

JQuery

- Librería Javascript.
- Documentación: <http://docs.jquery.com>
- Para utilizarla:
 - Incluir la librería
 - Utilizar la etiqueta `<script>`

¿DOM Ready?

No es posible interactuar con el contenido de una página hasta que no se encuentre preparado para su manipulación.

El bloque `$(document).ready()`

```
$(document).ready(function() {  
    console.log('el documento está preparado');  
});
```

Selección de elementos

El concepto más básico de jQuery es el de "seleccionar algunos elementos y realizar acciones con ellos".

La biblioteca soporta gran parte de los selectores CSS3 y varios más no estandarizados.

Selección de elementos

Técnicas comunes para la selección de elementos:

En base a su ID

`$ (' #myId ') ;` // notar que los IDs deben ser únicos por página

En base al nombre de clase

`$ (' div.myClass ') ;` // si se especifica el tipo de elemento, se mejora el rendimiento de la selección

Selección por su atributo

`$ (' input[name=first_name] ') ;` // puede ser muy lento

Selección de elementos

Técnicas comunes para la selección de elementos:

Selección de selector CSS

```
$ ( '#contents ul.people li' );
```

Pseudo-selectores

```
$ ( 'a.external:first' ); // selecciona el primer elemento  
<a> con la clase 'external'
```

```
$ ( 'tr:odd' ); // selecciona todos los elementos <tr> impares  
de una tabla
```

Selección de elementos

Técnicas comunes para la selección de elementos:

```
$ ( ' #myForm :input ' ) ; // selecciona todos los elementos  
del tipo input dentro del formulario #myForm
```

```
$ ( ' div:visible ' ) ; // selecciona todos los divs visibles
```

NOTA Cuando se utilizan los pseudo-selectores `:visible` y `:hidden`, jQuery comprueba la visibilidad actual del elemento pero no si éste posee asignados los estilos CSS `visibility` o `display` — en otras palabras, verifica si el alto y ancho físico del elemento es mayor a cero.

Comprobar selecciones

Una vez realizada la selección de los elementos, querrá conocer si dicha selección entregó algún resultado.

Para ello, pueda que escriba algo así:

✗ `if ($('div.foo')) { ... }` Siempre devuelve TRUE

✓ `if ($('div.foo').length) { ... }` Chequea si el objeto devuelto tiene elementos.

Guardar selecciones en una variable

```
var $divs = $( 'div' );
```

Cada vez que se hace una selección, una gran cantidad de código es ejecutado. jQuery no guarda el resultado por si solo, por lo tanto, si va a realizar una selección que luego se hará de nuevo, deberá salvar la selección en una variable.

Refinamiento y filtrado de selecciones

`$('div.foo').has('p');` // el elemento div.foo contiene elementos <p>

`$('h1').not('.bar');` // el elemento h1 no posee la clase 'bar'

`$('ul li').filter('.current');` // item de li que posee la clase 'current'

`$('ul li').first();` // el primer item de una lista desordenada

`$('ul li').eq(5);` // el sexto item de una lista desordenada

Selección de elementos de un formulario

- :button Selecciona elementos `<button>` y con el atributo `type='button'`
- :checkbox Selecciona elementos `<input>` con el atributo `type='checkbox'`
- :checked Selecciona elementos `<input>` del tipo checkbox seleccionados
- :disabled Selecciona elementos del formulario que están deshabilitados
- :enabled Selecciona elementos del formulario que están habilitados
- :file Selecciona elementos `<input>` con el atributo `type='file'`
- :image Selecciona elementos `<input>` con el atributo `type='image'`
- :input Selecciona elementos `<input>`, `<textarea>` y `<select>`
- :password Selecciona elementos `<input>` con el atributo `type='password'`

Selección de elementos de un formulario

:radio Selecciona elementos <input> con el atributo type='radio'

:reset Selecciona elementos <input> con el atributo type='reset'

:selected Selecciona elementos <options> que están seleccionados

:submit Selecciona elementos <input> con el atributo type='submit'

:text Selecciona elementos <input> con el atributo type='text'

```
$ ( ' #myForm :input ' ) ;
```

obtiene todos los elementos inputs dentro del formulario #myForm

Trabajar con selecciones

Encadenamiento

Si en una selección se realiza una llamada a un método, y éste devuelve un objeto jQuery, es posible seguir un "encadenado" de métodos en el objeto.

```
$( '#content' )  
    .find( 'h3' )  
    .eq( 2 )  
        .html( 'nuevo texto para el tercer elemento h3' )  
    .end( ) // reestablece la selección a todos los elementos h3 en #content  
    .eq( 0 )  
        .html( 'nuevo texto para el primer elemento h3' );
```

Obtenedores (getters) y establecedores (setters)

jQuery "sobrecarga" sus métodos, en otras palabras, el método para establecer un valor posee el mismo nombre que el método para obtener un valor. Cuando un método es utilizado para establecer un valor, es llamado método establecedor (en inglés setter). En cambio, cuando un método es utilizado para obtener (o leer) un valor, es llamado obtenedor (en inglés getter).

El método `$.fn.html` utilizado como establecedor

```
$ ( 'h1' ).html ( 'hello world' );
```

El método `html` utilizado como obtenedor

```
$ ( 'h1' ).html ( );
```

CSS, estilos y dimensiones

Obtener propiedades de CSS

```
$('h1').css('fontSize'); // devuelve una cadena de caracteres "19px"
```

Establecer una propiedad individual

```
$('h1').css('fontSize', '100px');  
$('h1').css('fontSize', '+=10px');
```

Establecer propiedades multiples

```
$('h1').css({ 'fontSize' : '100px', 'color' : 'red' });
```

Trabajar con clases

```
var $h1 = $( 'h1' );
```

```
$h1.addClass( 'big' );  
$h1.removeClass( 'big' );  
$h1.toggleClass( 'big' );
```

```
if ( $h1.hasClass( 'big' ) ) { ... }
```

Las clases también pueden ser útiles para guardar información del estado de un elemento, por ejemplo, para indicar que un elemento fue seleccionado.

Dimensiones

Métodos básicos sobre Dimensiones

`('h1').width('50px');` // establece el ancho de todos los elementos H1

`$('h1').width();` // obtiene el ancho del primer elemento H1

`$('h1').height('50px');` // establece el alto de todos los elementos H1

`$('h1').height();` // obtiene el alto del primer elemento H1

`$('h1').position();` // devuelve un objeto conteniendo información sobre la posición del primer elemento relativo al "offset" (posición) de su elemento padre

Atributos

Los atributos de los elementos HTML que conforman una aplicación pueden contener información útil, por eso es importante poder establecer y obtener esa información.

Establecer atributos

```
$( 'a' ).attr( {  
    'title' : 'all titles are the same too',  
    'href'  : 'somethingNew.html'  
} );
```

Obtener atributos

```
$( 'a' ).attr( 'href' );
```

Recorrer el DOM

`$('h1').next('p');` // seleccionar el próximo elemento <p> con respecto a H1

`$('div:visible').parent();` // seleccionar el elemento contenedor a un div visible

`$('input[name=first_name]').closest('form');` // seleccionar el elemento <form> más cercano a un input

`$('#myList').children();` // seleccionar todos los elementos hijos de #myList

`$('li.selected').siblings();` // seleccionar todos los items hermanos del elemento