

CTeSP PSI 2020/21

Introdução às Linguagens e Tecnologias Web

JQuery

Pedro Colarejo

pcolarejo@ua.pt

Escola Superior de Tecnologia e Gestão de Águeda

Universidade de Aveiro

2020/21

O que é?

jQuery: biblioteca de código Javascript, livre e compatível com todos os browsers, que simplifica a realização de tarefas no desenvolvimento Web tais como:

- acesso aos elementos do documento HTML (DOM)
- acesso aos estilos CSS
- tratamento de eventos
- animações e outros efeitos
- interações Ajax
- plugins Javascript
- utilitários



jQuery Website

<http://jquery.com/>

jQuery Learning Center

<http://learn.jquery.com/>

Como utilizar?

- Download da última versão a partir do website jquery.com
 - versão: **compressed** ou **uncompressed**?
- Incluir a última versão a partir de uma Content Delivery Network (CDN) (Google ou Microsoft)
 - **Microsoft CDN:**
`<script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.6.0.js"></script>`
 - **Google CDN:**
`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>`

Quais as vantagens/desvantagens de cada uma das opções?

Javascript vs jQuery

jQuery

Exemplo: criar dinamicamente um novo parágrafo no final de uma página Web.

Código em Javascript:

```
var no = document.createElement('p');  
no.innerHTML = "Texto do novo parágrafo";  
var pai = document.getElementById('body');  
pai.appendChild(no);
```

Código em jQuery:

```
$( 'body' ).append("<p>Texto do novo parágrafo</p>");
```

Sintaxe

- Sintaxe base:

`$(seletor).action()`

- Exemplos:

- **`$(this).hide()`** - esconde o elemento sobre o qual estamos a trabalhar
- **`$(‘p’).hide()`** - esconde todos os elementos do tipo `<p>`
- **`$(‘.test’).hide()`** – esconde todos os elementos com a classe "test"
- **`$(‘#test’).hide()`** – esconde o elemento com o id "test"

Funcionamento

jQuery

- O evento Document Ready:

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

- Exercício 1: Página HTML com uma div escondida.

```
<head>  
  <title>Exemplo jQuery</title>  
  
  <script  
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
  
  <script>  
    $(document).ready(function() {  
      $('div').hide();  
    });  
  </script>  
  
</head>
```

```
<body>  
  <h2>Lista de Emails</h2>  
  <div>  
    <p>joana@ua.pt</p>  
    <p>joao@ua.pt</p>  
  </div>  
</body>
```

Seletores jQuery

- Para além dos seletores baseados nos elementos HTML, #id, .class, existem outros seletores específicos em jQuery
- Existe também a possibilidade de indicar vários seletores em simultâneo.

Exemplo:

`$('div, p').hide()` – esconde todos os elementos do tipo `<div>` e `<p>`



jQuery Selectors

http://www.w3schools.com/jquery/jquery_ref_selectors.asp

Eventos em jQuery



- Adicionar eventos a elementos já existentes na página:

```
$(selector).event(function() {  
    // código relativo à ação  
});
```

Exemplo:

```
$('#button').click(function(){  
    $('#div').show();  
});
```

Exercício 2:

Adicione um botão à página do exercício 1 de forma a que ao ser clicado o conteúdo da div fique visível.

Eventos em jQuery

jQuery

- Para adicionar eventos, tanto a elementos que já existam ou que venham a ser criados após o carregamento da página, utiliza-se o método **on()**:

Exemplo:

```
$(document).on( 'click', '.item', function() {  
    $(this).hide();  
});
```

- Principais eventos:

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	ready
hover	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	select



jQuery Events

http://www.w3schools.com/jquery/jquery_ref_events.asp

Manipulação de Elementos HTML com jQuery



- Principais métodos para **adicionar novos elementos**:
 - append()** – adicionar conteúdo no final do elemento selecionado
 - prepend()** – adicionar conteúdo no início do elemento selecionado
 - after()** – adicionar conteúdo após o elemento selecionado
 - before()** – adicionar conteúdo antes do elemento selecionado
 - attr()** – adicionar novos atributos aos elementos

Exemplos:

```
$('#div p').attr( 'id', 'x' );
```

```
$('#div p').attr( { 'id':'x' , 'class':'x' } );
```

```
$('#x').append( " Novo texto" );
```

```
$('#div').after( "<div><p>Nova caixa de texto</p></div>" );
```

Manipulação de Elementos HTML com jQuery

Exercício 3:

Adicione um novo botão à página (utilizando jQuery) do exercício 2 de forma a que ao ser clicado o conteúdo da div fique invisível.

Exercício 4:

Adicione um novo parágrafo (`<p>sofia@ua.pt</p>`) no início do conteúdo da div (utilizando jQuery) do exercício 3.

