

jQuery

jQuery

Nota: Esta matéria **não será avaliada** em exame.

jQuery

- jQuery é um *framework Javascript*
 - Criado para facilitar a utilização do *Javascript*
 - Existem outros *frameworks JavaScript* (<https://colorlib.com/wp/javascript-frameworks/>)

Escrever menos e fazer mais

- Permite:
 - Selecção e manipulação de elementos HTML (DOM)
 - Selecção e manipulação de elementos CSS
 - *Event Handling*
 - Criação de efeitos e animações
 - ...

- A inserção de *jQuery* é feita com base na tag **<script>** e pode ser efectuada de duas formas:

- Através do download e inserção directa do ficheiro : jquery-3.2.1.js

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

```
<script>

    /* Código jQuery */

</script>
```

- Diretamente a partir de um *Content Delivery Network (CDN)*



- Potencialmente mais rápido mas tem como desvantagem a eventual indisponibilidade do ficheiro

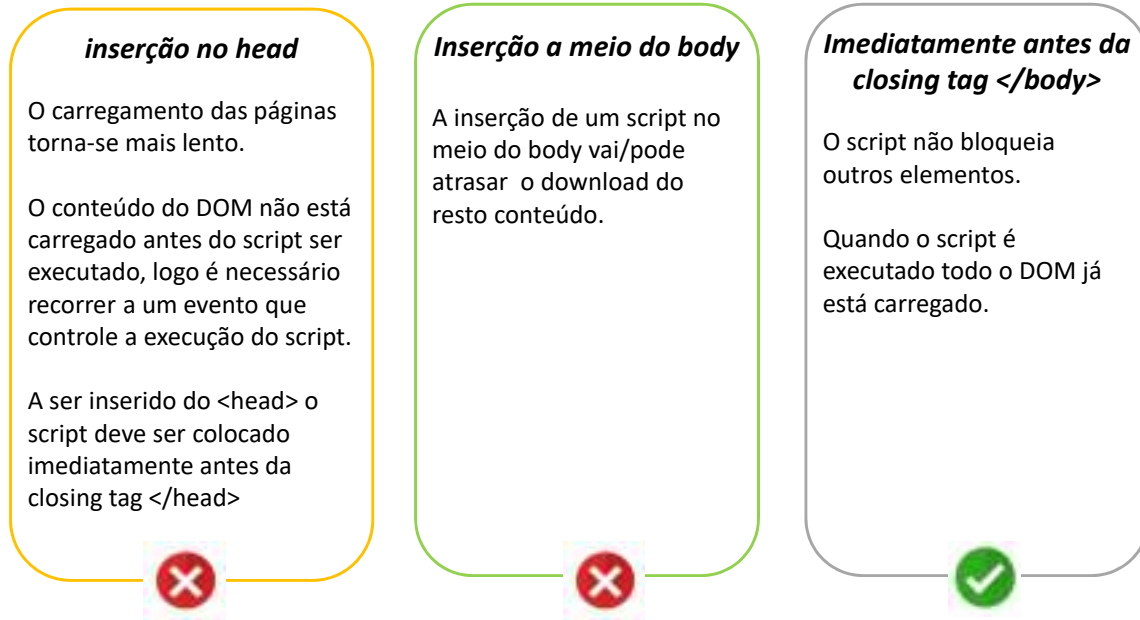
```
<script src="//ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

- Pode ser contornado verificando se o objeto *jQuery* foi criado:

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

<script>
    window.jQuery || document.write('<script src=" jquery-3.2.1.js"></script>')
</script>
```

- Inserção do script
 - a posição do script afeta a velocidade de carregamento de uma página



Sintaxe

- Sintaxe *jQuery*

`$(selector).method(parameters)`

- **\$**
 - Indica que se trata de um comando *jQuery*
- ***selector***
 - Identifica o(s) elemento(s) HTML alterados pelo método definido à direita (depois do ponto)
 - Baseia-se nos seletores CSS.
 - Mais simples uma vez que não necessita de recorrer aos métodos disponibilizados pelo DOM (*getElementById()*, *getElementsByTagName()*, ...)
- ***method (parameters)***
 - Define a operação a efetuar nos elementos.

Seletores jQuery

jQuery Selectors

- Seleção de elementos HTML (*selector*)
 - Seletores baseados em **seletores CSS**:
 - **Elemento, contexto, id, class, atributo**

HTML

jQuery

```
<body>
  <p>CSS selector</p>
  <p>CSS selector</p>
  <div id="d1">
    CSS selector
  </div>
  <script src="jquery-1.12.1.js"></script>
  <script>
    $("p").text("<p> Content Replaced!");
    $("#d1").text("<div> Text Content Replaced!");
  </script>
</body>
</html>
```

```
<p> Content Replaced!
<p> Content Replaced!
<div> Text Content Replaced!
```

jQuery Filters

- Para além dos **seletores CSS** podem ser aplicados **jQuery Filters**:

<https://api.jquery.com/category/selectors>

- **:first**

- *permite seleccionar o primeiro elemento de um grupo*

jQuery

```
<h2>Pets</h2>
<p id="p1"> first Paragraph!!! </p>
<p id = "p2">second Paragraph!!! </p>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('p:first').html('REPLACED');
</script>
```

Pets

REPLACED

second Paragraph!!!

- **:last**

- *permite seleccionar o último elemento de um grupo*

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('p:last').html('REPLACED');
</script>
```

Pets

first Paragraph!!!

REPLACED

jQuery Filters

- **:even**

- *permite seleccionar os elementos pares: index (0, 2, 4, ...)*

jQuery

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three">bird</li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:even').html('rabbit');
</script>
```

Pets

- rabbit
- cat
- rabbit
- fish

- **:odd**

- *permite seleccionar os elementos impares: index (1, 3, 5, ...)*

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:odd').html('rabbit');
</script>
```

Pets

- dog
- rabbit
- bird
- rabbit

jQuery Filters

■ :even

- permite seleccionar os elementos pares: index (0, 2, 4, ...)
- a indexação é sequencial ao longo de toda a estrutura HTML

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three">bird</li>
  <li id="four">fish</li>
  <li id="five">hamster</li>
</ul>

<ul>
  <li id="six">lizard</li>
  <li id="seven">horse</li>
  <li id="eight">turtle</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:even').html('rabbit');
</script>
```

Pets

- rabbit 0
- cat
- rabbit 2
- fish
- rabbit 4
- lizard
- rabbit 6
- turtle

■ :odd

jQuery Filters

■ :has(selector)

- permite seleccionar um elemento que contém outro elemento HTML (só aplicável a elementos HTML)

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:has(span)').html('rabbit');
</script>
```

Pets

- dog
- cat
- rabbit
- fish

■ :not(selector)

- selecciona todos os elementos menos o elemento especificado no seletor

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:not(#two)').html('rabbit');
</script>
```

