



Eusko Jaurlaritzaren Informatika Elkarte
Sociedad Informática del Gobierno Vasco

Firebug

Manual de usuario

Fecha: 24/07/2008

Referencia:

EJIE S.A.
Mediterráneo, 3
Tel. 945 01 73 00*
Fax. 945 01 73 01
01010 Vitoria-Gasteiz
Posta-kutxatila / Apartado: 809
01080 Vitoria-Gasteiz
www.ejie.es

Control de documentación

Título de documento: Firebug. Manual de Usuario

Histórico de versiones

Código:

Versión: 1.0

Fecha: 24/07/2008

Resumen de cambios:

Cambios producidos desde la última versión

Primera versión.

Control de difusión

Responsable: Ander Martínez

Aprobado por: Ander Martínez

Firma:

Fecha:

Distribución:

Referencias de archivo

Autor: Consultoría de áreas de conocimiento

Nombre archivo: Firebug. Manual de instalación v1.0.doc

Localización:

Contenido

Capítulo/sección	Página	
1	Introducción	5
2	Conceptos básicos	5
3	Integración con Firefox	5
3.1	Ventana de Firebug	7
4	Utilidad Práctica	8
4.1	Consola	8
4.1.1.	Monitorización de peticiones AJAX	9
4.1.2.	Depuración Javascript	10
4.2	HTML	12
4.2.1.	Visualización de código HTML	12
4.2.2.	Búsqueda de código HTML	16
4.2.3.	Edición de código HTML	18
4.2.4.	Modificación de las dimensiones de la caja de un elemento	21
4.3	CSS	22
4.3.1.	Modificación de propiedades de un estilo	22
4.3.2.	Modificación de estilos	23
4.3.3.	Deshabilitación de estilos	23
4.4	Script	23
4.4.1.	Puntos de ruptura	23
4.4.2.	Ejecución paso a paso	26
4.4.3.	Inspección de expresiones	27
4.4.4.	Gestión de scripts	28
4.5	DOM	29
4.5.1.	Visualización de la información	30
4.5.2.	Modificación de propiedades	31
4.6	NET	33

4.6.1. Filtrado por tipo de fichero	34
4.6.2. Control de las peticiones HTTP	34

1 Introducción

En este manual se describen los distintos aspectos que debe conocer el usuario sobre el componente para Firefox, **Firebug**.

2 Conceptos básicos

Firebug es una extensión (add-on) de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea y en tiempo de visualización, sin necesidad de modificar fuentes y recargar de nuevo la página.



Para obtener información adicional sobre el producto se puede acceder a su página Web:

<http://www.getfirebug.com/>

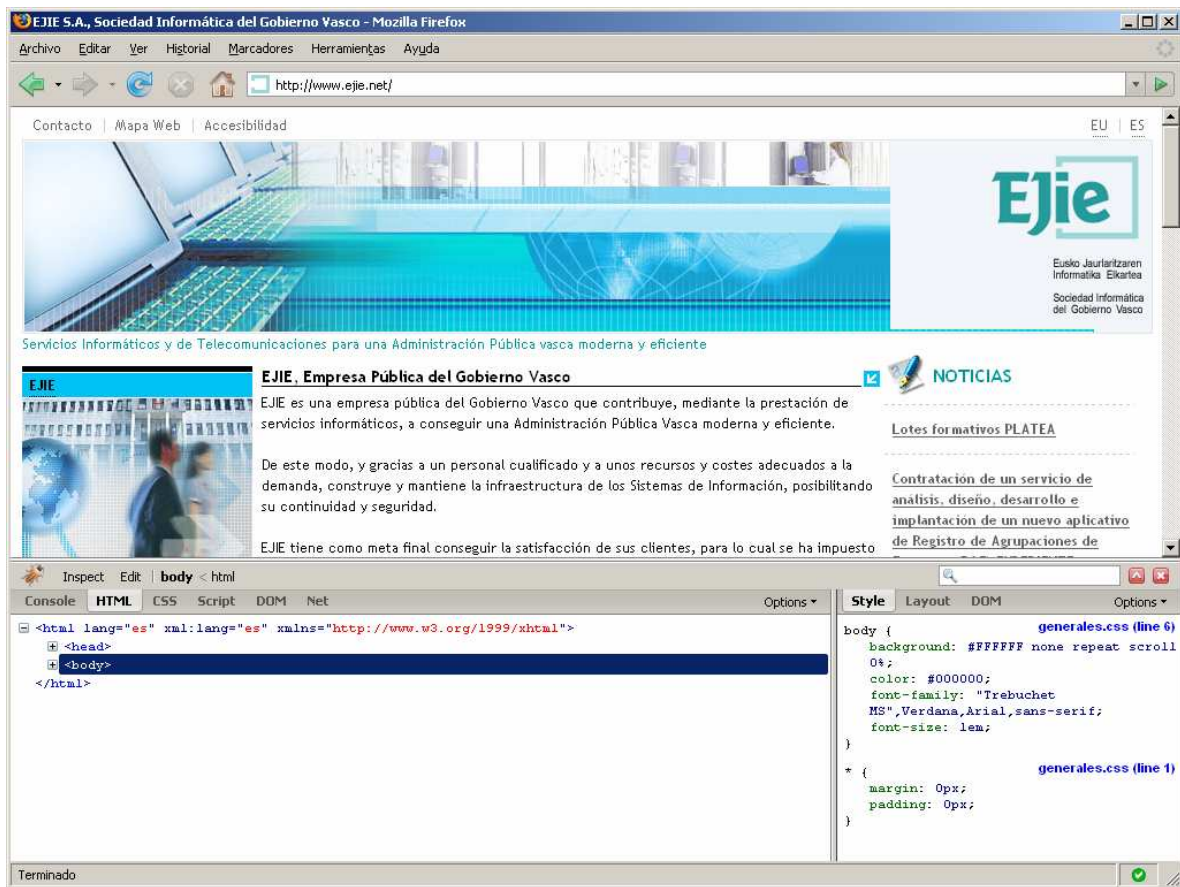
En el momento de escribir este manual la última versión para Firefox 2 es la 1.0.5, y para Firefox 3, la 1.2 (en estado beta):

<https://addons.mozilla.org/es-ES/firefox/addons/versions/1843>

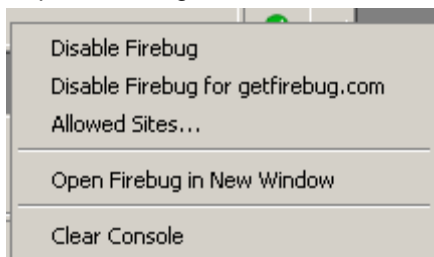
3 Integración con Firefox

Cuando se añade la extensión Firebug a Firefox, en la barra de estado inferior del navegador aparece, a la derecha, el símbolo:  que indica que Firebug está activo. Mientras Firebug está activo, se ralentiza un poco la carga de la página que se consulta, ya que se tiene que analizar el contenido de la página. Por ello, Firebug da la opción de deshabilitarlo o incluso de definir en que sitios web estará activo, si su opción por defecto es deshabilitado o en que sitios web estará deshabilitado si su opción por defecto es activo. Cuando Firebug está deshabilitado se muestra con el siguiente icono: .

Si se pincha el icono de Firebug con el ratón, la zona de navegación se hace más pequeña, y en la parte inferior que ocupaba ésta, aparece ahora un panel donde interaccionar con la funcionalidad que ofrece la extensión y que se explicará a fondo en este manual.