Manipulação de Elementos HTML com jQuery

- Principais métodos para **remover elementos**:
empty() – remover o conteúdo do elemento selecionado
remove() – remover o conteúdo e o próprio elemento selecionado

Exemplos:

```
$('#x').remove();
```

```
$('#p').remove( '.x' );
```

```
$('#div').empty();
```

Manipulação de Elementos HTML com jQuery

- Principais métodos para **modificar/obter os elementos e atributos HTML**:

text() – modificar/obter o conteúdo (texto) do elemento selecionado

html() – modificar/obter o conteúdo (html) do elemento selecionado

val() – modificar/obter o valor dos atributos de um dado formulário

Exemplos:

```
$('#x').html( );
```

```
$('#x').text( 'Novo texto' );
```

```
$('#x').html( $('#x').html( ) + 'Novo texto' );
```

```
$('#input').val( );
```

Manipulação de Elementos HTML com jQuery

Exercício 5:

Altere o exercício 4 de forma a que ao clicar sobre um endereço de email seja possível pedir ao utilizador para fornecer um novo endereço substituindo-o.

Exercício 6:

Altere o exercício 5 de forma a incluir uma nova caixa de input de texto que permita ao utilizador fornecer um novo endereço de email o qual será adicionado no início da div.

Manipulação de Estilos CSS com jQuery

- Principais métodos para **adicionar/remover/obter os estilos de elementos**:
addClass() – adicionar classes ao elemento selecionado
removeClass() – remover classes do elemento selecionado
toggleClass() – alterna entre adicionar/remover classes ao elemento
css() – adicionar/modificar/obter as propriedades dos estilos do elemento

Exemplos:

```
$('#div').addClass( 'caixa' );
```

```
$('#div').addClass( 'caixa esquerda' );
```

```
$('#div').css( 'background-color' );
```

```
$('#div').css( 'background-color', 'red' );
```

```
$('#div').css( { 'background-color': 'red', 'color': 'white' } );
```

Manipulação de Estilos CSS com jQuery

- Principais métodos para **manipular as dimensões dos elementos**:

width()

height()

innerWidth()

innerHeight()

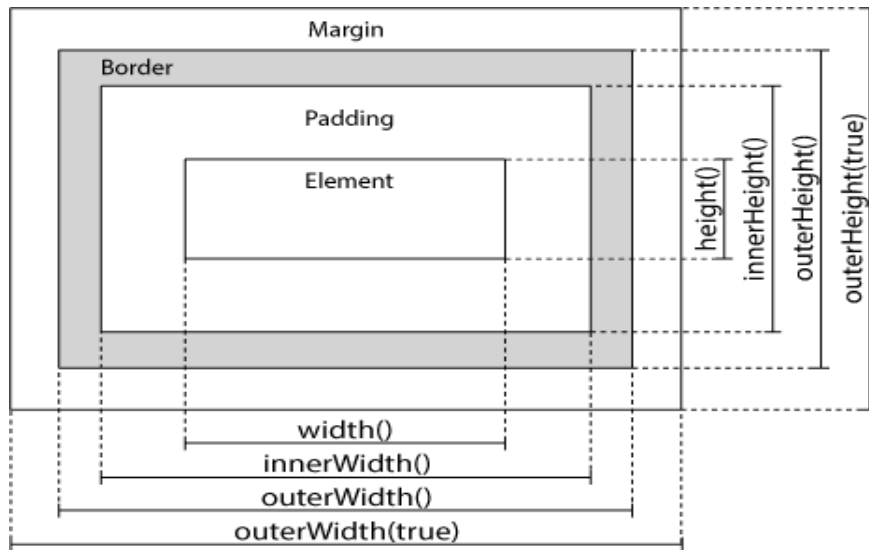
outerWidth()

outerHeight()

Exemplos:

```
$('div').width( 500 );
```

```
$('div').innerWidth( 520 );
```



Manipulação de Estilos CSS com jQuery



Exercício 7:

Altere o exercício 6 de forma a adicionar uma classe CSS “titulo” com os estilo cor verde e tamanho de fonte 24px. Utilizando jQuery aplique esse estilo ao elemento h2 da página.

Exercício 8:

Altere o exercício 7 de forma a que ao passar com o ponteiro do rato sobre cada um dos parágrafos da div a sua cor passe para branco e a cor de fundo para cor de rosa, repondo as cores logo que o ponteiro do rato deixe de estar sobre esse parágrafo.

Exercício 9:

Altere o exercício 8 de forma a que a div tenha um tamanho de 400px (utilizando jQuery).