Pets

- rabbit
- cat
- rabbit
- rabbit

jQuery Filters

■ :eq(index)

- Afeta o elemento que corresponde ao índice especificado como parâmetro
- Índice inicia em zero

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:eq(2)').html('rabbit');
</script>
```

Pets

- dog
- cat
- rabbit
- fish

■ :gt(index)

- afeta os elementos com um índice superior ao especificado como parâmetro
- índice inicia em zero

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:gt(3)').html('rabbit');
</script>
```

Pets

- dog
- cat
- bird
- fish

jQuery Filters

■ :lt(index)

- afeta os elementos com um índice inferior ao especificado como parâmetro
- índice inicia em zero

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:lt(2)').html('rabbit');
</script>
```

Pets

- rabbit
- rabbit
- bird
- fish

jQuery Filters

▪ :first-child

- Primeiro filho do seu pai.
- Especificado no **elemento filho**

HTML

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:first-child').html('rabbit');
</script>
```

Pets

- rabbit
- cat
- bird
- fish

▪ :last-child

- Último filho do seu pai
- Especificado no **elemento filho**

```
<script>
  $('li:last-child').html('rabbit');
</script>
```

Pets

- dog
- cat
- bird
- rabbit

jQuery Filters

▪ :first vs. :first-child

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

```
<ul>
  <li id="six">lizard</li>
  <li id="seven">horse</li>
  <li id="eight">turtle</li>
</ul>
```

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:first-child').html('rabbit');
</script>
```

```
<script>
  $('li:first').html('rabbit');
</script>
```

- rabbit
- cat
- bird
- fish

- rabbit
- horse
- turtle

- rabbit
- cat
- bird
- fish

- lizard
- horse
- turtle

jQuery Filters

■ :nth-child(index)

- Permite seleccionar o elemento que é o *nth child* do seu pai
- Especificado no **elemento filho**
- Este índice inicia-se com o **valor 1**

HTML

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:nth-child(3)').html('rabbit');
</script>
```

Pets

- dog
- cat
- rabbit
- fish

<https://api.jquery.com/category/selectors/child-filter-selectors/>

jQuery Filters

■ :only-child

- quando existe apenas 1 filho

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
<ul>
  <li id="five">horse</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li:only-child').html('rabbit');
</script>
```

Pets

- dog
- cat
- bird
- fish
- rabbit

jQuery Filters

- **:visible**
 - Permite selecionar o(s) elemento(s) visíveis
- **:hidden**
 - Permite selecionar o(s) elemento(s) não visíveis
- **:header**
 - afeta todos os elementos h1-h6
- **:focus**
 - O elemento que está a ser alvo do focus
- **:animated**
 - Elementos que estão a ser animados
- ...

<https://api.jquery.com/category/selectors/child-filter-selectors/>

jQuery Filters

■ Exemplo:

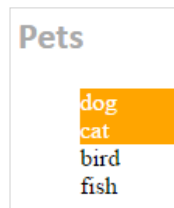
```
<style>
  ul{list-style: none;}
  .headline{color:darkgray;font:100;font-family:calibri}
  .listOptions{color:white;background-color: orange}
</style>
```

HTML

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $(':header').addClass('headline');
  $('li:lt(2)').addClass('listOptions');
</script>
```



- Seletores específicos para formulários:
 - Geralmente **não são** a forma mais eficaz de aceder a um elemento
 - Os *CSS selectors* são interpretados de forma mais rápida, pelo que a sua utilização deve ser privilegiada;
 - Estes *jQuery Filters* podem ser utilizados em seleções (subconjuntos do documento) que permitam restringir o espaço de pesquisa
 - `:input` `:text` `:password` `:radio` `:checkbox` `:file`
 - `:submit` `:reset` `:button` `:image`
 - `:selected` (itens selecionados de uma lista *drop-down*)
 - `:enabled` `:disabled`
 - `:checked` (aplicado a *radio buttons* e *checkboxes*)

- `:checked`

```
<form id="messageForm" action="#">
  <input type="checkbox" checked/>
  <br/>
  <input type="submit" value="confirm"/>
</form>
```

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $(':checked').css('width', '100px').css('height', '100px');
</script>
```

Estrutura HTML



Após o script



Elements Selection

■ Seleção de elementos HTML

- Sempre que um seletor é interpretado , é criado um **jQuery object**
 - Quando se pretende efetuar múltiplas referências ao mesmo elemento HTML, deve ser declarada uma variável para guardar esse objeto:

```
var $nomeVar = $('div');  
var $nomeVar=$('p').next();
```

- A utilização da variável torna o código muito mais eficaz quando comparado com a interpretação direta do seletor de forma repetida.
 - São aplicáveis à variável os métodos e propriedades do jQuery object que ela contém
- Por convenção os nomes das variáveis referentes a *jQuery Objects* iniciam-se com o símbolo \$

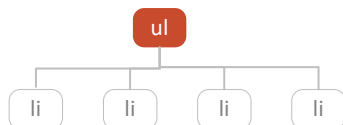
Elements Selection

■ jQuery object

single element

Se um seletor retorna apenas 1 elemento, o *jQuery object* contém uma referência apenas para esse nó.

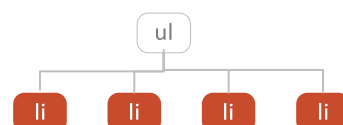
`$('ul')`



multiple elements

Se um seletor retorna vários elementos, o *jQuery object* contém uma referência para cada um dos nós.

`$('li')`



```
$('li').html('new content');  
Atualiza o conteúdo de todos os elementos <li>
```

```
var content=$('li').html();  
Devolve o conteúdo apenas do 1º elemento <li>
```

jQuery methods

jQuery method()

■ *\$(selector).method(parameters);*

■ Tipos de ação (métodos):

- Seleção de Elementos
- Alteração de Conteúdo
- Alteração de Estrutura
- Alteração de Atributos
- Efeitos
- Animação
- ...

jQuery method()

■ Chaining

- O jQuery permite o encadeamento de vários métodos
 - Se um dos métodos não funcionar, os restantes métodos que se encontram encadeados também não vão funcionar

jQuery

```
$('#ex1').hide().fadeIn(1400).css('border','2px solid #FF0');
```

ou

```
$('#ex1')  
.hide()  
.fadeIn(1400)  
.css('border','2px solid #FF0');
```

Facilita a leitura do código
Cada linha inicia com a dot notation
A cadeia termina com ;

jQuery method()

■ Looping

- O jQuery simplifica de forma significativa a aplicação de loops
 - Ao contrário do *JavaScript* (necessária uma estrutura de repetição), no *jQuery* é o próprio seletor que determina os elementos aos quais vai ser aplicado uma dada ação/efeito

HTML

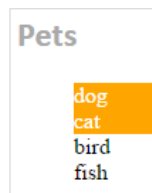
```
<h2>Pets</h2>  
<ul>  
  <li id="one">dog</li>  
  <li id="two">cat</li>  
  <li id="three"><span>bird</span></li>  
  <li id="four">fish</li>  
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>  
<script>  
  $(':header').addClass('headline');  
  $('li:lt(2)').addClass('listOptions');  
</script>
```

CSS