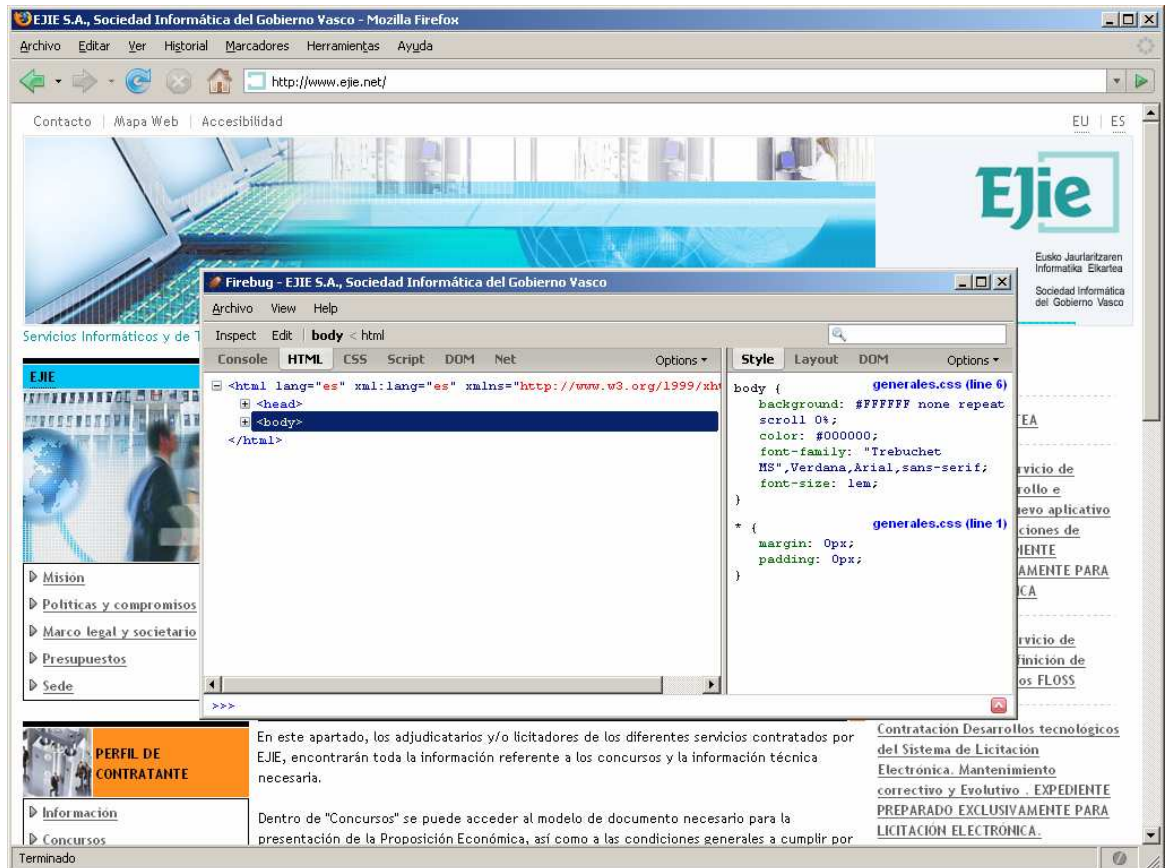
Pinchando con el botón derecho del ratón sobre el icono de Firebug que está en la barra inferior de Firefox, aparece el siguiente menú contextual:



Las acciones que se pueden realizar son:

- Deshabilitar Firebug (Disable Firebug): con esta opción Firebug deja de analizar los contenidos de las páginas visitadas.
- Deshabilitar Firebug for...: con esta opción Firebug deja de analizar los contenidos del sitio web que se está visitando en ese momento.
- Configuración de sitios web (Allowed Sites...): con esta opción se permite configurar para qué sitios web se activará Firebug o para cuales se bloqueará su acción, es decir, se deshabilitará; independientemente de la opción por defecto que esté activa.

- Abrir Firebug en una nueva ventana (Open Firebug in New Window): Como se ha explicado anteriormente, Firebug, se despliega ocupando parte de la ventana de navegación, con esta opción, la zona de navegación no se modifica y se lanza el panel de Firebug en una ventana a parte.

