

# **Capítulo 15**

## ***Programación del lado del cliente***

### **Objetivo**

Al finalizar el capítulo, el alumno:

- Usar Ajax control tool kit.
- Utilizar JavaScript en aplicaciones Web.
- Utilizar jQuery para simplificar el acceso al DOM.
- Utilizar jQuery en Páginas ASP.NET.

### **Temas**

1. ASP.NET AJAX
2. Controles básicos de AJAX
3. Instalación y uso de AJAX Toolkit
4. Introducción a jQuery
5. Usando jQuery en ASP.NET
6. Introducción a JSON
7. AJAX con jQuery

## 1. Introducción a jQuery

### Introducción a jQuery

- jQuery es una biblioteca o framework de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

---

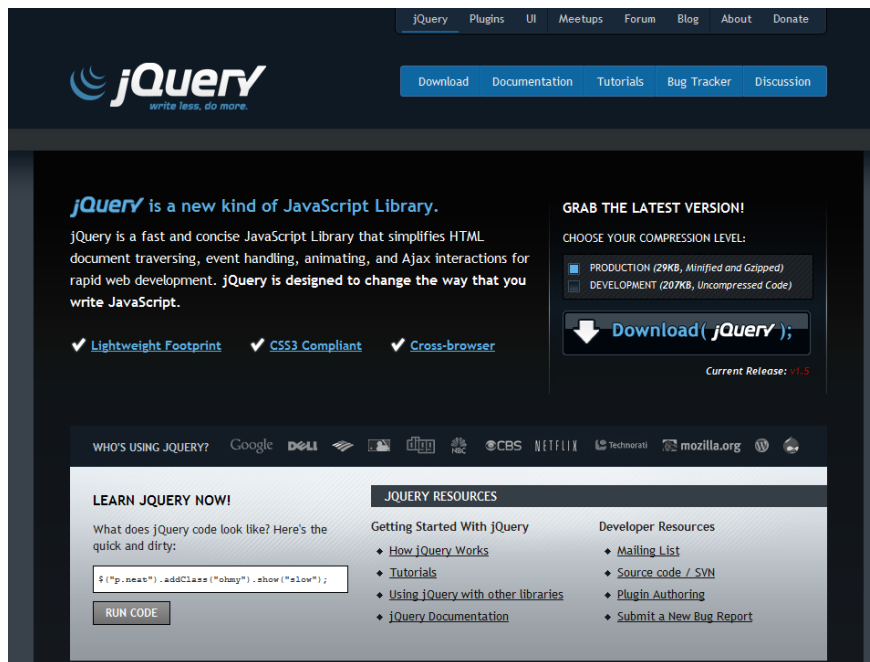
jQuery es una biblioteca o framework basado en Javascript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery es software libre y de código abierto, jQuery ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

La biblioteca jQuery aporta las siguientes ventajas:

- Ahorra líneas de código.
- Hace transparente el soporte del código JavaScript de nuestra aplicación para los navegadores principales.
- Provee de un mecanismo para la captura de eventos.
- Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- Integra funcionalidades para trabajar con AJAX.

La página oficial de jQuery es <http://www.jquery.com> y desde aquí se descarga la biblioteca JQuery.



Actualmente, existen 2 versiones de desarrollo en jQuery; está el desarrollo 1.x que se enfoca en brindar soporte a todos los navegadores incluyendo IE desde la versión 6; y está el desarrollo 2.x que se enfoca en adoptar nuevas características, pero que no tiene soporte para IE 6, 7 y 8.

## 2. Usando jQuery en ASP.NET

### Usando jQuery en ASP.NET

#### Con jQuery

```
// Seleccionar un elemento
$("#miParrafo");
// Seleccionar todos los enlaces
$("a");
// Seleccionar todo lo anterior
$("#miParrafo, a")
// Seleccionar mediante la clase CSS
$(".miClase")
```

Para utilizar jQuery, solamente es necesario descargar la librería y enlazar en nuestras páginas el único archivo JavaScript de la siguiente forma:

```
<script src="js/jquery-1.4.4.min.js" type="text/javascript"></script>
```

En este caso se ve que se está referenciando el archivo jquery-1.4.4.min.js, este lleva como parte de su nombre la palabra **min** porque se trata de una versión **minificada** del archivo, esto quiere decir que se le quitaron varios espacios y se renombraron las variables de dicho archivo, todo esto con el propósito de que ocupe menos espacio y pueda ser transferido con más rapidez cuando sea requerido por un request.