```
<style>  
  ul {list-style:none;}  
  .headline{color:brown;font:100;font-family:Calibri}  
  .listOptions {background-color:orange;color:white}  
</style>
```



Atribui a class *listOptions* aos primeiros dois elementos

- Propriedade `.length`

- Esta propriedade é particularmente importante (**muito útil**) pois permite obter o número de elementos num *jQuery object*:

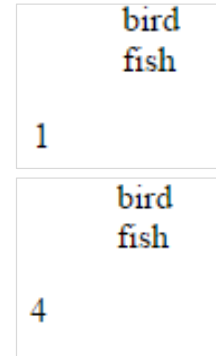
- **Contagem de elementos / Detetar objetos vazios**

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="pet">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
<div id="contagem"></div>

<script src="jquery-3.2.0.min.js"></script>
<script>
  var n=$('#ul li.pet').length;
  $('#contagem').html(n);

  $('li').addClass('pet');

  n=$('#ul li.pet').length;
  $('#contagem').html(n);
</script>
```



jQuery methods

(Seleção de Elementos)

jQuery method()

■ Traversing methods

- utilizados como complemento dos *selectors*, *permite aceder*:

Método	Descrição
next()	ao “irmão” imediatamente seguinte do elemento selecionado
prev()	ao “irmão” imediatamente anterior do elemento selecionado
nextAll()	a todos os “irmãos” seguintes do elemento selecionado
prevAll()	a todos os “irmãos” anteriores do elemento selecionado
siblings()	aos “irmãos” do elemento selecionado
children()	a todos os filhos do elemento selecionado
parents()	a todos os pais do elemento selecionado
parent()	ao pai direto do elemento selecionado

http://www.w3schools.com/jsref/dom_obj_document.asp

jQuery method()

■ jQuery Traversing Methods:

- *next()*
 - permite aceder ao “irmão” (sibling) imediatamente seguinte de um dado elemento

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li.third').next().css('background-color','lightgray');
</script>
```

Pets

dog
cat
bird
fish
lizard

jQuery method()

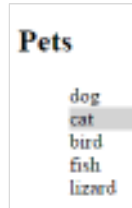
■ jQuery Traversing Methods :

■ prev()

- permite aceder ao “irmão” (sibling) imediatamente anterior de um dado elemento

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li.third').prev().css('background-color','lightgray');
</script>
```

jQuery



jQuery method()

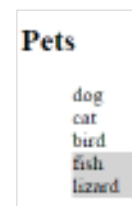
■ jQuery Traversing Methods :

■ nextAll()

- permite aceder aos irmãos seguintes do elemento selecionado

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li.third').nextAll().css('background-color','lightgray');
</script>
```

jQuery



■ prevAll()

- permite aceder aos irmãos anteriores do elemento selecionado

jQuery

```
<script>
  $('li.third').prevAll().css('background-color','lightgray');
</script>
```



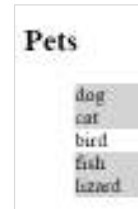
jQuery method()

jQuery Traversing Methods :

siblings()

- permite aceder aos irmãos do elemento seleccionado

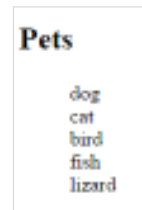
```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li.third').siblings().css('background-color', 'lightgray');
</script>
```



children()

- permite aceder a todos os filhos (elementos HTML) do elemento seleccionado

```
<script>
  $('li.third').children().css('background-color', 'lightgray');
</script>
```



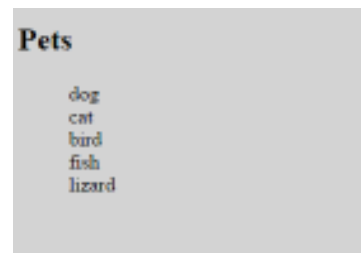
jQuery method()

jQuery Traversing Methods :

parents()

- permite aceder a todos os pais do elemento seleccionado

```
<div>
  <h2>Pets</h2>
  <ul>
    <li id="one">dog</li>
    <li id="two">cat</li>
    <li id="three" class="third">bird</li>
    <li id="four">fish</li>
    <li id="five">lizard</li>
  </ul>
</div>
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li.third').parents().css('background-color', 'lightgray');
</script>
</div>
</body>
```

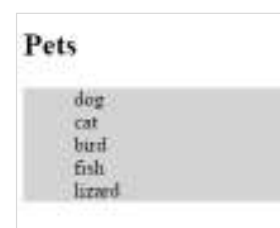


<body> <div>

parent()

- permite aceder ao pai directo do elemento seleccionado

```
<script>
  $('li.third').parent().css('background-color', 'lightgray');
</script>
```



jQuery methods

(Alteração de Conteúdo)

jQuery method()

- *Alteração de Conteúdo dos Elementos HTML*
 - Existem alguns métodos para adicionar/alterar conteúdo

Método	Descrição
<i>html()</i>	Permite obter/alterar o conteúdo HTML
<i>text()</i>	Permite obter/alterar texto
<i>val()</i>	Permite obter/alterar o valor de um elemento (forms)

jQuery method()

■ `html()`

- permite **ler e/ou substituir** o conteúdo de um elemento.
- **Sem a especificação** de parâmetros
 - permite **obter/ler** o conteúdo HTML no interior do elemento especificado no seletor.

```
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  alert($('ul').html());
</script>
```

```
<li id="one">dog</li>
<li id="two">cat</li>
<li id="three" class="third">bird</li>
<li id="four">fish</li>
<li id="five">lizard</li>
```

OK

jQuery

jQuery method()

■ `html()`

- **Com especificação** de parâmetros
 - permite **alterar/definir** o HTML no interior do elemento especificado no seletor.

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('ul').html('<li>New Pet - Horse </li>');
</script>
```

jQuery

Pets

New Pet - Horse

jQuery method()

■ text()

- operação semelhante ao *html()* mas não aceita a definição de tags HTML, apenas texto

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three" class="third">bird</li>
  <li id="four">fish</li>
  <li id="five">lizard</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('ul').text('<li>New Pet - Horse </li>');
</script>
```

jQuery

Pets

New Pet - Horse

Não possibilita a interpretação das tags HTML.

Considera todo o conteúdo como texto, neste caso aplicado ao elemento

jQuery method()

■ val()

- operação semelhante aos métodos anteriores mas destinado a alterar/obter o valor de um elemento (*form elements*)

