

Modificación do contido dos elementos

Imos ver nesta sección como podemos editar contido HTML ou de texto no documento dun xeito moi sinxelo con jQuery.

Sumario

- 1 Modificar contido HTML ou de texto
- 2 Engadir, mover e copiar elementos
- 3 **Sobre o comando `append()`** Esta instrucción engade todos os enlaces que empreguen a clase `engadeMe` ó final do párrafo que empregue a clase `engadirAqui`. A disposición dos elementos orixinais depende do número de elementos que teñamos como destino. Si soamente hai un destino (un só párrafo `engadirAqui`, o elemento orixinal é movido da súa posición orixinal ó novo pai. No caso de que haxa máis dun elemento de destino o elemento orixinal queda na súa posición orixinal e copias deste elemento son engadidas a cada párrafo de destino.
- 4 **Sobre `append`, `prepend`, `insertAfter`, `insertBefore`** Polo tanto para evitar posibles erros entre navegadores recomendaría empregar os métodos `append`, `prepend` e variantes con obxectos que teñan marca de peche `</marca>`. Os métodos de `insertAfter` e `insertBefore` poderán empregarse con calqueira obxecto de destino.
- 5 Envolvendo elementos
- 6 Eliminar elementos
- 7 Clonar elementos
- 8 Valores dos elementos nun formulario

Modificar contido HTML ou de texto

Un primeiro comando que temos para recuperar o contido HTML dun elemento é mediante o comando **`html()`**. Si chamamos a este método sin parámetros recuperamos o contido HTML do elemento (idéntico á propiedade `innerHTML` en JavaScript), e se lle pasamos como parámetro código HTML estamos modificando ese contido.

Tamén podemos obter o texto plano dese obxecto empregando o comando **`text()`**. Vexamos o seguinte exemplo:

```
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
<ul id="unhaLista">
  <li>Un</li>
  <li>Dous</li>
  <li>Tres</li>
  <li>Catro</li>
</ul>

<script language="javascript">
var meuHtml = $('#unhaLista').html();
var meuTexto = $('#unhaLista').text();

alert (meuHtml);
alert (meuTexto);
</script>

meuHtml conterá: <li>Un</li><li>Dous</li><li>Tres</li><li>Catro</li>
meuTexto conterá: Un Dous Tres Catro
```

Tamén se pode usar o comando **`text()`** para modifica-lo contido dese obxecto. Simplemente hay que pasarlle texto como parámetro.

```
$("#li:eq(2)").text("Elemento actualizado");
```

Engadir, mover e copiar elementos

Para engadir contido ó interior dun obxecto na zoa final, empregaremos o comando **`append()`**.

Este comando acepta unha cadea con contido HTML, unha referencia a un elemento actual ou creado recentemente, ou ben un grupo de elementos jQuery.

Por exemplo:

```
$('#p').append('<b>algun texto mais</b>');
```

Esta instrucción engade o contido HTML pasado como parámetro ó final de cada párrafo pero sempre dentro do párrafo. Por exemplo o comando anterior tería como resultado nun posible párrafo:

```
<p>
Texto de probas...etc..etc...<b>algún texto mais</b>
</p>
```

Un uso máis complexo deste comando pode ser:

```
$( "p.engadirAqui" ).append( $ ("a.engadeMe" ) )
```



Sobre o comando append()

Esta instrucción engade todos os enlaces que empreguen a clase engadeMe ó final do párrafo que empregue a clase engadirAqui. A disposición dos elementos orixinais depende do número de elementos que teñamos como destino. Si soamente hai un destino (un só párrafo engadirAqui, o elemento orixinal é movido da súa posición orixinal ó novo pai. No caso de que haxa máis dun elemento de destino o elemento orixinal queda na súa posición orixinal e copias deste elemento son engadidas a cada párrafo de destino.

