



C.F.G.S.: DESARROLLO DE APLICACIONES WEB
Módulo: DESARROLLO WEB EN ENTORNO CLIENTE

02 SELECTORES Y FILTROS. MODIFICACIÓN DE ESTILOS Y HTML

1. SELECTORES

El primer paso a la hora de trabajar con jQuery es seleccionar los elementos del DOM sobre los que queremos trabajar.

La selección de nodos se realiza a través de la función **`$(selector)`**, donde *selector* es una cadena de caracteres que sirve como criterio para identificar los nodos. Admite cualquier cadena que sea un selector CSS válido. Podemos utilizar todos los selectores CSS que ya conocemos:

- De etiqueta: 'p', 'div', 'a'
- De id: '#caja'
- De clase: '.titulo'
- De atributo: "[atributo]" tienen ese atributo, "[atributo='valor']" tienen ese atributo con ese valor, "[atributo!=valor]" atributo con valor distinto
- De grupo: 'p, span'
- Para formularios ':input'

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

| Selector | Example | Selects |
|-------------------------|------------------------------|--------------------------------|
| <code>*</code> | <code>\$("*")</code> | All elements |
| <code><u>#id</u></code> | <code>\$("#lastname")</code> | The element with id="lastname" |

| | | |
|---------------------|---------------------------------|---|
| <u>.class</u> | <code>\$(".intro")</code> | All elements with class="intro" |
| <u>.class.class</u> | <code>\$(".intro,.demo")</code> | All elements with the class "intro" or "demo" |
| <u>element</u> | <code>\$("p")</code> | All <p> elements |
| <u>el1,el2,el3</u> | <code>\$("h1,div,p")</code> | All <h1>, <div> and <p> elements |
| <u>:input</u> | <code>\$(":input")</code> | All input elements |
| <u>:text</u> | <code>\$(":text")</code> | All input elements with type="text" |
| <u>:password</u> | <code>\$(":password")</code> | All input elements with type="password" |
| <u>:radio</u> | <code>\$(":radio")</code> | All input elements with type="radio" |
| <u>:checkbox</u> | <code>\$(":checkbox")</code> | All input elements with type="checkbox" |
| <u>:submit</u> | <code>\$(":submit")</code> | All input elements with type="submit" |
| <u>:reset</u> | <code>\$(":reset")</code> | All input elements with type="reset" |
| <u>:button</u> | <code>\$(":button")</code> | All input elements with type="button" |
| <u>:image</u> | <code>\$(":image")</code> | All input elements with type="image" |

2. FILTROS

Además jQuery nos permite aplicar filtros a los resultados de los selectores anteriores. Estos filtros se construyen añadiendo el signo dos puntos: al selector y el nombre del filtro (que puede requerir algún argumento).

Los filtros de selectores pueden clasificarse en las siguientes categorías:

2.1. Filtros básicos. Determinan los nodos seleccionados por su posición dentro del objeto jQuery devuelto por la función \$. En la siguiente tabla se describen algunos de los más importantes.

| Filtro | Descripción |
|----------------------|---|
| :first | Reduce la selección al primer elemento del objeto jQuery. |
| :last | Reduce la selección al último elemento del objeto jQuery. |
| :eq(<i>índice</i>) | Reduce la selección al elemento que ocupa la posición <i>índice</i> dentro del objeto jQuery. El índice 0 corresponde al primer elemento. |
| :gt(<i>índice</i>) | Reduce la selección a los elementos que ocupan una posición estrictamente mayor (no igual) que <i>índice</i> dentro del objeto jQuery. |
| :lt(<i>índice</i>) | Reduce la selección a los elementos que ocupan una posición estrictamente menor (no igual) que <i>índice</i> dentro del objeto jQuery. |
| :even | Reduce la selección a los elementos cuyos índices en el objeto jQuery son pares, es decir, a los elementos que ocupan las posiciones impares (el primero, el tercero, ...). |
| :odd | Reduce la selección a los elementos cuyos índices en el objeto jQuery son impares, es decir, a los elementos que ocupan las posiciones pares (el segundo, el cuarto, ...). |
| :not(selector) | Reduce la selección a los elementos que no pueden ser identificados con selector. |

| | | |
|---------------|----------------|------------------------|
| <u>:first</u> | \$("#p:first") | The first <p> element |
| <u>:last</u> | \$("#p:last") | The last <p> element |
| <u>:even</u> | \$("#tr:even") | All even <tr> elements |
| <u>:odd</u> | \$("#tr:odd") | All odd <tr> elements |

| | | |
|----------------|---------------------|--|
| :eq(index) | \$("li:eq(3)") | The fourth element in a list (index starts at 0) |
| :gt(no) | \$("li:gt(3)") | List elements with an index greater than 3 |
| :lt(no) | \$("li:lt(3)") | List elements with an index less than 3 |
| :not(selector) | \$("p:not(:first)") | Párrafos que no son el primero |