```
<form action="#">
  <input type="text" />
  <button> New Pet!</button>
</form>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('button').on('click',function(){
    $('input').val('lizard');
  });
</script>
```

jQuery

New Pet!

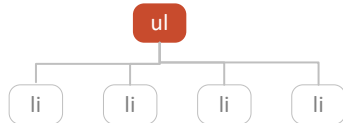
New Pet!

■ `html()`

single element

Se um seletor retorna apenas 1 elemento, o *jQuery object* contém uma referência apenas para esse nó.

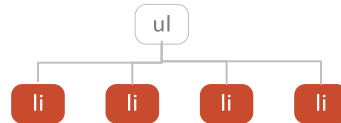
`$('#ul')`



multiple elements

Se um seletor retorna vários elementos, o *jQuery object* contém uma referência para cada um dos nós.

`$('.li')`



Funcionamento específico do método `html()` →

`$('.li').html('new content');`
Atualiza o conteúdo de todos os elementos ``

`var content=$('#li').html();`
Devolve o conteúdo apenas do 1º elemento ``

jQuery methods

(Alteração de Estrutura)

jQuery method()

■ Alteração da Estrutura

- Conjunto de métodos destinados à alteração da estrutura da DOM Tree

Método	Descrição
<i>append()</i>	adiciona um elemento como último filho do elemento selecionado <i>existingElement.append(newElement);</i>
<i>appendTo()</i>	adiciona um elemento como último filho do elemento selecionado <i>newElement.appendTo(existingElement);</i>
<i>prepend()</i>	adiciona um elemento como primeiro filho do elemento selecionado <i>existingElement.prepend(newElement);</i>
<i>prependTo()</i>	adiciona um elemento como primeiro filho do elemento selecionado <i>newElement.prependTo(existingElement);</i>
<i>before()</i>	adiciona um elemento imediatamente antes do elemento selecionado
<i>after()</i>	adiciona um elemento imediatamente depois do elemento selecionado
<i>remove()</i>	remove o elemento selecionado
<i>replaceWith()</i>	substituição de um elemento
<i>clone()</i>	faz uma cópia integral (texto, atributos, filhos) do elemento selecionado

jQuery method()

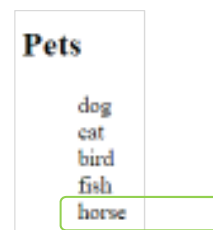
■ *append()*

- Adiciona um elemento HTML como ultimo filho do elemento selecionado
 - *existingElement.append(newElement);*

jQuery

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('ul').append('<li>horse</li>');
</script>
```



jQuery method()

■ *appendTo()*

- Adiciona um elemento HTML como ultimo filho do elemento selecionado
- *newElement.appendTo(existingElement);*
- útil para funcionar em conjunto com clone()

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('<li>horse</li>').appendTo('ul');
  $('ul').append('<li>horse</li>');
</script>
```



jQuery method()

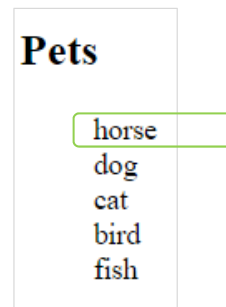
■ *prepend()*

- Adiciona um elemento HTML como primeiro filho do elemento selecionado

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('ul').prepend('<li>horse</li>');
</script>
```

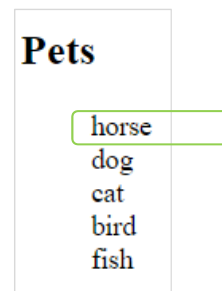
jQuery



■ *prependTo()*

- O mesmo que *prepend()* com uma sintaxe diferente
- *newElement.prependTo(existingElement);*

```
<script>
  $('<li>horse</li>').prependTo('ul');
</script>
```



jQuery method()

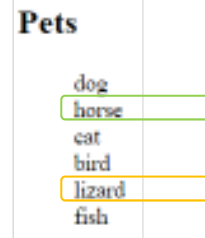
■ after(); before()

- permite o posicionamento depois/antes de uma dada referência

jQuery

```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('#two').before('<li>horse</li>');
  $('#three').after('<li>lizard</li>');
</script>
```



jQuery method()

■ remove()

- Elimina o conteúdo identificado pelo seletor
- Pode ser aplicado a vários elementos

jQuery

```
<h2>Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>

<script src="jquery-3.2.0.min.js"></script>
<script>
  $('.rmv').remove();
</script>
```

Pets

fish

jQuery method()

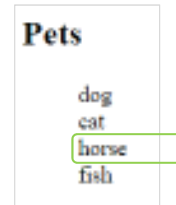
■ `replaceWith()`

- substitui o elemento identificado pelo seletor, pelo conteúdo especificado como argumento

```
<h2>Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('#three').replaceWith('<li>horse</li>');
</script>
```



jQuery method()

■ `clone()`

- faz uma cópia integral (conteúdo, atributos) do elemento selecionado.

```
<h2>Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
<div> ul : copied through clone()</div>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('#ul').clone().appendTo('div');
</script>
```

Pets

dog
cat
bird
fish

ul : copied through clone()

dog
cat
bird
fish

jQuery methods

(Atributos)

jQuery method()

■ Atributos

- Conjunto de métodos destinados à alteração dos atributos, permitem:

Método	Descrição
<i>attr()</i>	adicionar um atributo
<i>removeAttr()</i>	remover um atributo
<i>addClass()</i>	adicionar uma class
<i>removeClass()</i>	remover uma class
<i>css()</i>	definir o valor de propriedades CSS
...	...

jQuery method()

■ attr()

- Se especificado **apenas** o nome do atributo, permite a leitura do valor desse atributo

```
<style>
  ul{list-style: none;}
  .headline{color:lightgreen;font:100;font-family:calibri}
  .listOptions{color:white;background-color:orange}
</style>
```

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  var $nameClass=$( 'h2' ).attr( 'class' );
  $( 'h2' ).html( $nameClass );
</script>
```

Most Popular Pets

dog
cat
bird
fish

jQuery method()

■ attr()

- Se especificados dois argumentos (atributo; valor do atributo) é possível alterar o valor de qualquer um dos atributos especificados num dado elemento

```
<style>
  ul{list-style: none;}
  .headline{color:lightgreen;font:100;font-family:calibri}
  .listOptions{color:white;background-color:orange}
</style>
```

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $( 'h2' ).attr( 'class', 'headline' );
</script>
```

Altera o valor do atributo class, o que vai originar a aplicação da formatação definida no código css

Pets

dog
cat
bird
fish

jQuery method()

■ `addClass()`

- Adicionar o atributo `class` a um elemento
- Este método não substitui o valor do atributo `class`, adiciona aos valores já existentes.

```
<style>
  ul{list-style: none;}
  .headline{color:lightgreen;font:100;font-family:calibri}
  .listOptions{color:white;background-color:orange}
</style>
```