Código do exemplo completo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Exemplo de mover e copiar</title>
  <style type="text/css">

</style>
</head>
<body>
<ul id="unhaLista">
  <li>Un</li>
  <li>Dous</li>
  <li>Tres</li>
  <li>Catro</li>
</ul>

<a href="http://www.lavozdeg Galicia.es/24horas/index.htm" class="engadeMe">24 Horas</a>
<a href="http://www.lavozdeg Galicia.es/lomas/index.htm" class="engadeMe">Lo más visto</a>
<a href="http://www.lavozdeg Galicia.es/fotos/index.htm" class="engadeMe">Fotos</a>
<a href="http://www.lavozdeg Galicia.es/videos/index.htm" class="engadeMe">Vídeos</a>
<p class="engadirAqui">Párrafo de probas 1: </p>
<p>Outr párrafo</p>
<p class="engadirAqui">Párrafo de probas 2: </p>

<script src="http://code.jquery.com/jquery-2.1.3.min.js"></script>
<script>
  $(document).ready(function()
  {
    var meuHtml = $('#unhaLista').html();
    var meuTexto = $('#unhaLista').text();

    alert(meuHtml);
    alert(meuTexto);
    $( "p.engadirAqui" ).append( $ ("a.engadeMe" ) );
  });
</script>
</body>
</html>
```

O comando **appendTo(destino)** ten o mesmo funcionamento que **append() movendo** (se o destino é único) ou **copiando** (se hai múltiples destinos) o

elemento ou elementos seleccionados á zoa interior final do destino indicado como parámetro.

Exemplos de uso:

```
$('#flor').appendTo('#destinos p')
$('#coche').appendTo('#destinos p:first')
```

No primeiro caso fará copias do elemento con ID flor e os engadirá nos párrafos contidos no obxecto con ID destinos. No segundo caso moverá o coche ó primeiro párrafo contido no obxecto con ID destinos.

Exemplo completo:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Mover e Copiar con jQuery</title>

<style>
  fieldset {
    background-color:#FFCC66;
    border-color:#00457B;
    margin-bottom:12px;
  }

  legend {
    background-color:#6699CC;
    border:2px ridge #00457B;
    color:white;
    font-weight:bold;
    padding:2px 16px;
  }
</style>
</head>
<body>
<h2>Probando a mover e copiar<br>
O debuxo de Formación soamente se copia ou move ó primeiro párrafo. </h2>
<fieldset id="orixe">
  <legend>Elementos Orixinais</legend>
  
  
</fieldset>
<fieldset id="destinos">
  <legend>Elementos de Destino</legend>
  <p></p>
  <p></p>
  <p></p>
</fieldset>
<div>
  <form action="" onSubmit="return false;">
    <div>
      <input type="radio" name="command" value="appendTo" checked="checked"/>
      appendTo
      <input type="radio" name="command" value="prependTo"/>
      prependTo
      <input type="radio" name="command" value="insertAfter"/>
      insertAfter
      <input type="radio" name="command" value="insertBefore"/>
      insertBefore
    </div>
    <button type="button" id="testButton" onclick="execute()">Probar</button>
  </form>
</div>

<script src="http://code.jquery.com/jquery-2.1.3.min.js"></script>
<script>
(function) execute() {
  var command = $('input[name=command]:checked').val();
  $('#profesor')[command]('#destinos p')
```

```
    $('#formacion')[command]('#destinos p:first')
  }
</script>

</body>
</html>
```


prepend() e **prependTo()** funcionan de xeito semellante a **append()** ou **appendTo()**, pero a inserción do contido faise ó principio da zoa interior do elemento de destino e non ó final.

Imaxen do exemplo anterior:

Probando a mover e copiar

O debuxo de Formación soamente se copia ou move ó primeiro párrafo.

Elementos Orixinais



Elementos de Destino



appendTo prependTo insertAfter insertBefore

Probar

Outra opción consiste en empregar o método **insertAfter()** ou **insertBefore()**:

Con **insertAfter()**, o contido insértase despois do indicado polo párametro, é dicir non estará contido polo obxecto destino (método **append**).

