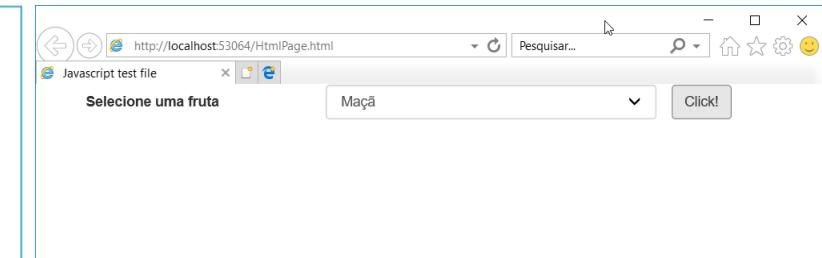
Seletores jQuery (5)

Nos exemplos anteriores os seletores foram quase sempre utilizados para verificar os valores dos diversos elementos html.

O jQuery pode também ser usado para atuar e alterar os elementos.

Exemplo:

```
<label for="fruta" class="col-sm-4 form-control-static">Selecione uma fruta</label>
<div class="col-sm-6">
    <select id="fruta" class="form-control">
        <option>--- Escolha uma fruta ---</option>
        <option value="1">Banana</option>
        <option value="2">Maçã</option>
        <option value="3">Pera</option>
    </select>
</div>
<div class="col-sm-2">
    <button type="button" id="myButton" class="btn btn-default">Click!</button>
</div>
```



```
$(document).ready(function () {
    $("#myButton").click(function () {
        $("#fruta").val(2);
    });
});
```

Outros seletores jQuery

| Sintaxe | Descrição |
|--|---|
| <code>\$('*')</code> | Seleciona todos os elementos do documento |
| <code>\$(this)</code> | Seleciona o elemento html que está a ser manipulado |
| <code>\$("p.intro")</code> | Seleciona todos os elementos <code><p></code> com <code>class="intro"</code> |
| <code>\$("p:first")</code> | Seleciona o primeiro elemento <code><p></code> do documento |
| <code>\$("ul li:first")</code> | Seleciona o primeiro elemento <code></code> do primeiro elemento <code></code> |
| <code>\$("ul li:first-child")</code> | Seleciona o primeiro elemento <code></code> de todos os elementos <code></code> |
| <code>\$("[href]")</code> | Seleciona todos os elementos que possuam o atributo "href" |
| <code>\$("a[target='_blank']")</code> | Seleciona todos os elementos <code><a></code> com o atributo target igual a " <code>_blank</code> " |
| <code>\$("a[target!='_blank']")</code> | Seleciona todos os elementos <code><a></code> com o atributo target diferente de " <code>_blank</code> " |
| <code>\$(":button")</code> | Seleciona todos os elementos <code><button></code> e/ou <code><input></code> com <code>type="button"</code> |
| <code>\$("tr:even")</code> | Seleciona todas as linhas pares dos elementos <code><tr></code> |
| <code>\$("tr:odd")</code> | Seleciona todas as linhas ímpares dos elementos <code><tr></code> |

O evento `$(document).ready()` (1)

Como pudemos observar nos exemplos apresentados nos slides anteriores, todos os métodos de jQuery estão sempre dentro de um evento `$(document).ready()`

Isso evita que qualquer código jQuery seja executado antes do documento carregar completamente (ou seja, só depois do documento estar pronto - **is ready**).

Também pode ser utilizada outra notação equivalente:

`$(()).ready()` ou `$("document").ready()`

O evento `$(document).ready()` (2)

É uma boa prática esperar que o documento seja totalmente carregado e esteja pronto antes de atuar nele.

Exemplos de ações que podem falhar se os métodos forem executados antes que o documento seja totalmente carregado:

Tentar ocultar um elemento que ainda não foi criado;

Tentar obter o tamanho de uma imagem que ainda não está carregada; etc..

Eventos jQuery

O jQuery foi feito para responder a eventos numa página HTML.

O que é um Evento?

Um evento representa o momento em que algo acontece.

Todas as ações que um visitante realiza numa página da web é chamado de evento.

Exemplos de eventos numa página web:

Mover o rato sobre um elemento;

Selecionar um botão de opção;

Clicar num elemento.

Categorias de eventos jQuery

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

Sintaxe jQuery para métodos associados com eventos

Em jQuery, a maioria dos eventos DOM tem um método jQuery equivalente. Para atribuir um evento de clique a todos os parágrafos de uma página, por exemplo, pode fazer o seguinte:

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

O próximo passo deve definir o que acontece quando o evento é disparado. Assim, deve ser indicada uma function ao evento.

```
<script>
  $(document).ready(function(){
    $("p").click(function(){
      // o que será feito deve ser programado aqui!!
    });
  });
</script>
```

Neste exemplo gerimos de dois eventos: o **document.ready()** e ainda o **click()** dos elementos do tipo **<p>...</p>**

Manipulação do DOM com jQuery - GET

Os três métodos para manipulação de elementos DOM são:

text() - Retorna o conteúdo de texto dos elementos selecionados;

html() - Retorna o conteúdo dos elementos selecionados;

val() - Retorna o valor dos campos de um formulário (<input>).

O método **attr()** é usado para obter valores de atributos. O exemplo a seguir demonstra como obter o valor do atributo href em um link:

```
$(document).ready(function(){
    $("button").click(function () {
        alert($("#nextPage").attr("href"));
    });
});
```

Manipulação do DOM com jQuery - SET

Os métodos são os mesmos: text(), html(), val() e attr() mas a sintaxe é distinta.

