

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Introducción a jQuery © ADRINFOR S.L.

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Indice

Introducción a jQuery.	3
Introducción	3
Historia	4
¿Cuál es el uso principal de JavaScript?	5
Qué es jQuery	6
¿Por qué utilizar jQuery?	7
Qué necesito para programar con jQuery	8
Navegadores Compatibles	9
Cómo incluir jQuery en nuestra página	10
Alojando la librería en nuestro servidor	10
Utilizando la librería desde un alojamiento externo	12
¿Qué opción debo utilizar?	13
Ejercicios	14
Ejercicio 1: Instalar un editor de texto plano	14
Ejercicio 2: Entorno de desarrollo	15
Recursos	16
Enlaces de Interés	16

Introducción a jQuery.

Introducción

Bienvenido al curso de jQuery.

En este curso vamos a aprender a hacer uso de una de las librerías más útiles y populares en la programación de scripts del lado del cliente. El objetivo a conseguir es alcanzar el estatus de "Ninja" en la creación de scripts con esta librería, lo que nos hará capaces de poder manipular el comportamiento de nuestras páginas web, dotándolas de funcionalidades adicionales para las que el lenguaje HTML resulta insuficiente.

Este largo camino hasta el "conocimiento" nos enfrentará a retos mentales, que pondrán a prueba nuestro ingenio y poder de abstracción, dándonos la capacidad de cambiar la forma en la que, hasta ahora, veíamos una página web.



Ninja: veloz, eficaz, versátil.

La mayoría de internautas ven una página web como un entorno interactivo, que puede ofrecerles tanto un divertimento como una nueva forma de comunicarse, trabajar o gestionar su vida, además de ser una gran fuente de información y conocimiento.

Esta característica resulta realmente atractiva para estos usuarios, ya que ofrece un entorno "vivo", con contenidos que se actualizan sin necesidad de navegar "Adelante y Atrás", comportamiento que hasta hace relativamente poco teníamos interiorizado.

Toda esta interactividad se consigue por medio de fragmentos de programación (también conocidos como "snippets" o, más correctamente, "scripts") que son ejecutados por el navegador del "cliente" (o usuario, o internauta, etc.) bien automáticamente o a partir de interacciones del usuario - como hacer clic en un botón, o pulsar determinadas teclas del teclado. Así, intentaremos conocer una librería con la que podamos dotar de interactividad a los elementos que conforman la vista de una página web.

Historia

Como acabamos de comentar, la tecnología web actual nos brinda un entorno en el que ya no solamente se utiliza Internet como fuente informativa - hemos dicho "solamente" ya que sigue siendo una gran fuente informativa- sino que además ofrece un sinfín de recursos y aplicaciones con las que interactuar (servicios de streaming de vídeo o música online, redes sociales, webmails, paneles de control de nuestros productos bancarios o de servicios de las teleoperadoras, etc.).

Retornando a la historia de jQuery, vamos a comenzar diciendo que los orígenes de la librería nos acercan a JavaScript. JavaScript es un lenguaje de programación procedente del estándar ECMAScript. El código generado en este lenguaje es ejecutado desde el lado del cliente (navegador web), lo que quiere decir que, pese a que puede realizar interacciones con datos remotos (servidor), los fragmentos de código (también conocidos como scripts) escritos en este lenguaje son ejecutados en el ordenador del internauta que accedió a la página.

Como la mayoría de lenguajes de script, el código escrito en JavaScript no genera un fichero ejecutable. En cambio, necesita de un programa que se encargue de interpretarlo. Anteriormente hemos comentado que los scripts JavaScript son ejecutados en el ordenador del cliente, y, a su vez, que este código necesita un programa que lo interprete. Pero... ¿Cuál es el programa encargado de interpretarlo? EL NAVEGADOR.

Cuando un usuario del World Wide Web accede a una página web, el navegador es el encargado de obtener todos los recursos que la página visitada solicita cargar. Estos recursos pueden ser imágenes, elementos multimedia, hojas de estilo y documentos JavaScript. Además, los navegadores también son capaces de interpretar los fragmentos de código JavaScript incrustados en la propia página.

Puede parecer que esta ejecución no controlada de pequeños "programas" lleve a hacer de Internet un entorno inseguro para el usuario, ya que en su propio ordenador se ejecutan fragmentos de programas que él no ha solicitado expresamente.



