

Programación Web

Tema 3.4 Java Script – Librerías: jQuery

Miguel Ángel Manso
Emerson Castañeda
Ramón Alcarria
ETSI en Topografía, Geodesia y Cartografía - UPM

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Contenido

- Introducción
- La librería jQuery
- Rehaciendo ejemplos
- Otros frameworks

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Introducción (I)

- Las aplicaciones web son cada vez más complejas, ya que incluyen efectos e interacciones que hasta hace poco tiempo eran exclusivas de las aplicaciones de escritorio
- Al mismo tiempo, la programación de estas aplicaciones avanzadas se complica por varios motivos
- En primer lugar, las aplicaciones comerciales deben funcionar correctamente e igual de bien en al menos cinco navegadores diferentes: **Internet Explorer 6 y 7, Firefox, Opera y Safari**

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Introducción (II)

- En segundo lugar, el tiempo disponible para el desarrollo de cada nueva característica se reduce cada vez más por la necesidad continua de incluir novedades en las aplicaciones antes de que las incluya la competencia
- Por todo lo anterior, han surgido librerías y frameworks específicos para el desarrollo de aplicaciones con **JavaScript**
- Utilizando estas librerías, se reduce el tiempo de desarrollo y se tiene la seguridad de que las aplicaciones funcionan igual de bien en cualquiera de los navegadores más populares

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (I)

- **jQuery** <http://jquery.com/> es la librería **JavaScript** que ha irrumpido con más fuerza como alternativa a **Prototype**
- Su autor original es John Resig, al igual que sucede con las demás librerías, recibe contribuciones de varios programadores
- Su versión comprimida ocupa 20 KB
- La documentación de **jQuery** es muy completa e incluye muchos ejemplos <http://docs.jquery.com/>

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (II): Funciones y métodos de jQuery

- La función básica de **jQuery** y una de las más útiles es `$()`.
- La cadena de texto que se pasa como parámetro puede hacer uso de **Xpath** o de **CSS** para seleccionar los elementos. Además, separando expresiones con un carácter `,` se puede seleccionar un número ilimitado de elementos

```
// 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')
```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (III)

- Las posibilidades de la función `$()` van mucho más allá de estos los anteriores ejemplos, ya que soporta casi todos los selectores definidos por **CSS3** y también permite utilizar **Xpath**

```
// 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')
// Selecciona todos los elementos pares de una lista
$('ul#menuPrincipal li:even')
// Selecciona todos los elementos impares de una lista
$('ul#menuPrincipal li:odd')
// Selecciona los 5 primeros párrafos de la página
$('p:lt(5)')
```

- Debido a que las posibilidades de la función `$()` son prácticamente ilimitadas, la documentación disponibles de **jQuery** sobre los selectores <http://docs.jquery.com>Selectors> es la mejor forma de descubrir todas sus posibilidades

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (IV)

- 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() {
    ...
};
```

- Hasta que no se carga la página, el navegador no construye el árbol **DOM**, lo que significa que no se pueden utilizar funciones que seleccionen elementos de la página, ni se pueden añadir o eliminar elementos
- El problema de **window.onload** es que el navegador espera a que la página se cargue completamente, incluyendo todas las imágenes y archivos externos que se hayan enlazado

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (V)

- **jQuery** propone el siguiente código para ejecutar las instrucciones una vez que se ha cargado la página

```
$(document).ready(function() {
  ...
});
```

- La gran ventaja del método propuesto por **jQuery** es que la aplicación no espera a que se carguen todos los elementos de la página, sino que 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
- De esta forma, las aplicaciones **JavaScript** desarrolladas con **jQuery** pueden iniciarse más rápidamente que las aplicaciones **JavaScript** tradicionales

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (VI)

- En realidad, **ready()** no es más que una de las muchas funciones que componen el módulo de los eventos. Todos los eventos comunes de **JavaScript** (click, mousemove, keypress, etc.) disponen de una función con el mismo nombre que el evento
- Si se utiliza la función sin argumentos, se ejecuta el evento

```
// 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();
```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (VII)

- No obstante, el uso más habitual de las funciones de cada evento es el de establecer la función manejadora que se va a ejecutar cuando se produzca el evento

```
// Establece la función manejadora del evento 'onclick'
// a todos los párrafos de la página
$('p').click(function() {
  alert($(this).text());
});

// Establece la función manejadora del evento 'onblur'
// a los elementos de un formulario
$('#formulario :input').blur(function() {
  valida($(this));
});
```

- Entre las utilidades definidas por **jQuery** para los eventos se encuentra la función **toggle()**, que permite ejecutar dos funciones de forma alterna cada vez que se pincha sobre un elemento

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

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (VIII)

- Las aplicaciones web más avanzadas incluyen efectos visuales complejos para construir interacciones similares a las de las aplicaciones de escritorio. **jQuery** incluye en la propia librería varios de los efectos más comunes