Exemplos:

```
$(document).ready(function(){
    $("button").click(function () {
        $("title").text("This is a text");
        $("#pageTitle").html("This is a <strong>text</strong>");
        $("#name").val("Dolly Duck");
        $("#nextPage").attr("href", "http://www.ua.pt");
    });
});
```

JQuery - Obter e definir classes CSS

JQuery tem vários métodos para manipulação CSS. Examinaremos os seguintes métodos:

addClass() - Adiciona uma ou mais classes aos elementos selecionados;

removeClass() - Remove uma ou mais classes dos elementos selecionados;

toggleClass() - Alterna entre adicionar / remover classes dos elementos selecionados;

css() - Define ou retorna o atributo de estilo.

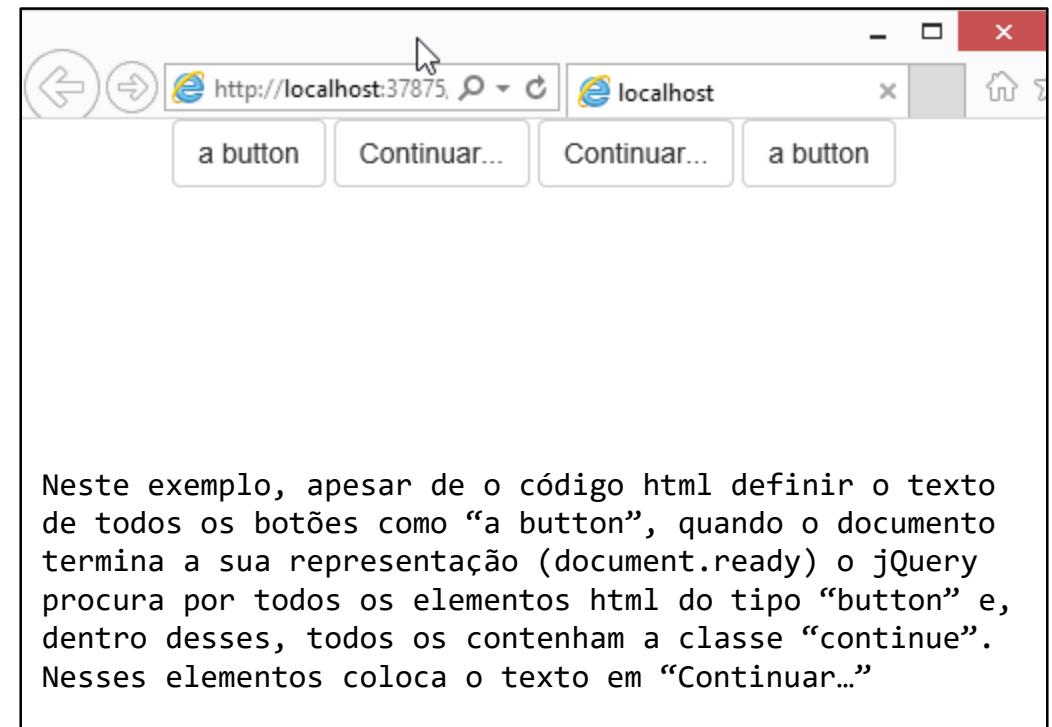
```
<script>
$(document).ready(function () {
    $(".bannerTop").css({"background-color": "yellow", "font-size": "200%"});
});
</script>
```

Exemplos de manipulação do DOM (Document Object Model)

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <div class="container text-center">
        <button class="start btn btn-default">a button</button>
        <button class="continue btn btn-default">a button</button>
        <button class="continue btn btn-default">a button</button>
        <button class="end btn btn-default">a button</button>
    </div>
    <script src="Scripts/jquery-2.1.4.min.js"></script>
    <script>
        $(document).ready(function() {
            $("button.continue").html("Continuar...");
        });
    </script>
</body>
</html>

```



Notas importantes:

- O manuseamento do evento document.ready
- A instrução jQuery começa com o caráter \$

Exemplos de manipulação do DOM (Document Object Model)

Formulário de validação do formulário da Aula 7 - Versão javascript e versão jQuery

```
function testFormValid() {
    var retVal = true;
    var PersonData_FirstName = document.getElementById("PersonData_FirstName");
    var PersonData_FirstName_Error = document.getElementById("PersonData_FirstName_Error");
    if (PersonData_FirstName.value.trim().length < 3) {
        if (PersonData_FirstName_Error.classList.contains("no_error"))
            PersonData_FirstName_Error.classList.remove("no_error");
        if (!PersonData_FirstName_Error.classList.contains("error"))
            PersonData_FirstName_Error.classList.add("error");
        retVal = false;
    }
    else {
        if (PersonData_FirstName_Error.classList.contains("error"))
            PersonData_FirstName_Error.classList.remove("error");
        if (!PersonData_FirstName_Error.classList.contains("no_error"))
            PersonData_FirstName_Error.classList.add("no_error");
    }
    return retVal;
}
```



```
$(document).ready(function () {
    $('form').submit(function (event) {
        var retVal = true;
        if ($.trim($('#PersonData_FirstName').val()).length < 3) {
            if ($('#PersonData_FirstName_Error').hasClass("no_error"))
                $('#PersonData_FirstName_Error').removeClass("no_error").addClass("error");
            retVal = false;
        }
        else {
            if ($('#PersonData_FirstName_Error').hasClass("error"))
                $('#PersonData_FirstName_Error').removeClass("error").addClass("no_error");
        }
        return retVal;
    });
});
```



AJAX (Asynchronous JavaScript and XML)

AJAX é a arte de trocar dados com um servidor e atualizar partes de uma página da Web - sem recarregar a página inteira.

Através da utilização de AJAX, o carregamento de dados é feito em segundo plano e exibido na página da Web, sem recarregar a página inteira.

Exemplos de aplicativos usando o AJAX: Gmail, Google Maps, Facebook...



JSON - JavaScript Object Notation.

JSON é um formato leve de intercâmbio de dados que é independente da linguagem de programação utilizada, auto-descritiva e, por isso, fácil de entender.

Usa a sintaxe JavaScript, mas o formato JSON é somente texto. O texto pode ser lido e usado como formato de dados por qualquer linguagem de programação.

Exemplo:

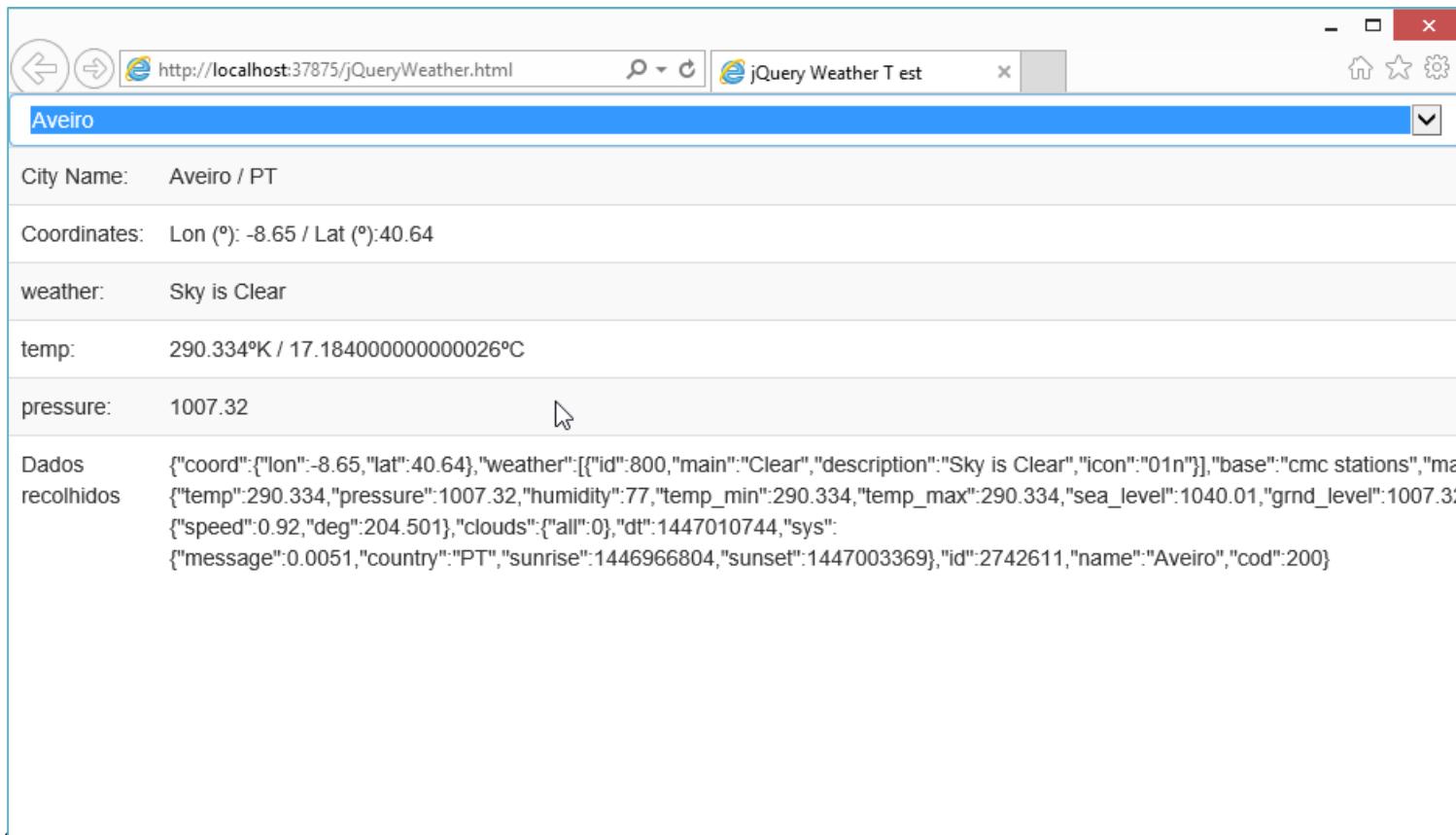
```
<html>
<body>
    <h2>Manipulação JSON em JavaScript</h2>
    <p id="demo"></p>
    <script>
        var text = '{"name":"Zé Maria Pincel","address":"Rua 8 de Maio, 5","phone":"351 123456789"}';
        var obj = JSON.parse(text);
        document.getElementById("demo").innerHTML =
            obj.name + "<br>" +
            obj.address + "<br>" +
            obj.phone;

    </script>
</body>
</html>
```

Manipulação do DOM + AJAX

(jQuery + Asynchronous JavaScript And XML + JSON)

Exemplo: Elaborar um formulário onde o utilizador pode ir buscar, remotamente e sem voltar a carregar o documento as condições meteorológicas de uma qualquer cidade no mundo!



The screenshot shows a web browser window titled "jQuery Weather Test". The address bar indicates the URL is <http://localhost:37875/jQueryWeather.html>. The page displays weather information for the city "Aveiro". The data is presented in a table-like structure:

| Aveiro | |
|---------------------|--|
| City Name: | Aveiro / PT |
| Coordinates: | Lon (°): -8.65 / Lat (°): 40.64 |
| weather: | Sky is Clear |
| temp: | 290.334°K / 17.184000000000026°C |
| pressure: | 1007.32 |
| Dados recolhidos | <pre>{"coord":{"lon":-8.65,"lat":40.64},"weather":[{"id":800,"main":"Clear","description":"Sky is Clear","icon":"01n"}],"base":"cmc stations","main":{"temp":290.334,"pressure":1007.32,"humidity":77,"temp_min":290.334,"temp_max":290.334,"sea_level":1040.01,"grnd_level":1007.32,"speed":0.92,"deg":204.501}, "clouds":{"all":0}, "dt":1447010744, "sys": {"message":0.0051, "country": "PT", "sunrise":1446966804, "sunset":1447003369}, "id":2742611, "name": "Aveiro", "cod":200}</pre> |



Quem fornece a informação do tempo?

<http://api.openweathermap.org/>

Erro