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('h2').addClass('headline');
</script>
```

Pets

dog
cat
bird
fish

jQuery method()

■ `removeClass()`

- Remover o valor do atributo `class`
- caso não seja especificado um valor, são removidos todos os valores do atributo `class`

```
<style>
  ul{list-style: none;}
  .headline{color:lightgreen;font:100;font-family:calibri}
  .listOptions{color:white;background-color:orange}
</style>
```

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('h2').addClass('headline');
  $('h2').removeClass('headline');
</script>
```

Pets

dog
cat
bird
fish

■ `toggleClass()`

- Alternar o atributo `class` (se estiver definido então remove/caso não esteja definido adiciona)
- Muito útil para alterar o mesmo elemento dependendo do seu estado (ex: botão selecionado)

jQuery method()

■ css()

■ Aceder ao valor de uma propriedade

- Deve ser especificado o valor da propriedade pretendida

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
<div id="fontSize"></div>
<div id="fontColor"></div>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  var fSize=$('#two').css('font-size');
  var fColor=$('#two').css('color');

  $('#fontSize').html(fSize);
  $('#fontColor').html(fColor);
</script>
```

Pets

dog
cat
bird
fish

16px
rgb(0, 0, 0)

jQuery method()

■ css()

■ Definir o valor de **uma** propriedade

- `$('#body').css('font-size','200%');`

■ Definir o valor de **múltiplas** propriedades: 3 formas distintas

```
$('#ex1').css('font-size','200%');
```

```
$('#ex1').css('border','2px solid #FF0');
```

Propriedade e valor separados por vírgula
Menos eficiente uma vez que obriga a múltiplas interpretações do seletor

```
$('#ex1').css('font-size','200%')
```

```
.css('border','2px solid #FF0');
```

chaining
Propriedade e valor separados por vírgula

```
$('#ex1').css({'font-size':'200%',
```

```
'border':'2px solid #FF0'});
```

Propriedade e valor separados por :
sendo a vírgula usada para separar propriedades

```
<script>
  $('#h2').css('color','lightgreen').css('font-size','2em');
</script>
```

jQuery

Pets

dog
cat
bird
fish

jQuery method()

■ css()

- Definir o valor de múltiplas propriedades

HTML

```
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script>
  $('li').css({
    'background-color':'lightgreen',
    'color':'white',
    'border':'1px solid lightgray',
    'width':'10%',
    'padding':'5px',
    'font-family':'verdana',
    'margin-top':'1px'
  });
</script>
```

Pets

dog

cat

bird

fish

jQuery methods

(Loops)

jQuery method()

■ *loops*

- método destinado à implementação de estruturas de repetição:

Método	Descrição
<i>each()</i>	associa a cada um dos elementos um conjunto de processamento

jQuery method()

■ *each()*

- Permite criar um *loop* para situações específicas:
 - obter informação sobre cada um dos elementos num determinado conjunto
 - realizar um conjunto de operações em cada um dos elementos
- o parâmetro do ***each()*** é uma ***anonymous function*** que define o conjunto de operações a desenvolver em cada um dos elementos selecionados
 - em conjunto com ***each()***, ***this*** refere-se ao elemento que está a ser referenciado pelo método ***each()***
 - ***\$(this)*** cria um *jQuery object* que contém o elemento referenciado por ***each()*** o que permite utilizar neste elemento todos os métodos *jQuery*

jQuery method()

■ each()

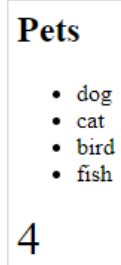
```
<h2>Pets</h2>
<ul>
  <li id="one">dog</li>
  <li id="two">cat</li>
  <li id="three">bird</li>
  <li id="four">fish</li>
</ul>
<div id="contagem"></div>

<script src="jquery-3.2.1.js"></script>
<script>
  $("li").addClass("pet");
  var n=0;

  $(".li.pet").each(function(){
    n=n+1;
  });

  $("#contagem").html(n);
</script>
```

exemplo: Contagem de elementos



each() utilizado para contagem de elementos.
Percorre todos os com class "pet" e incrementa o contador

jQuery method()

■ each()

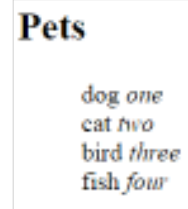
■ Exemplo: Alteração de estrutura

HTML

```
<h2 class="Most Popular Pets">Pets</h2>
<ul>
  <li id="one" class="rmv">dog</li>
  <li id="two" class="rmv">cat</li>
  <li id="three" class="rmv"><span>bird</span></li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
  $('li').each(function(){
    var itemId = $(this).attr('id');
    $(this).append('<em> ' + itemId + '</em>');
  });
</script>
```



Em cada (percorre todos os)

Aceder ao id do elemento li (atual)

Criar um como filho do . O \$(this) permite criar um jQuery object ao qual podem ser aplicados os métodos jQuery (neste caso append)

Neste contexto o \$(this) refere-se a cada um dos li

Eventos jQuery

Events

- Ação que pode ser detetada e origina a execução de um processamento específico
- *Event handling* envolve:
 - Selecção de elementos
 - Definição do evento
 - Passar a função a um evento
 - Definir o conjunto de ações que ocorrem quando o evento especificado é disparado

Events

■ **on()**

`$(selector).on('event',function(){...})`

■ Parâmetros **Obrigatórios**:

- **event**: evento ao qual se pretende responder

Categoria	Eventos
<i>user Interface</i>	<i>focus, blur, change</i>
<i>keyboard</i>	<i>input, keydown, keyup, keypress</i>
<i>Mouse</i>	<i>click, dblclick, mouseup, mousedown, mouover, mousemove, mouseout, hover</i>
<i>form</i>	<i>submit, select, change</i>
<i>document</i>	<i>ready, load, unload</i>
<i>browser</i>	<i>error, resize, scroll</i>

- **function(){...}**: código que se pretende executar após ser disparado o evento.

- Declarado através de uma *anonymous function*.

Events

■ **mouseover/mouseout**:

HTML

```
<h2 class="MostPopularPets"> Pets </h2>
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>
```

CSS

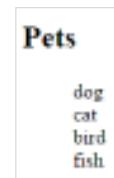
```
.change{background-color: cadetblue; color:white}
```

jQuery

```
<script src="jquery-3.2.0.min.js"></script>
<script>
$(function(){

  $('li').on('mouseover',function(){
    var itemId =$(this).attr('id');
    $(this).append('<em class="change"> '+itemId +'</em>');
  });

  $('li').on('mouseout',function(){
    $(this).children('em').remove();
  });
});
</script>
```



Event Object

- As event handling functions podem receber um event object