Sin embargo y pese a que el usuario inexperto pueda tener esta inseguridad, la ejecución de código JavaScript en nuestros navegadores es completamente seguro. Esto se debe a que, los intérpretes de JavaScript integrados en los navegadores no permitan acceder a recursos propios del dispositivo (ordenador, tableta, etc.) que estemos usando para acceder a la página, ni almacenar ni ejecutar procesos críticos para el mismo.

Durante mucho tiempo, JavaScript ha tenido la reputación de lenguaje torpe e inadecuado para el desarrollo serio. Esta reputación se ha debido, en gran parte, a las implementaciones incompatibles del lenguaje y del DOM en varios navegadores que ocasionaban molestos errores en su ejecución. Estos errores causaban gran frustración tanto a los visitantes como a los propios programadores que intentaban corregirlos.

Con el tiempo y la evolución de los navegadores, JavaScript ha conseguido evolucionar, y, hoy en día, su uso puede ofrecer gran versatilidad en el diseño web. En cambio, las diferencias entre los intérpretes de los navegadores han seguido manteniendo sus incompatibilidades tanto en la interpretación de JavaScript como en el renderizado de los estilos CSS que cada uno de éstos soportan.

En Febrero de 2006, John Resig presentó la primera versión de jQuery. Esta librería sorprendió gratamente a la comunidad de desarrolladores web puesto que simplificaba en gran medida la programación de código JavaScript para interactuar, tanto con los elementos del DOM como para gestionar los diferentes eventos que se producen en la visita de una página web. Desde su publicación, la librería ha evolucionado solucionando bugs, optimizando su código y ampliando las funcionalidades ofrecidas.

Google haciendo uso de ella... Microsoft y Nokia incluyéndola en sus plataformas... Millones de páginas utilizándola y miles de librerías desarrolladas con la librería son algunos de los argumentos que dan cuenta de la fama que jQuery ha obtenido a lo largo de sus años de historia.

¿Cuál es el uso principal de JavaScript?

Hasta hace muy poco tiempo, el uso de JavaScript estaba limitado a tareas simples y no críticas (validación de formularios y uso meramente ornamentístico). Sin embargo, actualmente es utilizando para escribir código complejo que a menudo es responsable de buena parte de la funcionalidad de un sitio.

Mediante JavaScript podremos interactuar con el DOM (Document Object Model) de cualquier página, consiguiendo un amplio abanico de interacciones con la misma.



Anotación: El Document Object Model o DOM

Definición de DOM - Fuente: Wikipedia

El Document Object Model o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

La evolución del mundo web, empezando por los navegadores y continuando por los estándares HTML y CSS, además de la aparición de librerías que han facilitado la programación en JavaScript, tales como Prototype, MooTools o jQuery, han facilitado enormemente la creación de código organizado y escalable en JavaScript.

JavaScript permite la interactividad... el dinamismo en una página... y, desde la aparición de la tecnología AJAX, la cual permitía hacer llamadas asíncronas con el servidor web que aloja el sitio, la obtención de datos del propio servidor cuando la página ya se encuentra cargada en el navegador.

Qué es jQuery

Vale, muy bien la lección de historia. Pero... ¿qué es jQuery?

jQuery es una librería escrita en JavaScript que nos permite minimizar el código en nuestras páginas y simplificar los scripts de interacción y manipulación de los elementos del DOM de una página web. Además, ofrecer un entorno no intrusivo y estándar para todos los navegadores actuales, minimizando el impacto que tienen las diferencias de interpretación de los estándares por los navegadores.

La principal característica de jQuery es la sencillez combinada con el gran potencial que ofrece. Unas pocas líneas de código haciendo uso de la librería puede modificar por completo el funcionamiento de una página. Además, jQuery aprovecha la sintaxis utilizada en los selectores de CSS, lo que hace que los desarrolladores y diseñadores que están familiarizados con éstos puedan reutilizar éstos conocimientos.

jQuery es software libre y de código abierto (Licencia MIT y la Licencia Pública General de GNU v2) lo cual hace que sea una librería que podremos utilizar tanto en proyectos libres como privativos.

Con jQuery dispondremos de una amplia variedad 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.

A lo largo del curso podrás ver que con jQuery ("The Write less, do more library") conseguirás grandes resultados en menos tiempo y con muchas menos líneas de código.

¿Por qué utilizar jQuery?

No vamos a debatir si debemos utilizar o no jQuery. Si estás leyendo estas líneas, seguramente no me costará convencerte de utilizar esta librería. Pero, por si acaso, vamos a comentar algunos de los motivos por los que sí podríamos considerar jQuery como nuestro aliado a la hora de escribir nuestro código.