```
{"cod": "404", "message": "Error: Not found city"}
```

Sucesso

```
{"coord": {"lon": -8.61, "lat": 41.15}, "weather": [{"id": 802, "main": "Clouds", "description": "scattered clouds", "icon": "03n"}], "base": "stations", "main": {"temp": 301.15, "pressure": 1018, "humidity": 58, "temp_min": 301.15, "temp_max": 301.15}, "visibility": 16093, "wind": {"speed": 7.2, "deg": 50}, "clouds": {"all": 40}, "dt": 1447008780, "sys": {"type": 1, "id": 819, "message": 0.03, "country": "PT", "sunrise": 1446966858, "sunset": 1447003295}, "id": 2735943, "name": "Porto", "cod": 200}
```

<http://api.openweathermap.org/data/2.5/weather?q=oporto,pt&appid=b2b1df463182c3cca5276e9d3267cc95>
(#valid@13nov16)

Mensagem de sucesso “beautified”

```
{  
    "coord": {  
        "lon": -8.61,  
        "lat": 41.15  
    },  
    "weather": [  
        {  
            "id": 802,  
            "main": "Clouds",  
            "description": "scattered clouds",  
            "icon": "03n"  
        }  
    ],  
    "base": "stations",  
    "main": {  
        "temp": 301.15,  
        "pressure": 1018,  
        "humidity": 58,  
        "temp_min": 301.15,  
        "temp_max": 301.15  
    },  
    "visibility": 16093,  
    "wind": {  
        "speed": 7.2,  
        "deg": 50  
    },  
    "clouds": {  
        "all": 40  
    },  
    "dt": 1447008780,  
    "sys": {  
        "type": 1,  
        "id": 819,  
        "message": 0.03,  
        "country": "PT",  
        "sunrise": 1446966858,  
        "sunset": 1447003295  
    },  
    "id": 2735943,  
    "name": "Porto",  
    "cod": 200  
}
```

{JSON}

JavaScript Object Notation

O código da interface

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery Weather Test</title>
    <link href='https://fonts.googleapis.com/css?family=Roboto:300' rel='stylesheet' type='text/css'>
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <div class="container">
        <select id="citySelector" class="form-control">
            <option value="">Select a city name</option>
            <option value="Aveiro, PT">Aveiro</option>
            <option value="Oporto, PT">Porto</option>
            <option value="Paris, FR">Paris</option>
            <option value="London, UK">Londres</option>
            <option value="New York, USA">Nova Iorque</option>
        </select>
        <table class="table table-striped hidden">
            <tr><td class="col-xs-2">City Name:</td><td class="col-xs-10" id="cityName"></td></tr>
            <tr><td class="col-xs-2">Coordinates:</td><td class="col-xs-10" id="coordinates"></td></tr>
            <tr><td class="col-xs-2">weather:</td><td class="col-xs-10" id="weather"></td></tr>
            <tr><td class="col-xs-2">temp:</td><td class="col-xs-10" id="temp"></td></tr>
            <tr><td class="col-xs-2">pressure:</td><td class="col-xs-10" id="pressure"></td></tr>
            <tr><td class="col-xs-2">Dados recolhidos </td><td class="col-xs-10" id="allData"></td></tr>
        </table>
    </div>

```

O código de manipulação da informação

```
<script src="Scripts/jquery-3.3.1.min.js"></script>
<script>
$(document).ready(function () {
    $("#citySelector").change(function () {
        $.ajax({
            url: "http://api.openweathermap.org/data/2.5/weather",
            data: {
                q: $("#citySelector").val(),
                APPID: 'b2b1df463182c3cca5276e9d3267cc95'
            },
            success: function (data) {
                if (data.name){
                    $('table').removeClass('hidden');
                    $("#cityName").html(data.name + ' / ' + data.sys.country);
                    $("#coordinates").html('Lon (°): ' + data.coord.lon + ' / Lat (°):' + data.coord.lat);
                    $("#weather").html(data.weather[0].description);
                    $("#temp").html(data.main.temp.toString() + '°K / ' + (data.main.temp - 273.15).toString() + '°C');
                    $("#pressure").html(data.main.pressure);
                    $("#allData").html(JSON.stringify(data));
                }
                else {
                    $('table').addClass('hidden');
                    alert(data.message);
                }
            },
            error: function () {
                $('table').addClass('hidden');
                alert('Erro!');
            }
        });
    });
});
</script>
</body>
```

13/11/2017

©2014-17, JOAQUIM SOUSA PINTO



JavaScript Object Notation



Asynchronous Javascript And XML



29

jQueryUI – jQuery User Interface

jQuery UI é uma coleção de widgets de interface gráfica, efeitos visuais animados e temas implementados com jQuery, CSS's e HTML

- um widget é um pequeno aplicativo com funcionalidade limitada que pode ser instalado e executado dentro de uma página web

Esta livraria assenta sobre a livraria jQuery.

