

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.

3

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 ...  
});
```

4

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.

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

5

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 decarregada, 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/>

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

6

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)

7

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 element <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 "_blank"
<code>\$("a[target!='_blank']")</code>	Seleciona todos os elementos <code><a></code> com o atributo target diferente de "_blank"
<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>

13

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

Cmo pudémos 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.

16

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

17

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>`

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

18

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"));
    });
});
```

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

19

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>
```

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...



24

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO



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>
```

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

25

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 livreria assenta sobre a livreria jQuery.

30

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

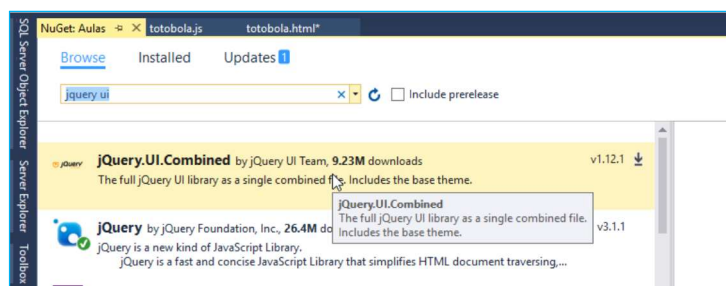
Utilização da biblioteca jQuery UI

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

Assim, esta livreria 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

31

13/11/2017
©2014-17, JOAQUIM SOUSA PINTO

jQuery UI Widgets

Acordeão – grupo de contentores 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 – animandas, ou não

Slider – barras de arrastamento totalmente personalizáveis

Spinner – gere o valor de um número com setas

Tabs - manipulação interface com tabuladores

Tooltip - Mostrar uma dica sobre um determinado conteúdo ou operação