```
// 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();
```

- Todas las funciones relacionadas con los efectos visuales permiten indicar dos parámetros opcionales: el primero es la duración del efecto y el segundo parámetro es la función que se ejecuta al finalizar el efecto visual

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

jQuery (IX)

- 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
 - ***fadeTo()*** establece la opacidad del elemento en el nivel indicado y 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

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Rehaciendo ejemplos (IV)

- EJ2: realizar peticiones **AJAX** al servidor para comprobar si un determinado nombre de usuario esta libre

```

30
31 function procesaRespuesta() {
32     if(peticion_http.readyState == READY_STATE_COMPLETE) {
33         if (peticion_http.status == 200) {
34             var login = document.getElementById("login").value;
35             if(peticion_http.responseText == "si") {
36                 document.getElementById("disponibilidad").innerHTML =
37                     "El nombre elegido ["+login+"] está disponible";
38             }
39             else {
40                 document.getElementById("disponibilidad").innerHTML =
41                     "NO está disponible el nombre elegido ["+login+"]";
42             }
43         }
44     }
45 }
46 window.onload = function() {
47     document.getElementById("comprobar").onclick = comprobar;
48 }

```

Rehaciendo ejemplos (VII)

- **jQuery** permite simplificar notablemente el código de la aplicación original – EJ6