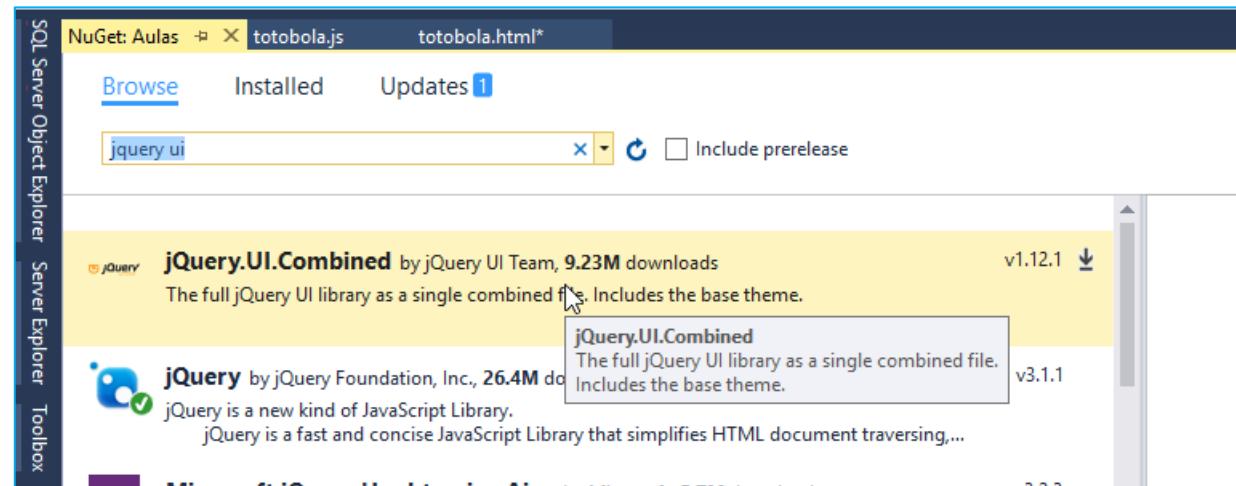
Utilização da biblioteca jQuery UI

A jQueryUI possui muitas funcionalidades que são também cobertas pelo Bootstrap.

Assim, esta livraria será abordada de modo genérico e identificadas apenas algumas funcionalidades que não cobertas pelo Bootstrap.

Utilização:

```
<link rel="stylesheet" href=" https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css">
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```



Acesso ao jQuery UI
no Visual Studio

jQuery UI Widgets

Acordeão – grupo de conteúdos organizados na forma de um acordeão

Autocomplete - caixas que permitem o preenchimento automático com base no que o utilizador digita

Button - botão com apresentação melhorada.

Permite que botões rádio e caixas de seleção sejam convertidos em botões

Datepicker – componente com calendário para recolha de campos com datas

Dialog - caixas de diálogo colocadas em cima de outros conteúdos

Menu – componente que permite mostrar e gerir os elementos de um menu

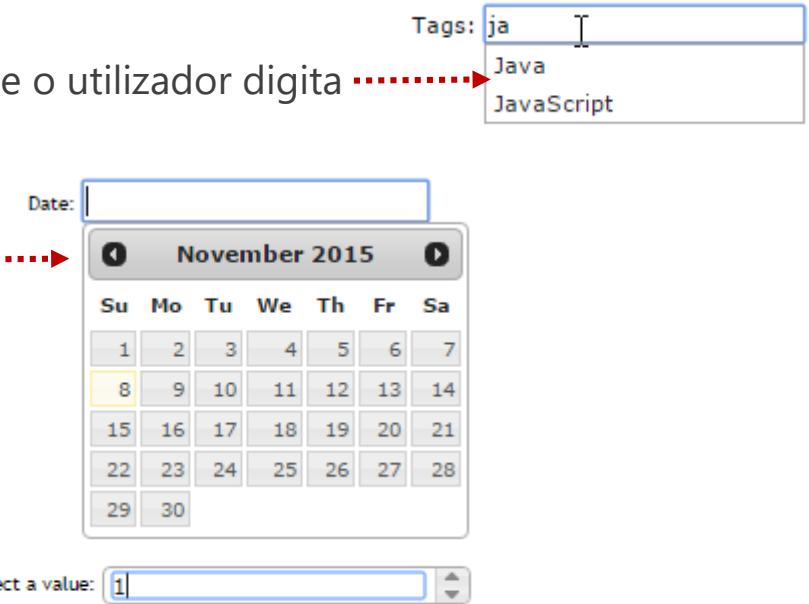
Progressbar - barras de progresso – animadas, ou não

Slider – barras de arrastamento totalmente personalizáveis

Spinner – gera o valor de um número com setas

Tabs - manipulação interface com tabuladores

Tooltip - Mostrar uma dica sobre um determinado conteúdo ou operação



Exemplos DOM

Para uma lista completa de widgets jQuery UI, ver <http://jqueryui.com/widget/>