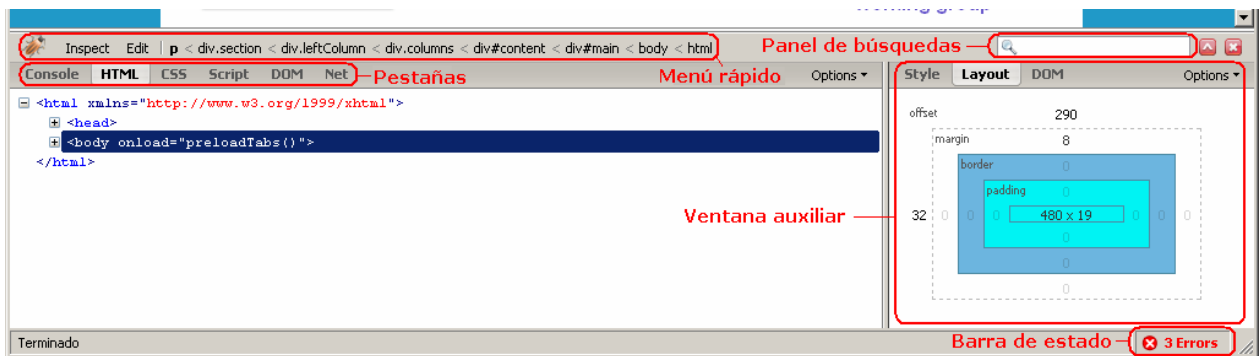


- Borrar consola (Clear Console): esta opción elimina el contenido de la consola de Firebug.

3.1 Ventana de Firebug

Como se ha comentado anteriormente la ventana de Firebug se despliega en la parte inferior del navegador, reduciendo la zona de navegación.

A continuación se muestra una figura con los elementos de la ventana de Firebug:



- Pestañas: Cada pestaña atañe a una parte de la funcionalidad que ofrece Firebug. Cada una presenta su propia información.
- Menú rápido: Cambia para cada pestaña seleccionada, para ampliar las opciones de cada funcionalidad.
- Ventana auxiliar: Aparece en determinadas funcionalidades para ampliar la información que se presenta.
- Panel de búsquedas: Muy útil para buscar palabras en el código de la página.
- Barra de estado: Da información de los errores que ocurren en la carga de la página.

4 Utilidad Práctica

En este punto se comentará la funcionalidad que ofrece Firebug, dividiéndola en base a las pestañas que aparecen en la ventana de la extensión.

- Pestaña Console: Muy útil en el desarrollo javascript de la página. Permite ejecutar comandos javascript. Además muestra las peticiones AJAX de una página.
- Pestaña HTML: Permite ver el HTML generado de una página web y lo más interesante: modificarlo y visualizar los cambios sin necesidad de cargar la página de nuevo. Además da facilidades para encontrar el código correspondiente a un elemento de la página.
- Pestaña CSS: Permite la edición de los estilos de la página de una forma sencilla e intuitiva. Los cambios en la página se muestran “en tiempo real”, es decir, inmediatamente después de ser realizados, sin necesidad de grabar y recargar la página.
- Pestaña Script: Muy útil para la depuración de código javascript, permite ver los scripts de la página, además de poner puntos de ruptura e incluso, puntos de ruptura condicionales.
- Pestaña DOM: Permite ver la información de los objetos DOM cargados por la página, además de su modificación de forma dinámica.
- Pestaña NET: Muestra el tiempo de carga de los diferentes elementos de la página. Muy útil para comprobar las prestaciones de la página.