```
$(selector).on('event',function(e){...})
```

propriedade	descrição
<i>type</i>	tipo de evento
<i>which</i>	botão ou tecla que foi pressionada
<i>data</i>	informação extra contida num objeto (<i>literal notation</i>) que também é passada à função
<i>target</i>	elemento DOM que iniciou o evento
<i>pageX</i>	posição do cursor do rato a partir do limite esquerdo do <i>viewport</i>
<i>pageY</i>	posição do cursor do rato a partir do limite superior do <i>viewport</i>
<i>timestamp</i>	Número de milissegundos decorridos até o evento ser disparado (referência 1/1/1970 – Unix time)
método	descrição
<i>preventDefault()</i>	Evita o comportamento por <i>default</i> (ex: submissão de um form)
<i>stopPropagation()</i>	Evita que o evento seja propagado para os elementos hierarquicamente superiores (<i>bubbling effect</i>)

Event Object

- Parâmetros Adicionais nos Event Handlers

```
$(selector).on('event',[selector],[data],function(e){...})
```

* **

* : permite identificar um subconjunto da seleção original

** : dados extra que são passados à função que contém o processamento que resulta do disparo do evento (definidos como **propriedades de data**)

jQuery

```
$(function () {  
    var listItem, itemStatus, eventType;  
  
    $('ul').on('click',  
        ':not(#four)', *  
        {status: 'important'}, **  
        function (e) {  
            listItem = 'Item: ' + e.target.textContent + '<br/>';  
            itemStatus = 'Status: ' + e.data.status + '<br/>';  
            eventType = 'Event: ' + e.type + '<br/>';  
            $('#msg').html(listItem + itemStatus + eventType);  
        });  
});
```

```
<ul>  
  <li id="one" class="cl1">dog</li>  
  <li id="two" class="cl1">cat</li>  
  <li id="three" class="cl1">bird</li>  
  <li id="four">fish</li>  
</ul>  
<div id="msg"></div>
```

Pets

dog
cat
bird
fish

Item: cat
Status: important
Event: click

- `$(document).ready(function){...}`

- Garante que o código *jQuery* só é executado **após o download** completo (evento *ready*) do documento HTML

- Esta função **pode ser incluída** nos script *jQuery*, de forma a evita os eventuais problemas relacionados com a *DOM Tree* originados pela localização do script

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

→ Todo o código *jQuery* deve estar contido nesta função

- Notação condensada:

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

jQuery methods

Efeitos
Animação

jQuery method()

■ Efeitos

Método	Descrição
<code>show()</code>	Mostra um elemento
<code>hide()</code>	Esconde um elemento
<code>toggle()</code>	Alterna a visibilidade de um elemento
<code>fadeIn()</code>	O elemento surge de forma progressiva. Tempo definido em ms
<code>fadeOut()</code>	O elemento desaparece de forma progressiva. Tempo definido em ms
<code>fadeToggle()</code>	Alterna os efeitos anteriores [<code>fadeIn()</code> ; <code>fadeOut()</code>]
<code>fadeTo()</code>	Permite definir o grau de transparência
<code>slideDown()</code>	Efeito deslizante para baixo
<code>slideUp()</code>	Efeito deslizante para cima
<code>slideToggle()</code>	Alternar os dois efeitos (efeito deslizante para baixo e para cima)

jQuery method()

■ `fadeIn/fadeOut` (exemplo)

HTML

```
<h2 class="MostPopularPets"> Pets </h2>
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script>
  $(function () {
    var $li = $('li');

    $li.hide().each(function (index) {
      $(this).delay(1000*index).fadeIn(1000);});

    $li.on('click', function () {
      $(this).fadeOut(2000);});
  });
</script>
```

Opções surgem de forma sequencial e progressiva com atraso proporcional à posição que ocupam na lista

Opções são eliminadas de forma progressiva (2 s) e após um click do rato.



jQuery method()

■ Animação

Método	Descrição
<code>animate()</code>	Permite definir uma animação de um elemento
<code>delay()</code>	definir um atraso no início da animação
<code>stop()</code>	Interromper a animação

- `animate()` pode animar as propriedades CSS desde que estas sejam expressas por um valor (ex: `height`; `width` and `font-size`).

- As variáveis cujo valor é expresso através de uma string (ex: `font-family`; `text-transform`) não podem ser animadas

```
.animate({styles to change}, speed, easing, function when complete);
```

jQuery method()

■ Animação

```
.animate({styles}, speed, easing, complete);
```

■ styles

- Exemplos dos equivalentes jQuery das propriedades a animar

bottom	left	right	backgroundPositionX	backgroundPositionY	height	width	
maxHeight	minHeight	maxWidth	minWidth	margin	marginBottom	marginLeft	marginRight
marginTop	padding	paddingBottom	paddingLeft	paddingRight	paddingTop	fontSize	
letterSpacing	wordSpacing	lineHeight	textIndent	borderRadius	borderWidth	borderBottomWidth	
borderLeftWidth	borderRightWidth	borderTopWidth	borderSpacing				

J. Duckett, JavaScript and jQuery, 2014

speed: duração da animação em ms

easing pode assumir dois valores: *linear* (velocidade da animação é constante); *swing* (acelera no meio da animação e começa/termina lento)

complete: usado para chamar uma função após a finalização da animação (callback function)

jQuery method()

■ animate()

HTML

```
<h2 class="MostPopularPets"> Pets </h2>
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>
```

jQuery

```
<script>
  $('li').on('click',function(){
    $(this).animate({opacity:'0',left:'+=80'},1000,
      function(){
        $(this).remove();
      });
  });
</script>
```

Animated properties can also be relative. If a value is supplied with a leading += or -= sequence of characters, then the target value is computed by adding or subtracting the given number from the current value of the property.



Opacity é colocada a 0, e o padding-left é aumentado (operador jQuery '+=' em 80px).

O tempo da animação é 1 seg.

É iniciada através de um click do rato em cada opção da animação.

No final da animação, a opção é eliminada.

<http://api.jquery.com/animate/>

jQuery method()

■ animate()

HTML

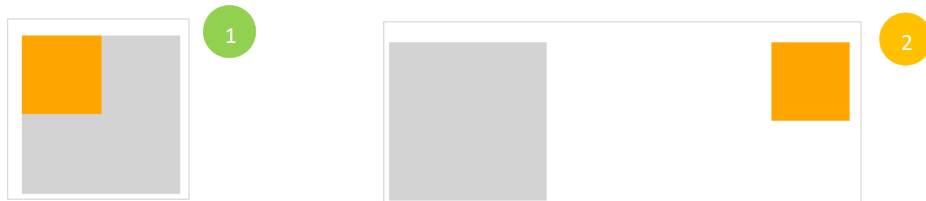
```
<div id="large"></div>
<div id="small"></div>
```

jQuery

```
<script>
  $('#large').on('click',function(){
    $('#small').animate({left:'+=250'},2000);
  });
</script>
```

```
#large{ width:100px;
        height:100px;
        background-color: lightgray;
        position: absolute
      }