Con **insertBefore()**, o contido insértase antes do indicado polo párametro, é dicir non estará contido polo obxecto destino (método **prepend**).

Exemplo:

```
$('#<p>Hola mundo!</p>').insertAfter('p img');
```

Esta liña creará un párrafo co texto **Hola mundo !** e este párrafo será insertado despois de cada imaxe que esté contida nun párrafo.

O mesmo exemplo con **append** e **insertAfter**:

```
$("#p").append("<b>saúdos a todos</b>");  
  
// Daría como resultado nun párrafo isto:  
  
<p>texto párrafo<b>saúdos a todos</b></p>  
  
// Pero si en cambio facemos un insertAfter:  
$("#<b>saúdos a todos</b>").insertAfter("p");  
// o resultado será:  
  
<p>texto párrafo</p><b>saúdos a todos</b>
```



Sobre **append**, **prepend**, **insertAfter**, **insertBefore**

Polo tanto para evitar posibles erros entre navegadores recomendaría empregar os métodos **append**, **prepend** e variantes con obxectos que teñan marca de peche `</marca>`.

Os métodos de **insertAfter** e **insertBefore** poderán empregarse con calqueira obxecto de destino.

[[Máis información sobre insert\(\), insertAfter\(\) e outros métodos de manipulación en jQuery](#)]

Envolvendo elementos

Si por exemplo queremos meter un elemento dentro doutro, podemos empregar o comando **wrap(envoltorio)**.

Por exemplo:

```
Texto orixe en HTML:  
<a class="azuis" href="www.terra.es">texto de probas.</a>  
  
$("#a.azuis").wrap("<div class='probas'></div>");
```

Este comando colle todos os enlaces que estén empregando a clase **azuis** e os mete dentro do envoltorio **div class=...** dando como resultado o seguinte código HTML:

```
<div class="probas">  
  <a class="azuis" href="www.terra.es">texto de probas.</a>  
</div>
```

Eliminar elementos

Si queremos vaciar ou eliminar un conxunto de elementos, podemos facelo empregando o comando **remove()**.

Exemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>jQuery</title>
  <style>p { background:yellow; margin:6px 0; }</style>
</head>
<body>
  <p>Hola</p>
  Como te
  <p>atopas hoxe?</p>
  <button>Chamar a remove() nos parrafos</button>

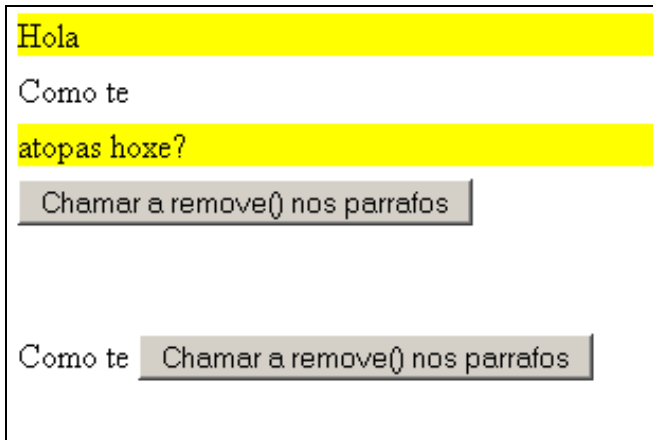
  <script src="http://code.jquery.com/jquery-2.1.3.min.js"></script>

  <script>
    $(document).ready(function(){

      $("button").click(function () {
        $("p").remove();
      });

    });
  </script>
</body>
</html>
```

Imaxe do resultado:



Si en cambio o que queremos é limpar o contido que hai dentro dun elemento podemos facelo co método **empty()**.

Exemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>jQuery</title>
  <style>p { background:yellow; margin:6px 0; }</style>
</head>
<body>
  <div class="aReemplazar">HolaComo te
    <p>atopas hoxe?</p>
  </div>
  <button>Chamar a empty()</button>

  <script src="http://code.jquery.com/jquery-2.1.3.min.js"></script>

  <script>
    $(document).ready(function(){

      $("button").click(function () {
```