¿Qué características de jQuery la hacen una librería a tener en cuenta?

1º

Es de código abierto - bajo las licencias MIT y GNU GPL

2º

Es de reducido espacio - por lo que no aumentará el tiempo de descarga de nuestra página

3º

Su popularidad va en aumento, con una gran comunidad de usuarios y desarrolladores creando plugins y recursos realizados con la librería

4º

Normaliza las diferencias entre los buscadores, permitiendo desarrollar código válido para todos ellos

5º

Dispone de una API completamente documentada, con códigos de ejemplo (disponible en inglés en <http://docs.jquery.com/>)

6º

Provée opciones con las que evitar conflictos con otras librerías JavaScript

7º

Su desarrollo es continuo, con constantes mejoras y solución de errores

8º

Incorpora las especificaciones estándar del W3C - soporta una amplia mayoría de selectores CSS3

9º

Ofrece grandes funcionalidades a diseñadores y programadores por igual

Símplemente, es una librería que te enamorará ;) .



Recuerda: jQuery es Javascript

No olvides que jQuery es un framework escrito en Javascript. Las diferencias en la sintáxis y el acceso a los elementos del DOM pueden confundir y hacer pensar que el lenguaje es diferente. Todas las limitaciones que los navegadores aplican sobre JavaScript también se aplicará sobre jQuery.

Qué necesito para programar con jQuery

Para programar nuestros scripts no es necesario utilizar un programa específico. El código escrito en JavaScript y, por siguiente, con jQuery, está contenido en archivos de texto plano. Esto supone que un simple editor de textos, como el Bloc de Notas integrado en Windows, puede ser suficiente para crear nuestros archivos.

Sin embargo, contar con un editor "enriquecido", que nos ofrezca alguna ayuda adicional a la hora de programar suele ser una mejor elección.

Si eres usuario de algún editor avanzado como Dreamweaver, FrontPage o Kompozer puedes utilizar este programa para la edición de tus scripts. Si por el contrario no dispones de ningún editor de páginas web, puedes hacer uso de un editor de texto más sencillo, como Notepad.



Notepad es un editor para entornos Windows gratuito. Soporta varios lenguajes de programación, y es completamente funcional para la edición de archivos Javascript.

Está disponible en varios formatos de descarga, con instalador y sin instalador. Es recomendable utilizar el primero de ellos, pero si lo deseas, también puedes utilizar la versión sin instalador, simplemente descomprimiendo el archivo de descarga y ejecutando directamente el programa.




Puedes descargar el editor Notepad desde el siguiente enlace: <http://notepad-plus-plus.org/download/>

Para facilitar el trabajo, también necesitaremos un navegador web sobre el que podamos utilizar un "depurador" para poder analizar los elementos del DOM en cualquier momento. La mayoría de navegadores actuales soporta esta funcionalidad y, en caso de no disponerlos, existen generalmente complementos para los navegadores, que ofrecen este valor añadido al ofrecido por el propio navegador.

La elección del navegador es, generalmente, una decisión muy personal. Internet Explorer, Firefox, Chrome, Opera... El listado de ellos es muy amplio y variado.

Como guía, vamos a indicar cómo disponer de un entorno de depuración en tres de ellos:

- Internet Explorer. Las últimas versiones del navegador incluyen una herramienta integrada de depuración. Podremos ejecutar esta herramienta pulsando la tecla "F12" o desde el menú del navegador, accediendo a Herramientas -> Herramientas de desarrollo. Este entorno es algo más sencillo que el del resto de navegadores, pero ofrece un entorno funcional para la depuración de nuestras páginas con éste editor.
- Mozilla Firefox: Al igual que ocurre con Internet Explorer, éste navegador ha incluido una herramienta de depuración integrada en el navegador. Sin embargo, y dado que todas las versiones de Firefox no incluyen esta característica, es recomendable utilizar el complemento FireBug, ya que sus características mejoran al que integra el programa. Una vez instalado el complemento, pulsaremos la tecla "F12" cada vez que queramos abrir el entorno de depuración. Firebug dispone de varios paneles desde los que podremos ver la consola del navegador, inspeccionar el DOM de la página que tengamos cargada y las peticiones hechas a servidores para obtención de recursos. Una característica muy interesante de Firebug es que podremos explorar mediante el ratón, pulsando el botón  y deslizando el cursor sobre las diferentes secciones de una página.
- Google Chrome: Una vez más, el navegador desarrollado por Google ha integrado esta funcionalidad en el navegador. Para abrir la herramienta de depuración, pulsaremos la tecla "F12" del teclado. En la parte inferior de la pantalla, aparecerá la herramienta de depuración, desde la que podremos explorar los elementos del DOM correspondientes a la página cargada, así como examinar el tráfico de red que realiza el navegador y observar la consola, que nos indicará si se ha producido algún error en el código.