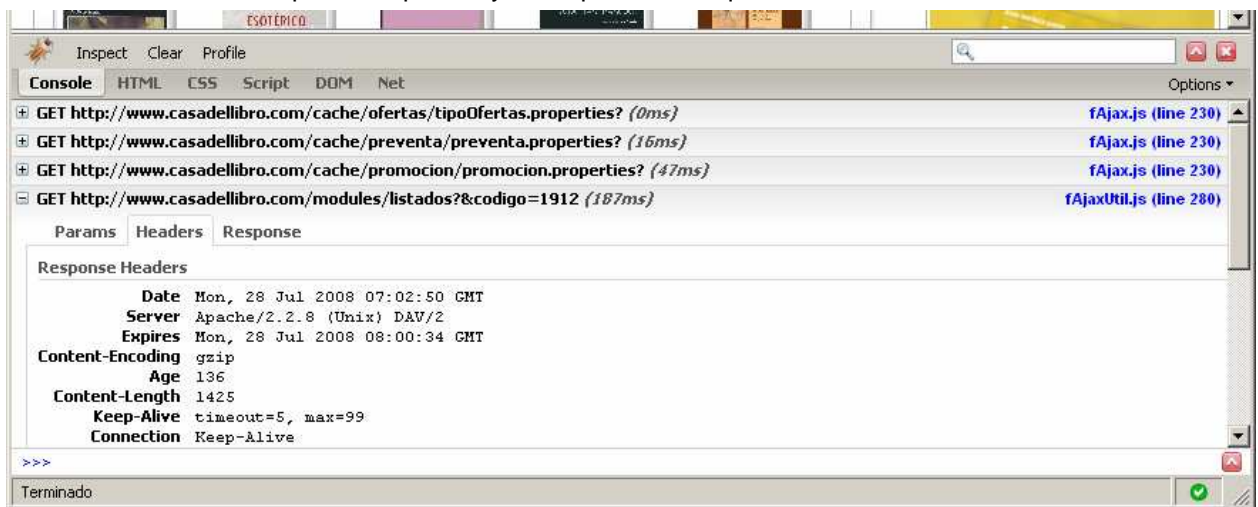
4.1 Consola

La pestaña con la consola, es una de las más útiles para el desarrollo Javascript y AJAX de una aplicación web. Las funcionalidades que ofrece la pestaña son, entre otras:

- Monitorización de peticiones AJAX.
- Depuración de código Javascript.

4.1.1. Monitorización de peticiones AJAX

Cuando cargamos en Firefox con Firebug activado, una página que tiene peticiones AJAX, en la pestaña 'Console', aparecen una línea con el resumen de cada petición, indicando el protocolo utilizado: GET, POST o HEAD, la URL llamada, el tiempo de respuesta y el script desde el que se realiza la llamada.

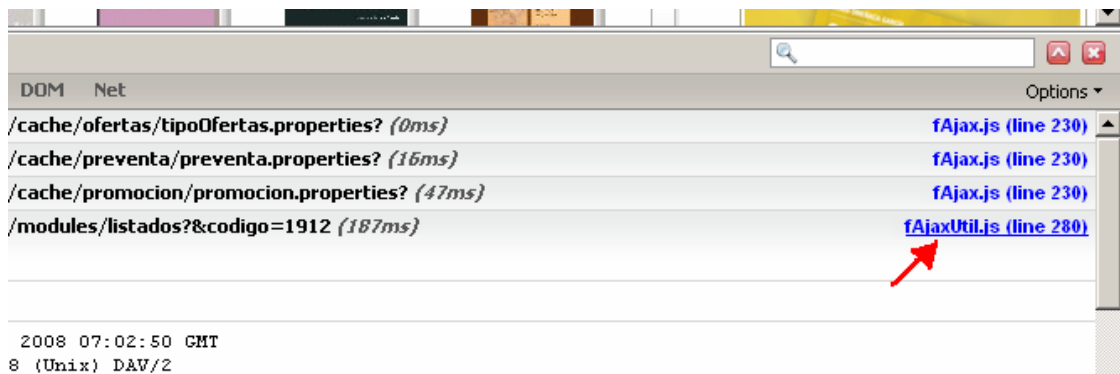


Si se despliega la línea de la petición, pinchando en el símbolo '+' o en la URL de la petición, aparece información detallada de la misma.

- Params: Los parámetros con los que fue invocada. No aparece información si no se han enviado parámetros.
- Headers: Las cabeceras, tanto de la petición como de la respuesta.
- Response: El contenido completo de la respuesta.

Esta información es particularmente útil cuando se invoca a servicios web que no ha desarrollado uno mismo y de los que no se tiene mucha información.

Si se pincha en la información del script que realiza la petición AJAX, Firebug presentará en la pestaña 'Script' la línea del fichero que realizó la invocación para su depuración.