#small{ width:50px;
        height:50px;
        background-color: orange;
        position: absolute}
```



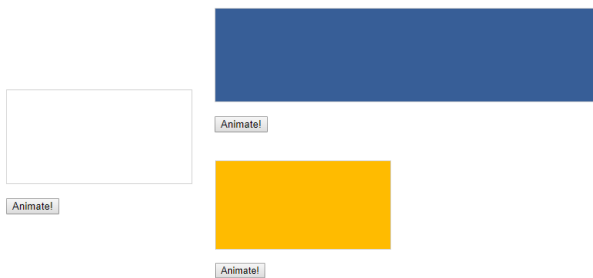
jQuery method()

■ animate()

- O *jQuery* não suporta animação de cores
- Sempre que se pretenda fazer animações que envolvam variações de cor, tem que se recorrer ao

jQuery UI

- jquery-ui.js é implementada sobre o jquery.js <http://jqueryui.com>



```
div{height:120px;
width:240px;
background-color:white;
border: 1px solid lightgray;}
```

```
<div id="d1"></div><br>
<button id="button">Animate!</button>

<script src="jquery-3.2.1.js"></script>
<script src="jquery-ui.js"></script>

<script>
  var state = true;
  $("#button").on( "click", function() {

    if ( state ) {
      $("#div").animate({
        backgroundColor: "#375E97",
        width: "500px"},
        1000 );
    }
    else {
      $("#div").animate({
        backgroundColor: "#FF8B00",
        width: "240px"}, 1000 );
    };
    state = !state;
  });
</script>
```

jQuery method()

- O método **on()** é a forma mais atual e flexível de efetuar o *event handling*
- Permite múltiplos *event handlers* associados ao mesmo element, para tal é definido um objeto (definição literal)

```
<div id="d1"></div><br>
<button id="button">Animate!</button>

<script src="jquery-3.2.1.js"></script>
<script src="jquery-ui.js"></script>

<script>
  $( "div" ).on({
    click: function() {
      $(this).addClass( "clicked" );
    }, mouseover: function() {
      $( this ).addClass( "inside" );
    }, mouseout: function() {
      $( this ).removeClass( "inside" );
    }
  });
</script>
```

```
<style>
  div{height:120px;
width:240px;
background-color:white;
border: 1px solid lightgray;}

  .inside{background-color:lightblue}
  .clicked{background-color:lightcoral}
</style>
```



mouseover



click

jQuery methods

Formulários

jQuery method()

■ Formulários

Método	Descrição
<i>val()</i>	Obter/estabelecer o valor de um elemento
<i>\$.isNumeric()</i>	Verifica se o valor é numérico (retorna um boolean). Método global, não é utilizado numa seleção jQuery.

Evento	Evento aplicado a formulários, ocorre quando:
<i>blur</i>	um elemento perde o focus
<i>change</i>	o valor do elemento é alterado
<i>focus</i>	um elemento ganha o focus
<i>select</i>	a opção de um elemento <select> é alterada
<i>submit</i>	um <i>form</i> é submetido

jQuery method()

■ Formulários (exemplo)

HTML

```
<h2 class="MostPopularPets"> Pets </h2>
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>

<div id="newItemButton"><button id="showForm">new item</button></div><br />
<form id="newItemForm">
  <input type="text" id="itemDescription" placeholder="Add description..." />
  <input type="submit" id="addButton" value="add" />
</form>
```



jQuery method()

jQuery

```
$(function () {
  var $newItemButton = $('#newItemButton');
  var $newItemForm = $('#newItemForm');
  var $textInput = $('#itemDescription');

  $newItemButton.show();
  $newItemForm.hide();
});
```

Declaração de variáveis com a seleção dos elementos (botão/form)

Mostra o botão; esconde o form



jQuery method()

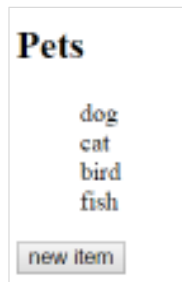
jQuery

```
$(function () {  
    var $newItemButton = $('#newItemButton');  
    var $newItemForm = $('#newItemForm');  
    var $textInput = $('#itemDescription');  
  
    $newItemButton.show();  
    $newItemForm.hide();  
  
    $('#showForm').on('click', function () {  
        $newItemButton.hide();  
        $newItemForm.show();  
    });  
});
```

Após um *click*:
Esconde o botão
Mostra o *form* para permitir adicionar
novo elemento

3

2



3



jQuery method()

jQuery

```
$(function () {  
    var $newItemButton = $('#newItemButton');  
    var $newItemForm = $('#newItemForm');  
    var $textInput = $('#itemDescription');  
  
    $newItemButton.show();  
    $newItemForm.hide();  
  
    $('#showForm').on('click', function () {  
        $newItemButton.hide();  
        $newItemForm.show();  
    });  
  
    $newItemForm.on('submit', function (e) {  
        e.preventDefault();  
        var newText = $textInput.val();  
        $('li:last').after('<li>' + newText + '</li>');  
    });  
});
```

Impede a submissão do *form*
Obtém o valor do campo de texto
Adiciona na ultima posição da lista

5

4



5



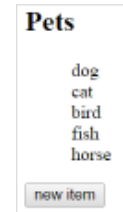
```
$(function () {  
    var $newItemButton = $('#newItemButton');  
    var $newItemForm = $('#newItemForm');  
    var $textInput = $('#itemDescription');  
  
    $newItemButton.show();  
    $newItemForm.hide();  
  
    $('#showForm').on('click', function () {  
        $newItemButton.hide();  
        $newItemForm.show();  
    });  
  
    $newItemForm.on('submit', function (e) {  
        e.preventDefault();  
        var newText = $textInput.val();  
        $('li:last').after('<li>' + newText + '</li>');  
        $newItemForm.hide();  
        $newItemButton.show();  
        $textInput.val() = '';  
    });  
});
```

jQuery

5



6



Esconde o form

Mostra o botão para adicionar nova opção

Elimina o conteúdo do campo de texto

6

jQuery methods

Cut & Copy

jQuery method()

■ Cut & Copy & Paste

Método	Descrição
<code>remove()</code>	Elimina um elemento e todos os seus descendentes da DOM Tree
<code>empty()</code>	Elimina os descendentes do nó que corresponde à seleção
<code>unwrap()</code>	Elimina os pais (diretos) dos elementos selecionados mantendo estes elementos
<code>detach()</code>	O mesmo que <code>remove()</code> mas mantém uma cópia do elemento em memória (<i>cut</i>)
<code>clone()</code>	cria uma cópia do elemento (incluindo descendentes e text nodes) (<i>copy</i>)

jQuery method()

■ Cut & Copy & Paste (exemplo)

HTML

```
<h2 class="MostPopularPets"> Pets </h2>
<ul>
  <li id="one" class="cl1">dog</li>
  <li id="two" class="cl1">cat</li>
  <li id="three" class="cl1">bird</li>
  <li id="four">fish</li>
</ul>
<p>Approximately 7.6 million companion animals enter animal shelters nationwide every year.</p>
```

