

JQuery

Antonio Espín Herranz

Contenidos

- Introducción.
- Selectores.
- Eventos.
- Efectos.
- Manipulación del DOM.
- Peticiones AJAX.
- Trabajo con Formularios.

jQuery

- Es una alternativa a la librería prototype.
- Tiene una versión empaquetada que ocupa 20 kb. Interesante a la hora de la descarga de nuestro sitio Web. En la zona de descargas tenemos una versión **pack**.
- La instalación: adjuntando el fichero de jquery a nuestro proyecto en una etiqueta script.
- La documentación la tenemos en:
<http://docs.jquery.com/>

La función \$

- Igual que en prototype pero esta es mas potente.
- La cadena de texto que se pasa como parámetro puede hacer uso de Xpath o de CSS para seleccionar los elementos.
- Separando expresiones con un carácter "," se puede seleccionar un número ilimitado de elementos.
 - En este caso nos devolvería una colección.

Ejemplos

// Selecciona todos los enlaces de la página:

`$('a')`

// Selecciona el elemento cuyo id sea "primero":

`$('#primero')`

// Selecciona todos los h1 con class "titular":

`$('h1.titular')`

// Selecciona todo lo anterior :

`$('a, #primero, h1.titular')`

Soporta casi todos los
selectores de CSS3 y XPath

Mas ejemplos de la función \$

// Selecciona todos los párrafos de la página que tengan al menos un enlace

`$('#p[a]')`

// Selecciona todos los radiobutton de los formularios de la página

`$('#input:radio')`

// Selecciona todos los enlaces que contengan la palabra "Imprimir"

`$('#a:contains("Imprimir")');`

// Selecciona los div que no están ocultos

`$('#div:visible')`

Funciones para eventos

- Una de las utilidades más interesantes de jQuery está relacionada con el evento ***onload*** de la página.
- Las aplicaciones Web más complejas suelen utilizar un código similar al siguiente para iniciar la aplicación:
window.onload = function() { ... };

Problemas con window.onload

- Cuando utilizamos este evento nos encontramos con el problema (**por lentitud**) que tenemos que esperar a que se descargue al completo la página (imágenes, ficheros adjuntos, etc.) y que se haya construido el árbol DOM por completo.
- Hasta que no se cargue el código no podemos seleccionar elementos de nuestra página (no los vamos a encontrar).
 - Tendríamos que utilizar este evento pero la aplicación se ralentiza un poco.

La alternativa jQuery

- **jQuery propone:**
`$(document).ready(function() { ... });`
*// Recibe por parámetro la función manejadora. Que en
// este caso la estamos asignando de forma anónima.*
- **¿La ventaja?** → la aplicación no espera a que se carguen todos los elementos de la página.
- Sólo espera a que se haya descargado el contenido HTML de la página, con lo que el árbol DOM ya está disponible para ser manipulado.
- La aplicación se inicia antes que con el evento `window.onload`.

Eventos

- ***ready()*** no es más que una de las muchas funciones que componen el módulo de los eventos.
- A) Soporta todos los eventos de JavaScript para asignar comportamiento a los elementos de la página: (*click, mousemove, keypress, etc.*)
- B) Y también podemos ejecutar un determinado evento: (*click(), mouseover(), etc.*)

Ejemplos

// Ejecuta el evento 'onclick' en todos los párrafos de la página:

```
$('p').click();
```

// Ejecuta el evento 'mouseover' sobre un 'div' con id 'menu': **`$('div#menu').mouseover();`**

- **Combina el selector con la función que hace saltar el evento.**

Uso habitual

- Mas que ejecutar un evento lo que vamos a hacer el 99% de las veces es asignar una función manejadora a un evento.

// Establece la función manejadora del evento 'onclick'
// a todos los párrafos de la página:

```
$( 'p' ).click(function() {  
    alert( $(this).text() );  
});
```

Asocia el evento mediante una función anónima. Utiliza función \$ para localizar el elemento que ha recibido el evento.

// Establece la función manejadora del evento 'onblur'
// a los elementos de un formulario:

```
$( '#elFormulario :input' ).blur(function() {  
    valida( $(this) );  
});
```

Utiliza el selector de CSS. Por att. id.

La función toggle()

- Se comporta como una **alternativa** dentro del evento del elemento:

```
$("#p").toggle(  
  function(){  
    alert("Me acabas de activar");  
  },  
  function(){  
    alert("Me acabas de desactivar");  
  });
```

Cada alternativa la estamos representando mediante una function anónima.
En este caso se lo asociamos a los párrafos.