- Navegação na árvore DOM com jQuery
 - Principais métodos para obter os elementos pai de um dado elemento:
 - **parent()** – obtém o **elemento pai** do elemento selecionado
 - **parents()** – obtém **todos** (ou os do tipo indicado no argumento no método) **os elementos pai** do elemento selecionado
 - **parentsUntil()** – obtém **todos os elementos pai entre** o elemento selecionado e o elemento indicado como argumento do método
 - Exemplos:
 - `$('p').parent();`
 - `$('p').parents('div');`
 - `$('p').parentsUntil('div');`

Navegação na árvore DOM com jQuery

- Principais métodos para **obter os elementos filho de um dado elemento**:

children() – obtém todos (ou os do tipo indicado no argumento no método) os elementos filho diretos do elemento selecionado

find() – obtém todos (ou os do tipo indicado no argumento no método) os elementos filho do elemento selecionado

Exemplos:

```
$('#div').children( );
```

```
$('#div').children('p.x');
```

```
$('#div').find('*');
```

Navegação na árvore DOM com jQuery

- Principais métodos para **obter os elementos que partilham um mesmo elemento pai:**

siblings() – obtém todos os elementos que partilham o mesmo elemento pai

next(), **nextAll()**, **nextUntil()**, **prev()**, **prevAll()**, **prevUntil()**

Exemplos:

```
$('#p').siblings( );
```

```
$('#p').next( );
```

Navegação na árvore DOM com jQuery

- Outros métodos para **obter elementos específicos (filtragem)**: **first()**
– obtém o primeiro elemento de entre os selecionados **last()** – obtém o último elemento de entre os selecionados

eq() – obtém o elemento com o índice especificado no argumento do método (os índices começam em 0)

filter() – obtém conjunto de elementos de acordo com o critério especificado no argumento do método

not() – o contrário do método filter()

Exemplos:



```
$('#p').filter( '.x' );
```

```
$('#p').first( );
```

jQuery Traversing Methods

http://www.w3schools.com/jquery/jquery_ref_traversing.asp

Efeitos com jQuery

- Métodos para **mostrar/ocultar elementos**: **hide()** – ocultar o(s) elemento(s) selecionado(s) **show()** – mostrar o(s) elemento(s) selecionado(s)

toggle() – comuta de forma alternada entre os 2 métodos anteriores

Sintaxe: **\$(selector).metodo(speed,callback);**

speed – velocidade de execução da ação (slow, fast ou milisegundos)

callback – função executada após o término do método

Exemplos:

```
$('#button').click(function(){  
    $('#p').toggle();  
});
```

Efeitos com jQuery

jQuery

- Métodos para **aplicar fade in e fade out aos elementos**: **fadeIn()** – aplicar fade in ao(s) elemento(s) selecionado(s) **fadeOut()** – aplicar fade out ao(s) elemento(s) selecionado(s)

fadeToggle() – comuta de forma alternada entre os 2 métodos anteriores

fadeTo () – permite aplicar fade até um dado nível de opacidade (entre 0 e 1)

Sintaxe: **\$(selector).metodo(speed,callback);**

speed – velocidade de execução da ação (slow, fast ou milisegundos)

callback – função executada após o término do método

Exemplos:

```
$('#button').click(function(){  
    $('#p').fadeTo("slow",0.2);  
});
```

Efeitos com jQuery

jQuery

- Métodos para **aplicar sliding aos elementos**:
slideDown() – aplicar sliding down ao(s) elemento(s) selecionado(s)
slideUp() – aplicar sliding up ao(s) elemento(s) selecionado(s)
slideToggle() – comuta de forma alternada entre os 2 métodos anteriores

Sintaxe: **\$(selector).metodo(speed,callback);**

speed – velocidade de execução da ação (slow, fast ou milisegundos)

callback – função executada após o término do método

Exemplos:

```
$(‘button’).click(function(){  
    $(‘p’).slideToggle(“slow”);  
});
```


Encadeamento de instruções jQuery



- De forma a aplicar vários métodos jQuery sobre um mesmo conjunto de elementos numa única instrução aplica-se a característica de **chaining** do jQuery, ou seja junção de vários métodos

Sintaxe: `$(selector).metodo(speed,callback);`

speed – velocidade de execução da ação (slow, fast ou milisegundos)

callback – função executada após o término do método

Exemplos:

```
$('#x').css( 'color','red').slideDown(2000).slideUp(2000);
```

```
$('#x').css( 'color','red')
```

```
.slideDown(2000)
```

```
.slideUp(2000);
```

Efeitos com jQuery

Exercício 10:

Altere o exercício 9 de forma a substituir os botões “Mostrar” e “Ocultar” por um só botão, alterando o texto do botão consoante o caso.

Exercício 11:

Altere o exercício 10 de forma a que após ser introduzido um novo email o conteúdo da caixa de input seja removido.