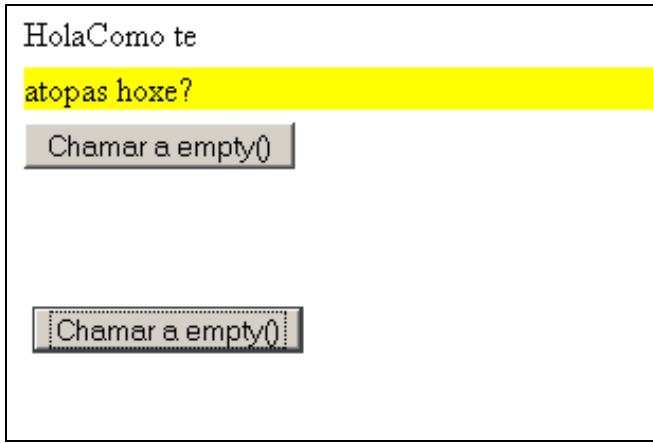
```

        $("#div.aReemplazar").empty();
    });

    });
</script>
</body>
</html>

```

Imaxe do exemplo anterior:



Un código moi empregado consiste en chamar a **remove()** despois de chamar a **after()**. Por exemplo no seguinte código:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>jQuery</title>
  <style>p { background:yellow; margin:6px 0; }</style>
</head>
<body>
  <div class="aReemplazar">HolaComo te
    <p>atopas hoxe?</p>
  </div>
  <button>Chamar a remove() despois de after()</button>

  <script src="http://code.jquery.com/jquery-2.1.3.min.js"></script>
  <script>
    $(document).ready(function() {

      $("#button").click(function () {
        $("#div.aReemplazar").after("<p>Estamos reemplazando o div</p>").remove();
      });

    });
  </script>
</body>
</html>

```

Debido a que **after()** (inserta o código despois do obxecto mencionado) devolve o envoltorio orixinal que contén o div, podemos removelo, resultando nun reemplazo do elemento orixinal.

Isto tamén se pode facer có método **replaceWith**. O exemplo anterior faríase con esta instrución tamén:

```

$("#div.aReemplazar").replaceWith("<p>Estamos reemplazando o div</p>");

```

Clonar elementos

Unha forma sinxela de facer copias de obxectos é empregando o método **clone()**.

Empregar **clone()** non ten sentido a non ser que fagamos algún tipo de operación cas copias.

Exemplos:

```
$('.img').clone().appendTo('fieldset.foto');  
  
$('.ul').clone().insertBefore('#aquí');  
  
$('.ul').clone().insertBefore('#aquí').end().hide();
```

A primeira liña: fai unha copia de tódalas imaxes e as engade a tódolos obxectos fieldset que empregan a clase foto.

A segunda liña: fai unha copia de tódalas listas desordenadas (incluídos os seus fillos) e esa copia é insertada antes de tódolos obxectos con id "aquí".

A terceira liña: fai o mesmo ca liña anterior, pero despois de insertar as copias o comando **end()** emprégase para seleccionar o conxunto de elementos orixinais e a continuación oculta-los con **hide()**.

Valores dos elementos nun formulario

Para acceder ós elementos nun formulario dispomos do **Form Plugin** desenrolado polos membros do jQuery team, que nos permite realizar un montón de tarefas con calquera tipo de elementos nun formulario.

A forma sinxela de acceder ós valores dos elementos do formulario é empregando o método **val()**.

Recordar que para xestionar ós atributos dun elemento faise con **attr("atributo")** ou **attr("atributo","valor")**.

Este método si o chamamos sin parámetros devolve o valor de ese obxecto e se lle pasamos parámetros podemos modifica-lo value dese elemento.

Exemplo:

```
var s = $('input#nome').val();
```

Devolve o valor dun campo tipo input con id "nome".

Se desexa máis información visite o seguinte enlace:

[Máis información sobre val\(\) en jQuery](#)

--Veiga (discusión) 14:07 26 ene 2015 (CET)