Navegadores Compatibles

Una de las ventajas de utilizar jQuery es que conseguimos librarnos de las características específicas de cada navegador. El código que escribamos se ejecutará correctamente en todos los navegadores, permitiéndonos disponer de un entorno en el que las diferencias en la interpretación del DOM no nos provoquen dolores de cabeza innecesarios.



jQuery es un framework no intrusivo. Abstrae las características peculiares de cada navegador, simplificando nuestros fragmentos de código y consiguiendo que funcione en todos los navegadores.

Esta cualidad de jQuery supone un significativo ahorro de trabajo para el desarrollador, ya que no tendrá que escribir el mismo código adaptado a las peculiaridades de sintaxis de cada intérprete, sino que se utilizarán métodos definidos por la librería, funcionales en todos los navegadores y plataformas.

Oficialmente, y según la documentación de la librería, jQuery es compatible con las siguientes versiones de los principales navegadores:

- Firefox: Todas las versiones (aunque es posible encontrar algún bug con las primeras versiones - 1.0.x - del navegador)
- Internet Explorer 6
- Safari
- Opera
- Chrome

Esta compatibilidad y los diferentes motores de los principales navegadores supone que prácticamente la totalidad de navegadores web actuales soportarán las funcionalidades ofrecidas por la librería.

A B C

jQuery: Framework no intrusivo

Cómo incluir jQuery en nuestra página

Alojando la librería en nuestro servidor

Desde la página oficial de jQuery, podemos descargar en cualquier momento la última versión de la librería. Ésta se encuentra disponible en dos formatos: la versión para desarrollo y la versión de producción.

Estas dos distribuciones son en realidad la misma librería y ofrecerá idéntica funcionalidad. La única diferencia entre ambas es que, mientras que en la primera el código se encuentra correctamente estructurado para facilitar su lectura, en la segunda, el código se encuentra comprimido, haciendo prácticamente imposible su lectura y consiguiendo una reducción significativa del tamaño (en Kb) de la librería. Esta segunda versión es recomendable para su uso en entornos de producción (abiertos al público) ya que se conseguirá que la página sea descargada más rápidamente.

Para poder incluir jQuery en nuestro sitio, guardaremos una copia de la librería (el archivo .js) dentro de la carpeta donde vayamos a guardar los archivos html sobre los que vayamos a operar.



Es importante mantener una estructura de contenidos bien organizada. Distribuir los diferentes tipos de archivos (css, js y imágenes) en subcarpetas, es una práctica, además de habitual, muy recomendable.

Una estructura correctamente organizada mostrará un aspecto similar al siguiente:

- / (Raíz) - en el directorio raíz se encontrarán los archivos html
- _ js - carpeta que contendrá los archivos javascript
- _ css - carpeta que contrandrá las definiciones de estilo
- _ img - carpeta donde se almacenarán las imágenes utilizadas por la página

Una vez disponemos de la librería en el directorio correspondiente, crearemos un nuevo documento de texto, llamado "documento1.html".

Este documento, tendrá el siguiente código:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Documento sin título</title>
  </head>
  <body>
  </body>
</html>
```

Para incluir jQuery en esta página, escribiremos el código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Documento sin título</title>
    <script type="text/javascript" src="js/jquery-1.8.2.min.js"></script>
  </head>
  <body>
  </body>
</html>
```

Con ello, la página incluirá el archivo "jquery-1.8.2.min.js" disponible dentro de la carpeta "js". Hecho esto, ya dispondremos de la base para todas las páginas que vayamos a crear con jQuery.



La etiqueta script permite incluir cualquier archivo javascript a una página. Para cargar jQuery haremos uso de esta etiqueta, asignando en el valor src de la etiqueta la ruta en la que se encuentra el archivo js.

Utilizando la librería desde un alojamiento externo

La mayoría de los desarrolladores utilizan en sus páginas una copia de la librería alojada en su propio servidor. Otra alternativa es incluir la librería por medio de un enlace a una copia de la misma que se encuentre disponible en un servidor externo. Este sistema de funcionamiento es posible gracias a los CDN. Uno de los CDN más utilizados es el ofrecido por Google, que entre otras ventajas, aporta la seguridad de que el recurso se encontrará siempre disponible.