Efectos visuales

- jQuery dispone de funciones para poder ocultar elementos de la página de una forma cómoda:

// Oculta todos los enlaces de la página:

```
$('a').hide();
```

// Muestra todos los 'div' que estaban ocultos:

```
$('div:hidden').show();
```

*// Muestra los 'div' que estaba ocultos y oculta
// los 'div' que eran visibles:*

```
$('div').toggle();
```

Efectos visuales

- Todas las funciones relacionadas con los **efectos visuales** permiten indicar **dos parámetros opcionales**:
 - el primero es la duración del efecto (String: slow, normal, fast) o mediante un número indicando los milisegundos.
 - el segundo parámetro es la función que se ejecuta al finalizar el efecto visual.
- Otros efectos visuales incluidos son los relacionados con el fundido o "*fading*":
 - *fadeIn()* muestra los elementos con un fundido suave,
 - *fadeOut()* oculta los elementos con un fundido suave y
 - *fadeTo()* establece la opacidad del elemento en el nivel indicado)
- El despliegue de elementos (
 - *slideDown()* hace aparecer un elemento desplegándolo en sentido descendente,
 - *slideUp()* hace desaparecer un elemento desplegándolo en sentido ascendente,
 - *slideToggle()* hace desaparecer el elemento si era visible y lo hace aparecer si no era visible).

Chequear

- Podemos preguntar si una capa está oculta.
- Lo podemos hacer así:
`$("#miCapa").is(":hidden")`
// Devuelve true / false.
// miCapa se sustituirá por cualquier id.
- Selecciona la capa mediante el atributo id.

Ajax con jQuery

- El método principal para realizar peticiones AJAX es `$.ajax()` (**importante no olvidar el punto entre \$ y *ajax***).
- A partir de esta función básica, se han definido otras funciones relacionadas, de más alto nivel y especializadas en tareas concretas: `$.get()`, `$.post()`, `$.load()`, etc.

Sintaxis

- `$.ajax(opciones);`
- Ejemplo:
`$.ajax({ url: '/ruta/hasta/pagina.php',
type: 'POST',
async: true,
data: 'parametro1=valor1¶metro2=valor2',
success: procesaRespuesta,
error: muestraError });`

Funciones CSS

- jQuery dispone de varias funciones para la manipulación de las propiedades CSS de los elementos.
- Todas las funciones se emplean junto con una selección de elementos realizada con la función `$()`.
- Si la función obtiene el valor de las propiedades CSS, sólo se obtiene el valor de la propiedad CSS del primer elemento de la selección realizada.
- Si la función establece el valor de las propiedades CSS, se establecen para todos los elementos seleccionados.

Ejemplos

// Obtiene el valor de una propiedad CSS

// En este caso, solo para el primer 'div' de la página

```
$('div').css('background');
```

// Establece el valor de una propiedad CSS

// En este caso, para todos los 'div' de la página

```
$('div').css('color', '#000000');
```

// Establece varias propiedades CSS

// En este caso, para todos los 'div' de la página

```
$('div').css({ padding: '3px', color: '#CC0000' });
```

Ejemplos

// Obtiene la altura en píxel del primer 'div' de la página

```
$('#div').height();
```

// Establece la altura en píxel de todos los 'div' de la página

```
$('#div').height('150px');
```

// Obtiene la anchura en píxel del primer 'div' de la página

```
$('#div').width();
```

*// Establece la anchura en píxel de todos los 'div' de la
página*

```
$('#div').width('300px');
```

Chequear el Navegador

// 'true' para navegadores de la familia Internet Explorer

`$.browser.msie;`

// 'true' para navegadores de la familia Firefox

`$.browser.mozilla;`

// 'true' para navegadores de la familia Opera

`$.browser.opera;`

// 'true' para navegadores de la familia Safari

`$.browser.safari;`

Trabajo con Formularios

- Una posibilidad es utilizar JQuery para la validación de formularios HTML.
- De esta forma si no está soportado por el Navegador la especificación de HTML5 se puede realizar de forma externa con esta librería.
- Utilizar la función \$ para validación.

Ejemplo

```
function validar(){  
    if ($("#v_email").val() == "" || ($("#v_pass").val() == "")){  
        alert('Rellene su email y el password.');        return false;  
  
    } else  
        return true;  
}
```

- **v_email** y **v_pass** son los atributos id de dos cajas de texto.

jQuery Resources

- Project website
 - <http://www.jquery.com>
- Learning Center
 - <http://docs.jquery.com/Tutorials>
 - <http://www.learningjquery.com/>
 - <http://15daysofjquery.com/>
- Support
 - <http://docs.jquery.com/Discussion>
 - <http://www.nabble.com/JQuery-f15494.html> mailing list archive
 - irc.freenode.net irc room: #jquery
- Documentation
 - http://docs.jquery.com/Main_Page
 - <http://www.visualjquery.com>
 - <http://jquery.bassistance.de/api-browser/>
- jQuery Success Stories
 - http://docs.jquery.com/Sites_Using_jQuery