```
<div id="draggable" class="ui-widget-content">
  <p>Drag me around</p>
</div>
```

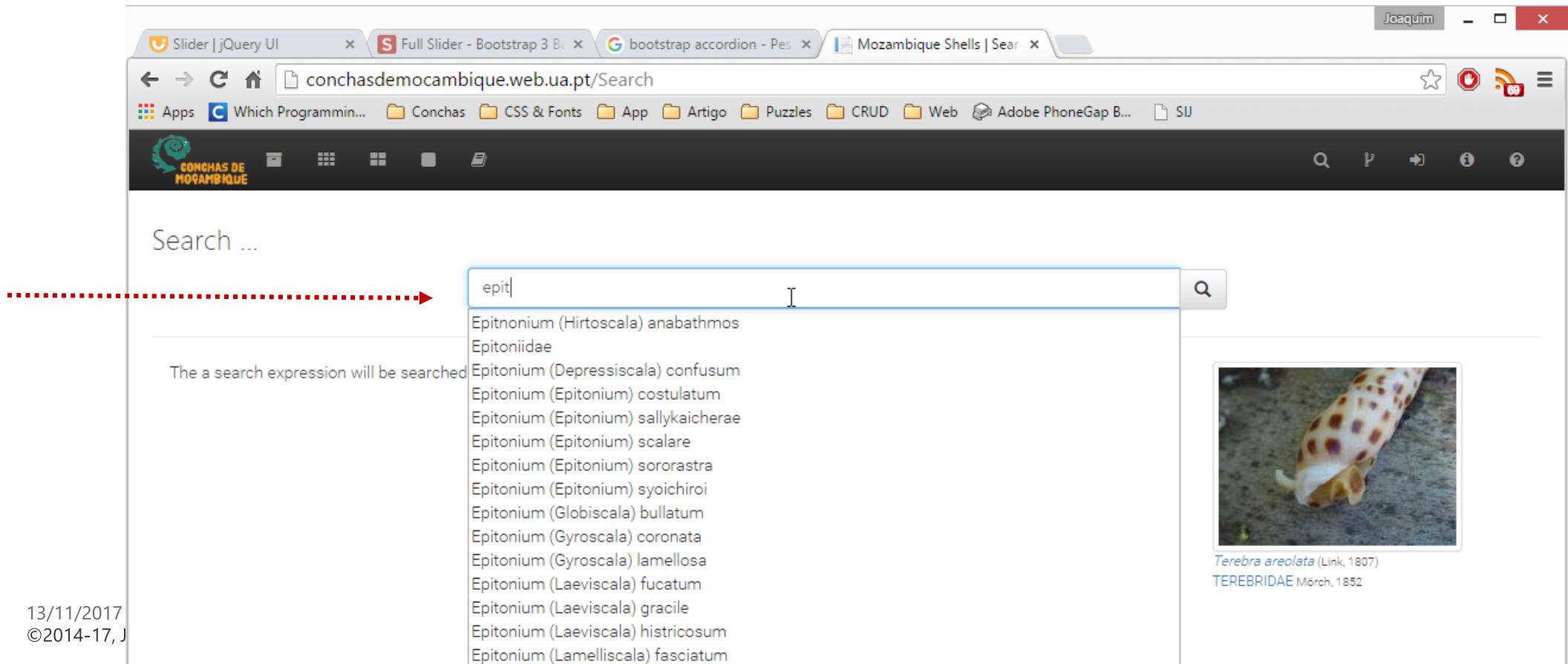
```
<script type="text/javascript">
// Make #draggable draggable
$(function () {
  $("#draggable").draggable();
});
</script>
```

```
<div id="demo">
  Some text
</div>
```

```
<script type="text/javascript">
// change css of an element
$(function () {
  $("#demo").css('color', '#069');
});
</script>
```

Exemplo de autocomplete

Este exemplo foi retirado do site <http://conchasdemocambique.web.ua.pt> e permite ao utilizador procurar pelo nome de uma classe, família, subfamília ou espécie de conchas.



The screenshot shows a web browser window with the title bar "Joaquim". The address bar contains the URL "conchasdemocambique.web.ua.pt/Search". The page itself has a dark header with a logo for "CONCHAS DE MOÇAMBIQUE". A search input field is present with the placeholder "Search ...". Below it, a message says "The a search expression will be searched". A red arrow points from the text "The a search expression will be searched" to the search input field. An autocomplete dropdown menu is open, showing a list of suggestions starting with "epit". The suggestions include various species names like "Epitonium (Hirtoscala) anabathmos", "Epitoniiidae", and "Epitonium (Epitonium) scalare". To the right of the dropdown, there is a small image of a brown and white patterned shell and some descriptive text: "Terebra areolata (Link, 1807)" and "TEREBRIDAE Mörch, 1852".

Slider | jQuery UI S Full Slider - Bootstrap 3 Bi... G bootstrap accordion - Pes... Mozambique Shells | Search

conchasdemocambique.web.ua.pt/Search

Apps Which Programmin... Conchas CSS & Fonts App Artigo Puzzles CRUD Web Adobe PhoneGap B... SJU

CONCHAS DE MOÇAMBIQUE

Search ...

The a search expression will be searched

epit

- Epitonium (Hirtoscala) anabathmos
- Epitoniiidae
- Epitonium (Depressiscala) confusum
- Epitonium (Epitonium) costulatum
- Epitonium (Epitonium) sallykaicherae
- Epitonium (Epitonium) scalare
- Epitonium (Epitonium) sororastra
- Epitonium (Epitonium) syoichiroi
- Epitonium (Globiscala) bullatum
- Epitonium (Gyroscala) coronata
- Epitonium (Gyroscala) lamellosa
- Epitonium (Laeviscalata) fucatum
- Epitonium (Laeviscalata) gracile
- Epitonium (Laeviscalata) histricosum
- Epitonium (Lamelliscala) fasciatum

Terebra areolata (Link, 1807)
TEREBRIDAE Mörch, 1852

O código

Nota:

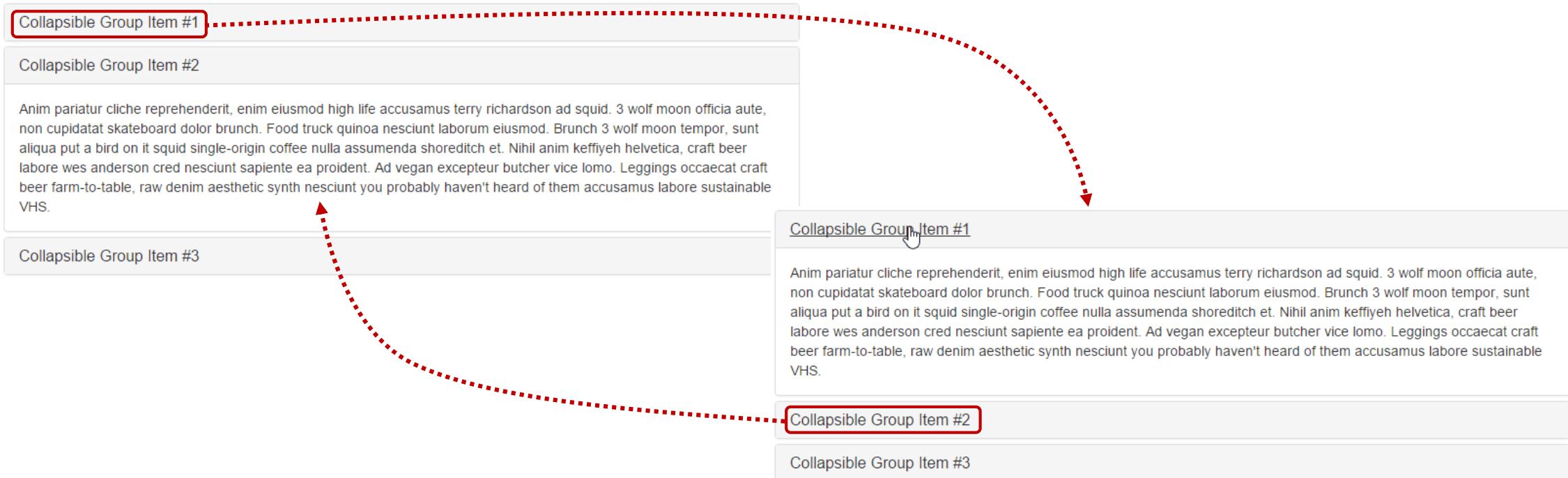
Este exemplo não é repetível fora do contexto porque o webService que serve a pesquisa ([DynamicShellSearch.asmx/SearchData](#)) está, intencionalmente, protegido de modo a só poder ser utilizado pelo próprio site.



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
    <link href="Content/themes/ui-darkness/jquery.ui.base.css" rel="stylesheet" />
</head>
<body>
    <div class="container">
        <input class="form-control" ID="SearchText" placeholder="Search expression" />
    </div>
    <script src="Scripts/jquery-2.1.4.min.js"></script>
    <script src="Scripts/jquery-ui-1.11.4.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $("#SearchText").autocomplete({
                minLength: 4, ←.....→
                source: function (request, response) {
                    $.ajax({
                        type: "POST",
                        contentType: "application/json; charset=utf-8",
                        url: "DynamicShellSearch.asmx/SearchData",
                        data: "{ 'DName': '" + $('#SearchText').val() + "'}",
                        dataType: "json",
                        success: function (data) {
                            response(data.d);
                        },
                        error: function (result) {
                            alert(result.statusText);
                        }
                    });
                });
            });
        });
    </script>
</body>
</html>
```