```

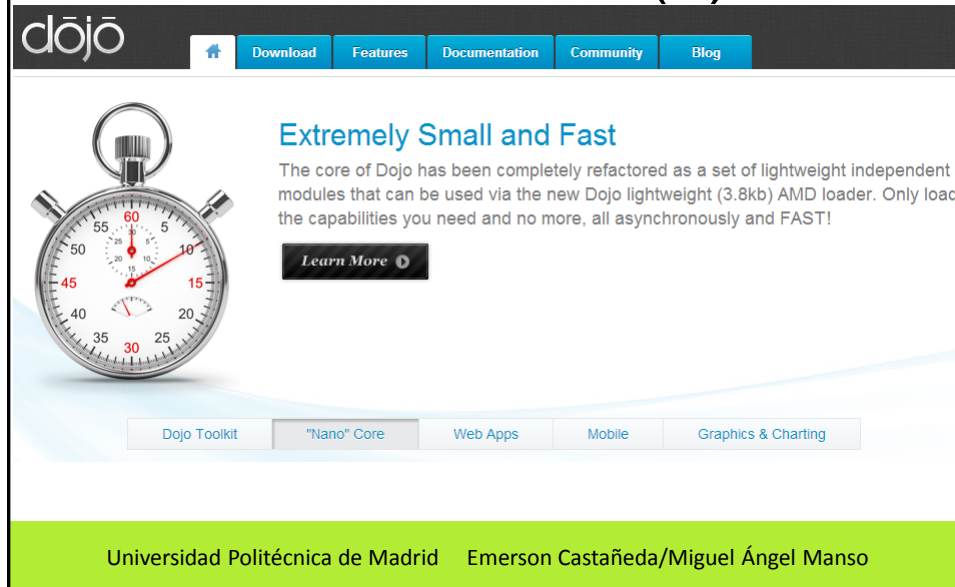
9  function comprobar() {
10     var login = $('#login').val();
11     var peticion = $.ajax({
12         url: 'compruebaDisponibilidad.php?nocache=' + Math.random(),
13         type: 'POST',
14         data: { login: login },
15         success: function(respuesta) {
16             $('#disponibilidad').html((respuesta == 'si') ?
17                 'El nombre elegido ['+login+] está disponible' :
18                 'NO está disponible el nombre elegido ['+login+]');
19         },
20         error: function() { alert('Se ha producido un error'); }
21     });
22 }
23
24 $(document).ready(function() {
25     $('#comprobar').click(comprobar);
26 });

```

Otros frameworks (I)

- La popularidad de las aplicaciones web con interfaces dinámicas complejas y que incluyen múltiples interacciones **AJAX** ha provocado la aparición de un gran número de frameworks especializados para el desarrollo de aplicaciones con **JavaScript**. Además de **Prototype** y **jQuery**, existen otros frameworks destacables
- **Dojo** <http://dojotoolkit.org/> Sus propios creadores lo denominan "el conjunto de herramientas ("toolkit") de JavaScript que permite desarrollar aplicaciones web profesionales de forma sencilla y más rápida". Dispone de una licencia de tipo software libre

Otros frameworks (II)



dojō

Download Features Documentation Community Blog

Extremely Small and Fast

The core of Dojo has been completely refactored as a set of lightweight independent modules that can be used via the new Dojo lightweight (3.8kb) AMD loader. Only load the capabilities you need and no more, all asynchronously and FAST!

Learn More

Dojo Toolkit "Nano" Core Web Apps Mobile Graphics & Charting

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Otros frameworks (III)

- **Mootools** <http://mootools.net/> es un framework que destaca por su reducido tamaño y por lo modular de su desarrollo. Al descargar el framework, se pueden elegir los componentes que se van a utilizar, para descargar una versión comprimida que sólo contenga los componentes escogidos. De esta forma, se puede reducir al mínimo el tamaño de los archivos descargados por los usuarios

Otros frameworks (IV)

mootools
a compact javascript framework

Home | Download | Docs | Forge | Blog | Demos

MooTools is a compact, modular, Object-Oriented JavaScript framework designed for the intermediate to advanced JavaScript developer. It allows you to write powerful, flexible, and cross-browser code with its elegant, well documented, and coherent API.

MooTools code respects strict standards and doesn't throw any warnings. It's extensively documented and has meaningful variable names: a joy to browse and a snap to understand.

Open Source License
MooTools is released under the Open Source MIT license, which gives you the possibility to use it and modify it in every circumstance.

Browser Compatibility
MooTools is compatible and fully tested with Safari, Internet Explorer 6+, Firefox, Opera, and Chrome.

Developer Tools
[Core Builder](#)
[More Builder](#)

Development
[GitHub Repository for Core](#)
[GitHub Repository for More](#)
[Bugs for Core](#)
[Bugs for More](#)

Resources
[Wear MooTools! \(US, EU\)](#)
[IRC Channel](#)
[MooTools on Twitter](#)
[MooTools Users Group](#)
[Unofficial MooTools Forum](#)
[The MooTorial](#)

People
[MooTools Developers](#)

Pro JavaScript with MooTools out now!
Ready to take your MooTools knowledge to a new level? Then [grab a copy](#) of *Pro JavaScript with MooTools* by Mark Obena, with a Foreword by MooTools Founder and Lead Developer Valerio Proietti. Geared towards intermediate and advanced users, *Pro JavaScript with MooTools* explores advanced features of the JavaScript language as well as the internals of MooTools 1.3. So if you're hungry for milk, [get your copy today](#).

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Otros frameworks (V)

- **Sencha Ext JS** <http://www.sencha.com/products/extjs/> es otro de los frameworks más populares de JavaScript. Comenzó siendo un añadido de la librería YUI de Yahoo <http://developer.yahoo.com/yui/>, pero luego adquirió entidad propia. Además de las utilidades comunes, incluye una serie de componentes listos para usar y tiene una licencia de tipo software libre y otra licencia de tipo comercial para obtener soporte técnico

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

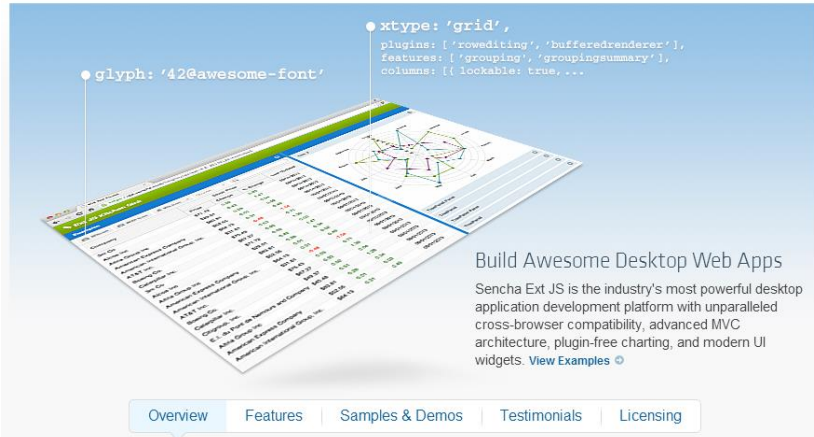
Otros frameworks (VI)

Home | Products

 **Sencha Ext JS** JavaScript Framework for Rich Desktop Apps

Buy

Download 



• glyph: '42@awesome-font'

• xtype: 'grid',
plugins: ['rowediting', 'bufferedrenderer'],
features: ['grouping', 'groupingsummary'],
columns: [{ lockable: true, ...

Build Awesome Desktop Web Apps

Sencha Ext JS is the industry's most powerful desktop application development platform with unparalleled cross-browser compatibility, advanced MVC architecture, plugin-free charting, and modern UI widgets. [View Examples](#)

Overview | Features | Samples & Demos | Testimonials | Licensing

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso