

O encapsulamento en jQuery

Cando se introduciu CSS nas tecnoloxías web fíxose co fin de separar o deseño de contido, de forma que puideramos facer referencia ós elementos da páxina dende as follas de estilo.

O método foi desenvolvido facendo uso de **selectores**, que **referencian os elementos en base os seus atributos ou a súa posición** no documento HTML.

Por exemplo o selector

```
p a
//
// fai referencia ó grupo dos enlaces (elementos <a>) que están dentro dun elemento p.
```

jQuery fai uso dos mesmos selectores, dando soporte ós selectores máis comúns empregados en CSS, e tamén a moitos selectores que aínda non están soportados por moitos navegadores.

Por exemplo o selector **nth-child** que vimos nun exemplo anterior é un bo exemplo da potencia de selección en jQuery.

Para facer referencia a un grupo de elementos, usaremos unha sintaxis moi simple:

```
$(selector)
//
// ou
jQuery(selector)
```

O máis cómodo é empregar a notación `$()`, aínda que resulte un pouco estraña ó principio.

Por exemplo, para recuperar o grupo de enlaces que está dentro dun elemento `p`, usaríamos o seguinte:

```
$("p a")
```

A función **`$()`** (alias de **`jQuery()`**) devolve un obxecto especial JavaScript que contén un array de todos os elementos DOM que cumpren o selector.

En termos de programación, estase a producir un encapsulamento xa que estamos accedendo a todos os elementos seleccionados e estamos aplicándolle unha funcionalidade extendida.

Por exemplo se queremos ocultar todos os elementos `div` dun documento que empregan a clase "colorido", en jQuery faríamos o seguinte:

```
$("#div.colorido").fadeOut();
```

Unha característica especial en moitos comandos de jQuery é que unha vez que ese comando terminou a súa acción (como por exemplo a operación anterior de ocultar `fadeOut()`), ese mesmo comando devolve o mesmo grupo de elementos, listos para usar nunha nova acción.

Por exemplo, se queremos engadir unha nova clase os elementos que ocultamos recentemente poderíamos facer:

```
$("#div.colorido").fadeOut().addClass("eliminados");
```

Este encadeamento de accións podería continuar indefinidamente.

Outro exemplo:

```
$("#unelemento").html("Engadimos texto ó elemento");
//
// ou
$("#unelemento")[0].innerHTML="Engadimos texto ó elemento";
```

No exemplo anterior empregamos un selector de ID ("`#id`"), polo que soamente un obxecto do documento cumprirá ese selector. No primeiro caso empregamos o método `html()`, o cal reempraza o contido do elemento co novo contido HTML.

Se queremos face-lo mesmo pero con varios elementos poríamos:

```
$("#div.novotexto").html("Engadimos texto a un grupo de div que están usando a clase novotexto");

ou

var elements = $("#div.novotexto");
for(i=0;i<elements.length;i++)
elements[i].innerHTML="Engadimos texto a un grupo de div que están usando a clase novotexto";
```

Outros exemplos:

```
$("#p:even"); //Este selector selecciona todos os párrafos que sexan pares.
$("#tr:nth-child(1)"); //Este selector selecciona a primeira fila de tódalas táboas do documento.
$("#body > div"); //Este selector selecciona os <div> fillos que colgan directamente de <body>.
$("#a[href$=pdf]"); //Este selector selecciona enlaces a ficheiros PDF.
$("#body > div:has(a)"); //Este selector selecciona <div> que son fillos directos de <body> e que ademáis conteñan enlaces.
```