Una de las instrucciones más utilizadas en las aplicaciones JavaScript es la siguiente:

```
window.onload = function() { ... }
```

La instrucción anterior permite que el código JavaScript se ejecute luego de que la página se ha cargado por completo. Se trata de una instrucción imprescindible en las aplicaciones web complejas, porque cualquier modificación en la página o la selección de cualquier elemento necesita que el árbol DOM de la página se haya construido por completo, y por tanto, que esté disponible el código HTML de la página entera. jQuery dispone de una instrucción muy similar equivalente a la anterior:

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

ó

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

La gran ventaja de esta función de jQuery, es que la página se puede manipular en cuanto se ha cargado su código HTML (y por tanto, se ha construido el árbol DOM de la página) mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales.

## **Selectores**

La función `getElementById()` es posiblemente la más utilizada al desarrollar aplicaciones web, ya que permite seleccionar un elemento a partir de su atributo "id". Además, existen otras dos funciones relacionadas: `getElementsByName()` y `getElementsByTagName()`.

jQuery incluye la función `$()` como una alternativa más rápida y completa para seleccionar cualquier elemento de la página:

Ejemplo:

### **Documento HTML a referenciar**

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<p id="miParrafo" class="miClase"></p>
<a href="http://www.google.com"></a>
<a href="http://www.microsoft.com"></a>
<p id="parrafoDeCierre" class="miClase"></p>
</body>
</html>
```

## Con JavaScript

```
// Seleccionar un elemento por su ID
document.getElementById("miParrafo");
// Seleccionar todos los enlaces
document.getElementsByTagName("a");
// Seleccionar todo lo anterior
?
// Seleccionar mediante la clase CSS
?
```

## Con jQuery

```
// Seleccionar un elemento por su ID
$("#miParrafo");
// Seleccionar todos los enlaces
$("a");
// Seleccionar todo lo anterior
$("#miParrafo, a")
// Seleccionar mediante la clase CSS
$(".miClase")
```

## Selectores avanzados

Uno de los componentes más impresionantes de jQuery es el de sus selectores, que permiten olvidarse de los aburridos `document.getElementById()` y del código JavaScript complejo. jQuery utiliza lo mejor de CSS 1, CSS 2, CSS 3 y XPath para seleccionar de forma sencilla cualquier elemento de la página sin necesidad de saturar el código XHTML con atributos "id". Ejemplos:

```
// Seleccionar todos los párrafos de la página que tengan al menos un enlace
$("p[a]")
// Seleccionar todos los "div" que no estén ocultos
$("div:visible")
// Seleccionar todos los radio buttons que han sido seleccionados
$("input[@type=radio][@checked]")
// Seleccionar varios elementos mediante sus "id" y "class" (devuelve un array)
$("p.importante, div#menu, span#introduccion p.especial a")
```

Además, jQuery soporta muchos de los selectores que define CSS 3, algo que los navegadores actuales solo pueden soñar con tener algún día, como `:nth-child(n)`, `:empty`, `:not`, `:disabled`.

Para los programadores más acostumbrados a trabajar con XML, jQuery también permite utilizar XPath (lenguaje de selección de elementos y atributos en XML) para seleccionar elementos, como por ejemplo:

```
// Seleccionar todos los párrafos de la página
$("html/body/p");
// Seleccionar todos los enlaces cuyo atributo "rel" valga "nofollow"
$("a[@ref='nofollow']");
```

jQuery es tan avanzado que incluso añade sus propios selectores que no existen ni en CSS ni en XPath, pero que facilitan mucho el desarrollo de las aplicaciones:

```
// Selecciona todos los elementos pares de una lista (":odd" selecciona los impares)
$("ul#menu li:even")
// Selecciona los 4 primeros párrafos de la página
$("p:lt(4) ")
// Selecciona todos los enlaces que contengan el texto " haga clic aquí"
$("a:contains(haga clic aquí)")
```

## Eventos

Además del evento `ready()` visto anteriormente, jQuery dispone de varias funciones relacionadas con la gestión de los eventos. El método `on()`, por ejemplo, permite asociar un número ilimitado de eventos a un mismo elemento:

```
// Al hacer clic sobre un párrafo de la página, se muestra un alert() con su contenido
$("p").on("click", function() {
    alert( $(this).text() );
});
```

El modelo de eventos de jQuery es muy completo, ya que cada evento dispone de dos posibilidades: si no se pasa ninguna función, se ejecuta el evento del elemento; si se pasa una función, se asocia esta función al evento del elemento:

```
// Ejecuta el evento "onclick" en todos los párrafos de la página
$("p").click();
// Asocia una función al evento "onclick" de todos los párrafos de la página
$("p").click(function() {
    alert( $(this).text() );
});
```

jQuery dispone de tantas funciones como eventos estándar de JavaScript. El nombre de cada función es el mismo que el del evento, sin el habitual prefijo "on": `focus()`, `blur()`, `keyup()`, `mouseover()`, `mouseup()`, `resize()`, `submit()`, etc.

Así mismo, jQuery añade un evento que no existe en JavaScript y que es realmente útil para muchas aplicaciones: `toggle()`. A este método se le pasan dos funciones, cuya ejecución se alterna en función de las veces que se hace clic sobre el elemento.

La primera vez que se hace clic sobre el elemento (y todas las veces impares), se ejecuta la primera función y la segunda vez que se pincha el elemento (y todas las veces pares) se ejecuta la segunda función:

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



### 3. Introducción a JSON

#### Introducción a JSON

- JSON (JavaScript Object Notation - Notación de objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.

15 - 9

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.



JSON (JavaScript Object Notation - Notación de objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, así como para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript (Standard ECMA-262 3rd Edition - Diciembre 1999).

JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocidos como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Ejemplo de sintaxis JSON:

```
{
  "menu":
  {
    "id": "file",
    "value": "File",
    "popup":
    {
      "menuitem":
      [
        { "value": "New" },
        { "value": "Open" },
        { "value": "Close" }
      ]
    }
  }
}
```

Características de JSON:

- Independiente de un lenguaje específico
- Basado en texto
- De formato ligero
- Fácil de parsear (por ejemplo convertir objetos y tipos de datos de .NET a JSON y viceversa, esto incluye cualquier otro tipo de datos que se manejan en otros lenguajes o Frameworks)
- No define funciones
- No tiene estructuras invisibles
- No tiene espacios de nombres (namespaces)
- No tiene validador
- No es extensible

## 4. AJAX con jQuery

### AJAX con jQuery

- Comando Ajax en jQuery:

```
$(function() {  
  $.ajax({  
    "url": "/myurl",  
    "type": "get",  
    "dataType": "json",  
    "success": function(data) {  
      console.log(data);  
    }  
  });  
});
```

Cuando algo es asíncrono, se ejecuta en segundo plano y no detiene el resto de su código en ejecución. Esto también se conoce como sin bloqueo. Algo básico como una llamada a `console.log` está bloqueando porque mientras se está ejecutando (aunque por una fracción del tiempo), nada más puede ejecutar hasta que esté terminado. AJAX implementa este comportamiento.

jQuery incorpora el método `$.ajax` que permite realizar llamadas asíncronas (requests) a un servidor web, teniendo la posibilidad de enviar y recibir datos en formato JSON. Estos datos pueden ser utilizados posteriormente para actualizar la UI. El formato de uso es el siguiente:

```
$(function() {  
  $.ajax({  
    "url": "/myurl",  
    "type": "get",  
    "dataType": "json",  
    "success": function(data) {  
      console.log(data);  
    },  
    "error": function (xhr, ajaxOptions, thrownError) {  
      alert(xhr.status);  
      alert(thrownError);  
    }  
  });  
});
```

El método AJAX de jQuery soporta múltiples argumentos. El argumento más importante que maneja es la URL, que puede ser una URL local (en el mismo dominio) o una URL externa (alojada en un dominio diferente).

La siguiente propiedad es **type**, el valor por defecto es GET, que es cuando se hace una solicitud y se adjuntan los datos de la URL. El otro valor que se puede usar es POST, que es cuando se requieren enviar varios valores con el propósito de modificar algún contenido en el servidor.

El siguiente es **dataType** que denota el tipo de datos que se devuelven. jQuery es capaz de adivinar con bastante precisión el tipo de dato, pero es mejor ser explícitos y establecer el tipo de dato.

Dentro de la configuración también puede especificar las funciones que se ejecutarán cuando se termine de ejecutar la llamada AJAX. Hay dos escenarios obvios que se brindan como opciones: **success** y **error**. Cada uno debe contener código con el que planeamos manejar ambos escenarios. Por ejemplo, si hago una llamada AJAX que me retorna un listado de precios que se manejaron para un producto en formato JSON puedo tomar esos datos y manipular la UI para mostrarlos como información complementaria al producto.

La devolución de llamada de error envía tres objetos. El primero es un objeto `jqXHR`. Este es el objeto jQuery que se envuelve alrededor de un `XMLHttpRequest`. El segundo es una cadena que contiene el estado, que suele ser "error", o "timeout" si la petición se tardó demasiado. Por último, está el argumento `errorThrown`, que es un objeto de error. En realidad, rara vez tiene que preocuparse por el primer o tercer argumento. A menudo, no es necesario preocuparse por los argumentos, porque la función de error se utiliza más para asegurarse de que la aplicación tiene alguna forma de manejar estos errores.