Bootstrap Collapse ≈ jQueryUI Accordion

<http://getbootstrap.com/javascript/#collapse>



The diagram illustrates the mapping between Bootstrap's Collapsible Groups and jQueryUI's Accordions. It shows two columns of three items each. The first column represents Bootstrap's Collapsible Groups, and the second column represents jQueryUI's Accordions. Red arrows indicate the correspondence: the top item in each row maps to the other, and the middle item in each row maps to the bottom item in the adjacent row. Specifically:

- Top row: "Collapsible Group Item #1" (Bootstrap) maps to "Collapsible Group Item #1" (jQueryUI).
- Middle row: "Collapsible Group Item #2" (Bootstrap) maps to "Collapsible Group Item #2" (jQueryUI), and "Collapsible Group Item #3" (Bootstrap) maps to "Collapsible Group Item #3" (jQueryUI).
- Bottom row: "Collapsible Group Item #1" (Bootstrap) maps to "Collapsible Group Item #3" (jQueryUI).

Each item contains placeholder text from the 'lorem ipsum' generator.

Bootstrap Buttons ≈ jQueryUI Button

<http://getbootstrap.com/javascript/#buttons>

EXAMPLE

Loading state

```
<button type="button" id="myButton" data-loading-text="Loading..." class="btn btn-primary"
autocomplete="off">
  Loading state
</button>
```

Copy

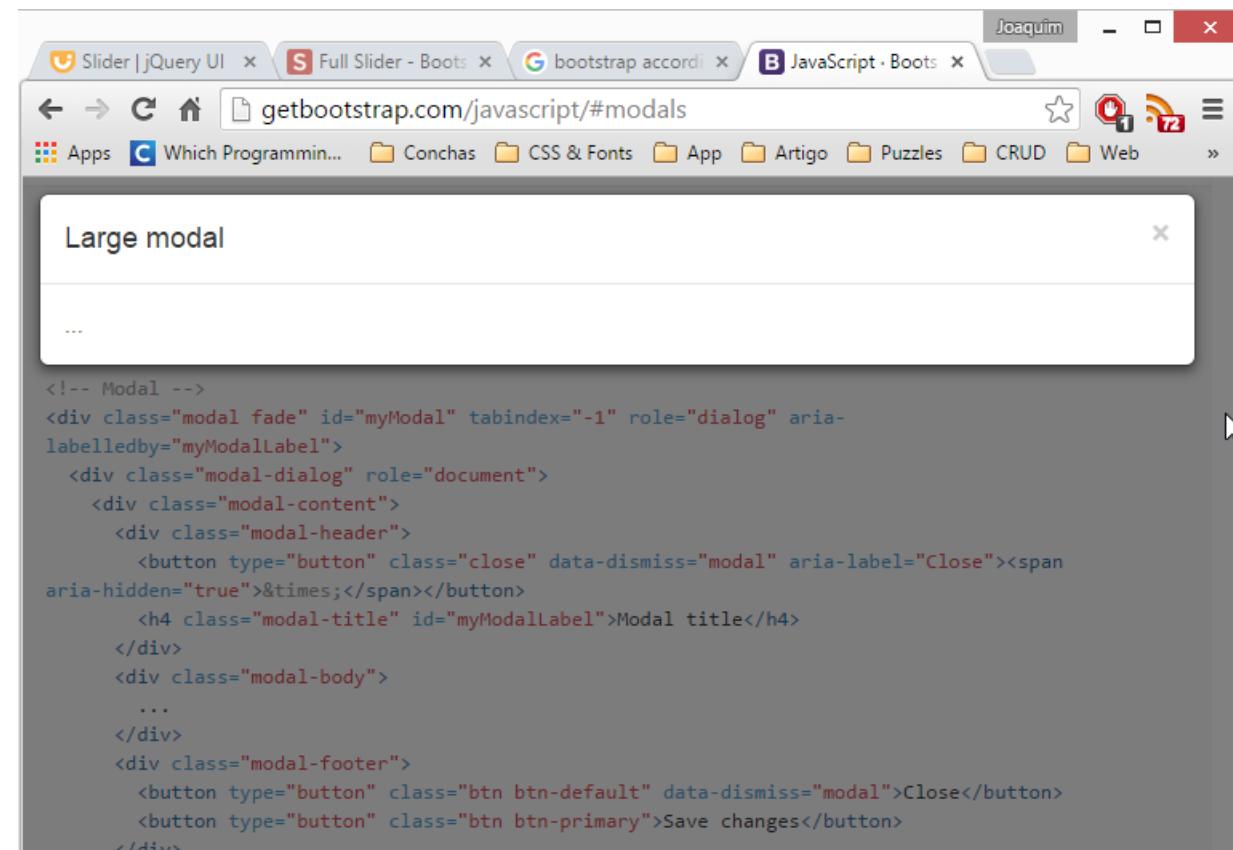
```
<script>
  $('#myButton').on('click', function () {
    var $btn = $(this).button('loading')
    // business logic...
    $btn.button('reset')
  })
</script>
```