```
<script>
  $(function () {
    var $p = $('p');
    var $clonedP = $p.clone();

    $p.remove();
    $('h2').after($clonedP);

    var $moveItem = $('#one').detach();
    $('ul').append($moveItem);
  });
</script>
```

Copia o parágrafo

Elimina o parágrafo
insere a cópia depois do h2

Cut e copy para a ultima
posição da lista

Pets

Approximately 7.6 million comp

cat
bird
fish
dog

jQuery

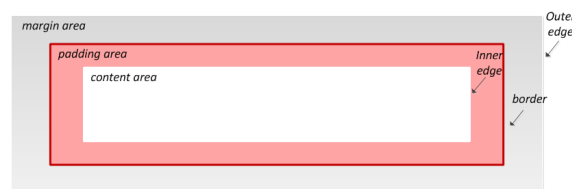
jQuery methods

Box Model

jQuery method()

■ Box model

- conjunto de métodos destinados a alterar o *box model*



Método	Descrição
<i>height()</i>	Obter/Definir altura da área de conteúdo (<i>content area</i>)
<i>width()</i>	Obter/Definir a largura da área de conteúdo (<i>content area</i>)
<i>innerHeight()</i>	Obter/Definir a altura da <i>content area</i> + <i>padding área</i>
<i>innerWidth()</i>	Obter/Definir a largura da <i>content area</i> + <i>padding área</i>
<i>outerHeight()</i>	Obter/Definir a altura da <i>content area</i> + <i>padding área</i> + <i>border</i>
<i>outerWidth()</i>	Obter/Definir a largura da <i>content area</i> + <i>padding área</i> + <i>border</i>
<i>outerHeight(true)</i>	Obter/Definir a altura <i>content area</i> + <i>padding área</i> + <i>border</i> + <i>margin</i>
<i>outerWidth(true)</i>	Obter/Definir a largura <i>content area</i> + <i>padding área</i> + <i>border</i> + <i>margin</i>

▪ box model (exemplo)

CSS

```
div{width:500px;
border: 1px solid lightgray;
padding:10px;}

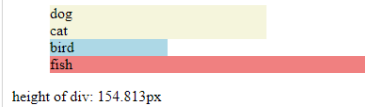
li{background-color:beige;}

ul{list-style: none}
```

HTML

```
<div id="global">
  <h2 class="MostPopularPets">Pets </h2>
  <ul>
    <li id="one" class="cl1">dog</li>
    <li id="two" class="cl1">cat</li>
    <li id="three" class="cl1">bird</li>
    <li id="four" class="cl1">fish</li>
  </ul>
</div>
```

Pets



jQuery

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

```
<script>
```

```
var listHeight=$('#global').height();
$('#div').append('height of div: ' + listHeight + 'px');
```

Cria um parágrafo com a indicação da altura da content área do <div>

```
$('#li').width('50%');
$('#three').width(125).css('background-color','lightblue');
$('#four').width('75%').css('background-color','lightcoral');
```

Altera a largura dos vários elementos li

```
</script>
```

% e em são passadas a width() como strings

jQuery methods

window & page

jQuery method()

■ window & page

Método	Descrição
<code>scrollLeft()</code>	Obter/definir a posição horizontal na scroll bar para o primeiro elemento da seleção
<code>scrollTop()</code>	Obter/definir a posição vertical na scroll bar para o primeiro elemento da seleção
<code>offset()</code>	Obter/definir as coordenadas de um elemento relativamente ao canto superior esquerdo do <i>document object</i>

jQuery method()

Exemplo: Posição da barra de scroll

```
<p class="desc">Pet or companion animal is an animal kept primarily for a person's company, protection, or entertainment rather than ...</p>
<p class="sTop"></p>

<script src="jquery-3.2.0.min.js"></script>
<script>
    $(function(){
        $('p').on('click',function(){
            var shiftTop=$('p').scrollTop();
            $('.sTop').html(shiftTop);
        });
    });
</script>
```

Pet or companion animal is an animal kept primarily for a person's company, protection, or entertainment rather than as a working animal, sport animal, livestock, or laboratory animal. Popular pets are often noted for their attractive appearances and their loyal or playful personalities. Pets provide their owners (or guardians[1]) physical and emotional benefits. Walking a dog can supply both the human and pet with exercise, fresh air, and social

0

with exercise, fresh air, and social interaction. Pets can give companionship to elderly adults who do not have adequate social interaction with other people, as well as other people that are living alone. There is a medically approved class of therapy animals, mostly dogs or cats, that are brought to visit confined humans. Pet therapy utilizes trained animals and handlers to achieve specific physical, social, cognitive, and emotional goals with

175

jQuery methods

Filtros

jQuery method()

- **Seleção / Filtros**
 - Alternativas à especificação direta dos seletores.

Método	Descrição
<code>add()</code>	Adiciona a uma seleção os elementos especificados como parâmetros

```
<div id="global">
  <h2 class="Most Popular Pets">Pets</h2>
  <ul>
    <li id="one" class="pet">dog</li>
    <li id="two">cat</li>
    <li id="three"><span>bird</span></li>
    <li id="four">fish</li>
  </ul>
</div>

<script src="jquery-3.2.1.js"></script>
<script>
  $('li').add('h2').css('background-color', 'orange').css('color', 'white');
</script>
```

Pets

dog
cat
bird
fish

<code>filter()</code>	Retorna os elementos que verificam um dado critério
-----------------------	---

```
$("p").filter(".intro").css("background-color", "yellow");
```

Aplica a formatação aos parágrafos cuja class é 'intro'

jQuery method()

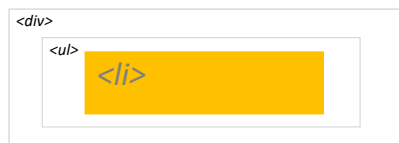
find()

Retorna os filhos dos elementos da seleção que verificam um critério

```
<div id="global">
  <h2 class="Most Popular Pets">Pets</h2>
  <ul>
    <li id="one" class="pet">dog</li>
    <li id="two">cat</li>
    <li id="three"><span>bird</span></li>
    <li id="four">fish</li>
  </ul>
</div>

<ul>
  <li>lizard</li>
</ul>

<script src="jquery-3.2.1.js"></script>
<script>
  $('div').find('li').css('background-color','orange').css('color','white');
</script>
```



Pets

dog
cat
bird
fish

• lizard

jQuery method()

is()

Retorna o resultado se a seleção verifica uma determinada condição (resultado é um boolean)

```
<body>
  <p class="desc">Pet ...
```

```
$('#p').on('click',function(){
  if($('#p').parent().is('body'))
    {alert('body is parent of p');}
});
```

body is parent of p