2.2. Filtros de contenido. Aplican criterios sobre el contenido de los elementos incluidos en el objeto jQuery devuelto por la función \$.

| Filtro | Descripción |
|------------------|---|
| :contains(texto) | Reduce la selección a los elementos en cuyo contenido se encuentra el texto indicado, bien directamente dentro del elemento o bien en cualquiera de sus descendientes. |
| :empty | Reduce la selección a los elementos vacíos; Útil por ejemplo para las celdas vacías de una tabla \$("td:empty") son las celdas vacías |
| :has(selector) | Reduce la selección a los elementos que tienen como descendiente (de cualquier nivel) un elemento identificado por selector. \$("div:has(p)") son los div que tienen párrafos dentro |
| :parent | Este filtro es el contrario de :empty, pues reduce la selección a los elementos que son padres, incluso aunque sus hijos sean nodos de texto en lugar de nodos de elemento. \$("td:parent") son las celdas no vacías porque son "padres" del texto que contienen |

Ejemplo 01

2.3. Filtros de visibilidad.

Los filtros de esta categoría son

`:hidden` y `:visible`, que reducen la selección a los elementos ocultos y visibles, respectivamente.

jQuery considera un elemento **oculto** cuando no ocupa espacio en el documento, es decir, cuando:

- Su propiedad CSS `display` posee el valor `none`
- Cuando sus propiedades CSS `width` y `height` poseen el valor 0
- Si se trata de un elemento `input` de tipo `hidden`.

Pero no si la propiedad CSS `visibility` posee el valor `hidden`.

Ejemplo 02

2.4. **Filtros de hijos.** Estos filtros reducen la selección a los elementos que ocupan una determinada posición entre sus hermanos.

| Filtro | Descripción |
|-----------------------------------|--|
| <code>:first-child</code> | Reduce la selección a los elementos que son hijos primogénitos. \$(<code>"p:first-child"</code>) son los párrafos que son hijo primero de algún elemento |
| <code>:last-child</code> | Reduce la selección a los elementos que son hijos benjamines (hijos menores). |
| <code>:nth-child(posición)</code> | Reduce la selección a los elementos que ocupan la <i>posición</i> indicada entre sus hermanos, considerando que posición empieza a contar por 1 para el primogénito. |
| <code>:only-child</code> | Reduce la selección a los elementos que son hijos únicos. |

Ejemplo 03

2.5. Otros filtros

| | | |
|------------------------|------------------------------|-----------------------------|
| <code>:enabled</code> | <code>\$(":enabled")</code> | All enabled input elements |
| <code>:disabled</code> | <code>\$(":disabled")</code> | All disabled input elements |
| <code>:selected</code> | <code>\$(":selected")</code> | All selected input elements |

| | | |
|------------------------|------------------------------|----------------------------|
| <code>:checked</code> | <code>\$(":checked")</code> | All checked input elements |
| <code>:animated</code> | <code>\$(":animated")</code> | All animated elements |

2.6. El método is()

El método `is(selector)` es muy útil para averiguar si al menos uno de los elementos del objeto jQuery sobre el que se aplica puede ser identificado por el selector aportado como argumento.

`$("p").is(":hidden")` devolverá true si hay algún párrafo oculto

`$("#OK").is(":checked")` devolverá true si el element OK está chequeado

Ejemplo 04

3. Estilos

jQuery incluye una manera útil de obtener y establecer propiedades CSS a los elementos.

```
$('h1').css('font-size');  
// devuelve en una cadena de caracteres el valor de la propiedad, p.ej. "19px"
```

```
$('h1').css('font-size', '100px');  
// establece una propiedad individual CSS
```

```
$('h1').css({ 'font-size' : '100px',  
              'color' : 'red' });  
// establece múltiples propiedades CSS
```

```
$('h1').css({ 'font-size' : '+=15px',  
              'paddingTop' : '+=20px'  
              });  
// suma 15px al tamaño fuente del elemento, suma 20px al padding superior
```

Ejemplo 05

4. Modificar html

Podemos cambiar el código html de un elemento de la siguiente manera

```
$(selector).html("nuevo código");
```

Por ejemplo:

```
$("p").html("Párrafos cambiados");
```

Y obtener su código html si al método no le pasamos contenido

```
$("p:first").html() devuelve el contenido html del primer párrafo
```

Si hay varios elementos al utilizar el selector devuelve el del primero

```
$("p").html() devuelve el contenido html del primer párrafo
```

5. Otros métodos

- `$(selector).val()` retorna el valor del objeto jquery correspondiente (p.ej. el value de un input text),

Por ejemplo:

```
$("#num").val() devuelve el valor del elemento que tiene id=num
```

Si le pasamos un valor asigna a ese selector el valor pasado

```
$("#num").val(25)
```

- `$(selector).change()` (función a realizar cuando cambie el valor de selector)

Por ejemplo:

```
$("#num").change(cargarTabla) cuando se modifica el valor del elemento que tiene id=num se realiza la función cargarTabla
```

Ejemplo 06