¿Qué es un CDN?

Se conoce como CDN (Content Delivery Network) a una red de ordenadores cuya funcionalidad es servir contenidos de forma escalable y rápida. Estos servidores pueden encontrarse distribuidos geográficamente, siendo el servidor más próximo al cliente el encargado de dar respuesta a las peticiones de un recurso.

Hacer uso de los CDN de Google nos ofrece dos ventajas que han de tenerse en cuenta a la hora de decidir si vamos a hacer uso de este servicio del gigante de internet:

- Velocidad en la descarga de la librería, y liberación de peticiones a nuestro servidor.
- Un gran número de usuarios dispondrá de una copia cacheada en su navegador

A su vez, hay que tener en cuenta que no podremos hacer uso de esta copia de la librería si vamos a necesitarla en un entorno sin conexión a internet, como puede ser un entorno de desarrollo en el que, puntualmente, podamos no disponer de acceso al recurso por la red.

¿Qué versiones de la librería podemos encontrar en el CDN de Google?

El CDN de Google da soporte a las siguientes versiones de jQuery: 1.8.2, 1.8.1, 1.8.0, 1.7.2, 1.7.1, 1.7.0, 1.6.4, 1.6.3, 1.6.2, 1.6.1, 1.6.0, 1.5.2, 1.5.1, 1.5.0, 1.4.4, 1.4.3, 1.4.2, 1.4.1, 1.4.0, 1.3.2, 1.3.1, 1.3.0, 1.2.6, 1.2.3



Entre las versiones disponibles, Google no ha incluido la 1.2.5 ni la 1.2.4, dado su corto periodo de "vida", y la falta de estabilidad de estas versiones de la librería.

¿Cómo incluyo una versión de la librería desde el alojamiento de Google?

Para incluir la librería desde el alojamiento de Google deberemos indicar en la ruta (atributo src) del script la URL correspondiente a la librería, de la siguiente forma:

`http://ajax.googleapis.com/ajax/libs/jquery/ 1.8.2/`**`jquery.min.js`**

DOMINIO Y RUTA DE ALOJAMIENTO

VERSIÓN

FORMATO - jquery.min.js o jquery.js

Así, una página básica que haga uso de la librería desde el repositorio de Google tendrá este aspecto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Documento sin título</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    </head>
    <body>
    </body>
  </html>
```

Al igual que en el proceso de descarga desde la página de jQuery, las distribuciones de la librería disponibles en el repositorio de Google se encuentran disponibles en los formatos comprimido y sin comprimir.

¿Qué opción debo utilizar?

En general, es recomendable utilizar la librería incluyéndola desde un alojamiento externo, de confianza. De este modo, delegaremos la carga de la librería en un recurso externo, lo que reducirá las solicitudes a nuestro servidor y permitirá una carga de la misma más ágil.

No obstante, si nuestro proyecto está orientado al uso off-line (sin conexión permanente a internet) entonces el escenario cambia radicalmente. Si fuese esta la situación, deberíamos optar por la descarga del archivo js de la librería.

Ejercicios

Ejercicio 1: Instalar un editor de texto plano

Duración estimada del ejercicio



10
minutos

Instala un editor de texto plano. Puedes elegir entre los siguientes:

Entorno Windows



Notepad : <http://notepad-plus-plus.org/>



EditPlus: <http://www.editplus.com/>

Entorno Linux



gedit: <http://projects.gnome.org/gedit/screenshots.html>



kate: <http://kate-editor.org/>

Entorno Mac



Komodo Edit: <http://www.activestate.com/komodo-edit>



TextWrangler: <http://itunes.apple.com/es/app/textwrangler/id404010395?mt=12>

Ejercicio 2: Entorno de desarrollo

Duración estimada del ejercicio



15
minutos

El punto de partida para utilizar jQuery es incluir la librería en una página HTML.

Crea un directorio en tu equipo en el que contener los documentos HTML en los que haremos uso de la librería. Añade también un subdirectorio "js" y descarga en él la librería jQuery.

Una vez definido el entorno de desarrollo, vamos a cargar la librería en dos documentos HTML diferentes.

En el primero, llamado "documento1.html" cargaremos la librería desde la carpeta "js".

En el segundo archivo, llamado "documento2.html" haz uso de una de las copias disponibles de la librería en el CDN de Google.

Recursos

Enlaces de Interés



<http://www.jquery4u.com/utilities/live-jquery-debugging-firebug/>

Guía de uso e instalación de Firebug para la depuración de código Javascript y jQuery (Inglés)