Checkbox 1 (pre-checked) Checkbox 2 Checkbox 3

Checkbox 1 (pre-checked) Checkbox 2 Checkbox 3

Bootstrap Modal ≈ jQueryUI Dialog

<http://getbootstrap.com/javascript/#modals>



Bootstrap Tabs ≈ jQueryUI Tabs

<http://getbootstrap.com/javascript/#tabs>



Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui.



Food truck fixie locavore, accusamus mcsweeney's marfa nulla single-origin coffee squid. Exercitation +1 labore velit, blog sartorial PBR leggings next level wes anderson artisan four loko farm-to-table craft beer twee. Qui photo booth letterpress, commodo enim craft beer mlkshk aliquip jean shorts ullamco ad vinyl cillum PBR. Homo nostrud organic, assumenda labore aesthetic magna delectus mollit. Keytar helvetica VHS salvia yr, vero magna velit sapiente labore stumptown. Vegan fanny pack odio cillum wes anderson 8-bit, sustainable jean shorts beard ut DIY ethical culpa terry richardson biodiesel. Art party scenester stumptown, tumblr butcher vero sint qui sapiente accusamus tattooed echo park.

A funcionalidade dos Tabs é muito semelhante à do Accordeon, ou seja, seccionar a quantidade de informação apresentada mas uma faz o seccionamento na vertical (acordeon) enquanto a outra faz na horizontal (Tabs)

Exemplos:

Mozambique Shells | Spec x conchasdemocambique.web.ua.pt/Specie?Shell_ID=646

Identification Bibliography **Images**

Shell ID: 646
 Specie: *Tutufa (Tutufa) bubo* (Linnaeus, 1758)
 Family: BURSIDAE Thiele, 1929
 Author, Year: (Linnaeus, 1758)

Dimensions min-max (mm): 150-200 mm
 Occurrence area: Moçambique
 Habitat: Coral reef
 Depth: Shallow subtidal

Exists in IL collection:
 IL Collection number: 598
 Unidentified:
 Visible:

Last update: 2009/08/25 16:45:39

© 2011-2015 - Conchas de Moçambique

Mozambique Shells | Spec x conchasdemocambique.web.ua.pt/Specie?Shell_ID=646

Identification Bibliography **Images**

Shell owner: Isabel Leitão
 Photographer: Isabel Leitão
 Date taken: 2008/08/31
 Camera: Sony Cybershot
 Details: Default: | Habitat: | Detail: | Copyright:
 Last update: 2015/10/29 13:49:16

Photographer: Isabel Leitão
 Date taken: 2015/10/27
 Camera: Sony Cybershot
 Details: Default: | Habitat: | Detail: | Copyright:
 Last update: 2015/10/30 12:11:25

© 2011-2015 - Conchas de Moçambique
 conchasdemocambique.web.ua.pt/Specie?Shell_ID=646#tabs-3

Bootstrap Tooltip/Popover ≈ jQueryUI Tooltips

<http://getbootstrap.com/javascript/#tooltip>

<http://getbootstrap.com/javascript/#popover>

Tooltip on the left

Tooltip on the top

Tooltip on the bottom

Tooltip on the right

EXAMPLE

Click to toggle popover

Popover title

And here's some amazing content. It's very engaging. Right?

EXAMPLE

Dismissible popover

Dismissible popover

And here's some amazing content. It's very engaging. Right?

Bootstrap Carousel ≈ jQueryUI Slider

<http://getbootstrap.com/javascript/#carousel>

Funcionalidade parcial.

O Bootstrap Carousel já foi introduzido em aula anterior. Estamos agora em condições de perceber “como o programar”.

Exemplo:

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-to="2"></li>
    <li data-target="#carousel-example-generic" data-slide-to="3"></li>
  </ol>
  <!-- Carousel
  ----- -->
  <div id="myCarousel" class="carousel slide">...</div>
  <!-- /.carousel -->
</div>
<script>
  $('.carousel').carousel({
    interval: 10000
  })
</script>
```

O carousel do elemento com a classe carousel, deve mudar de 10.000 e, 10.000 milissegundos, ou seja de 10 em 10 segundos.

Exemplo