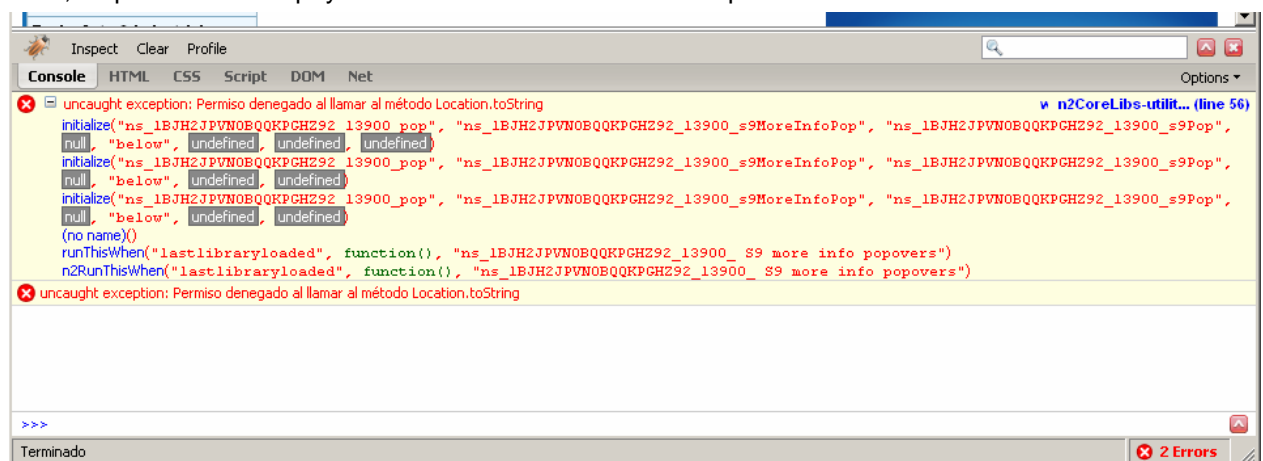
4.1.2. Depuración Javascript

Uno de los puntos fuertes de Firebug, es su uso para la depuración de código Javascript. Podemos dividir la funcionalidad que ofrece la pestaña 'Console' respecto a este punto en: visualización de errores Javascript, ejecución de código fuera de los scripts de la página, inserción de trazas en el código de la página y medición de las prestaciones del código ejecutado.

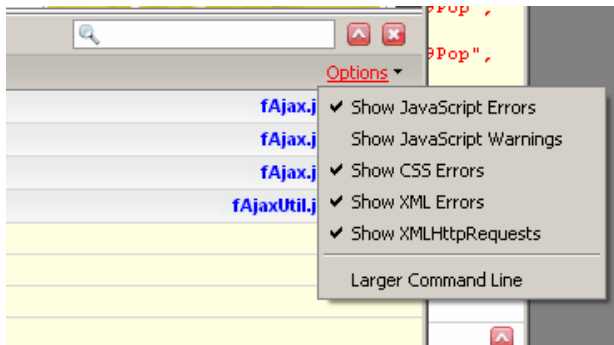
Visualización de errores Javascript

Firebug muestra los errores que ha habido en la carga de una página. Cabe resaltar, que muestra los errores de la página que se está visitando en cada una de las ventanas del navegador, a diferencia de otros depuradores Javascript, que vuelcan los errores de todas las páginas visitadas, en la misma presentación.

En la barra de estado se muestra el número de errores que tiene cada página: , y en la zona de la consola se muestra un resumen de los errores. Además del texto del error, si se despliega la línea con el error, se presenta el script y el número de línea donde se ha producido:



En la opción de menú 'Options' que aparece a la derecha de la barra de pestañas, se puede indicar que tipo de errores se desea que muestre Firebug:

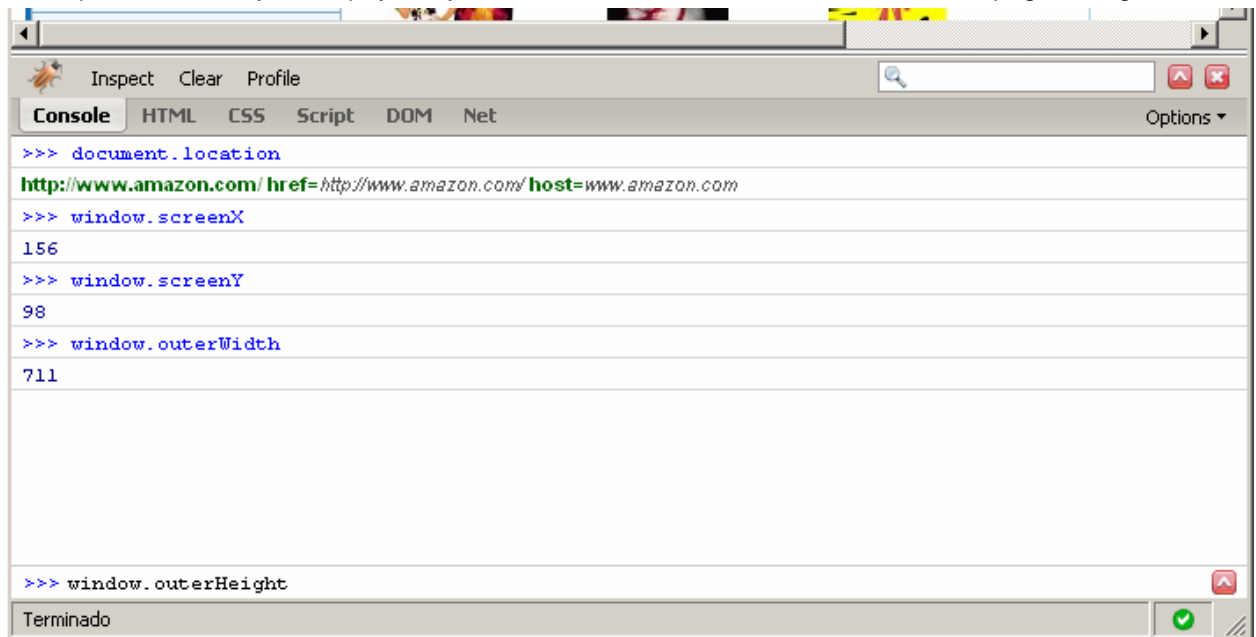


Los cambios serán efectivos en la siguiente carga de una página.

Ejecución de código Javascript

Algunas veces, en el desarrollo de una página se desea ejecutar código únicamente por comprobar que se tiene la sintaxis correcta. Firebug, ofrece en la Pestaña 'Console' esta posibilidad.

En la parte inferior de la zona de la consola, aparece una línea precedida de: '>>>', en la que se puede escribir cualquier sentencia javascript y es ejecutada inmediatamente en el contexto de la página cargada:



Esto es muy útil para hacer pruebas, inspeccionar y modificar el contenido de un objeto y lanzar eventos desde la línea de comandos.

Inserción de trazas

El punto más innovador de la funcionalidad de consola, tiene que ver con la variable global 'console' que Firebug añade a todas las páginas cargadas en Firefox, y al API que ofrece dicho objeto.

La variable console dispone de varios métodos para escribir información en la Consola de Firebug, que se pueden utilizar en el código para dejar trazas, evitando así el molesto uso de los 'alert' comúnmente utilizados

para ese fin.

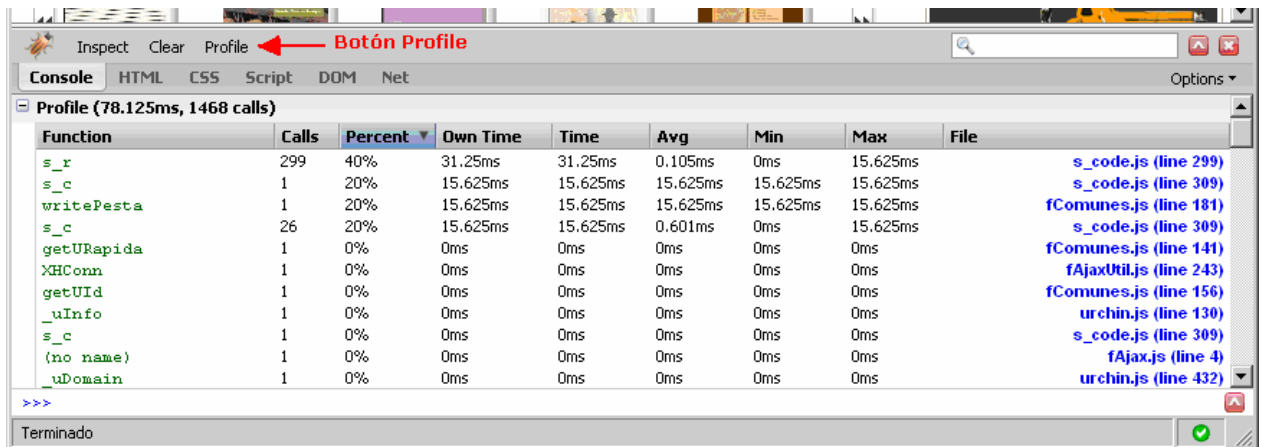
Una completa información sobre el API Console que ofrece Firebug puede encontrarse en:

<http://getfirebug.com/console.html>

Medición de las prestaciones

En el menú rápido de la pestaña 'Console', aparece el botón 'Profile'. Su funcionamiento es el siguiente:

1. En el momento que se pulsa el botón se lanza un medidor de tiempos de respuesta de las funciones javascript que se van ejecutando.
2. A continuación se realizará la acción o acciones que se quiere medir, por ejemplo pasar el ratón por los diferentes elementos de una página o cargar una página concreta y se volverá a pulsar el botón profile.
3. Deberá aparecer en la consola la información de rendimiento de las funciones ejecutadas, ver imagen:



The screenshot shows the Firebug Profile tool interface. At the top, there are tabs for 'Inspect', 'Clear', and 'Profile', with a red arrow pointing to the 'Profile' tab labeled 'Botón Profile'. Below the tabs is a table titled 'Profile (78.125ms, 1468 calls)'. The table has columns for 'Function', 'Calls', 'Percent', 'Own Time', 'Time', 'Avg', 'Min', 'Max', and 'File'. The data in the table is as follows:

Function	Calls	Percent	Own Time	Time	Avg	Min	Max	File
s_r	299	40%	31.25ms	31.25ms	0.105ms	0ms	15.625ms	s_code.js (line 299)
s_c	1	20%	15.625ms	15.625ms	15.625ms	15.625ms	15.625ms	s_code.js (line 309)
writePesta	1	20%	15.625ms	15.625ms	15.625ms	15.625ms	15.625ms	fComunes.js (line 181)
s_c	26	20%	15.625ms	15.625ms	0.601ms	0ms	15.625ms	s_code.js (line 309)
getURapida	1	0%	0ms	0ms	0ms	0ms	0ms	fComunes.js (line 141)
XHConn	1	0%	0ms	0ms	0ms	0ms	0ms	fAjaxUtil.js (line 243)
getUId	1	0%	0ms	0ms	0ms	0ms	0ms	fComunes.js (line 156)
_uInfo	1	0%	0ms	0ms	0ms	0ms	0ms	urchin.js (line 130)
s_c	1	0%	0ms	0ms	0ms	0ms	0ms	s_code.js (line 309)
(no name)	1	0%	0ms	0ms	0ms	0ms	0ms	fAjax.js (line 4)
_uDomain	1	0%	0ms	0ms	0ms	0ms	0ms	urchin.js (line 432)

At the bottom of the window, there is a status bar that says 'Terminado' with a green checkmark icon.

4.2 HTML

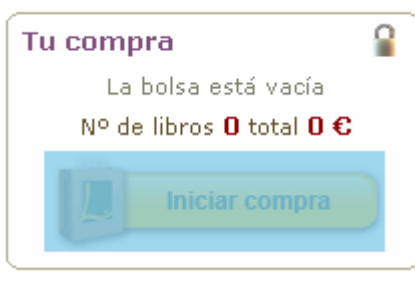
A través de esta pestaña, Firebug nos permite ver el código HTML generado y editarlo, visualizando los cambios inmediatamente, además mediante el inspector de HTML facilita mucho la tarea de encontrar el código relativo a un elemento de la página.

4.2.1. Visualización de código HTML

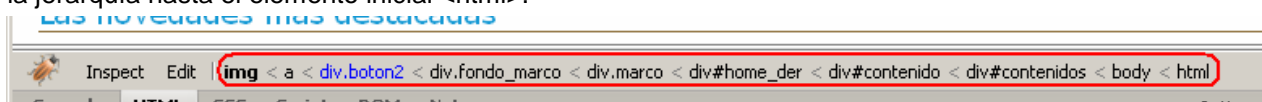
En la pestaña 'HTML', se muestra el código HTML generado para la página dividido jerárquicamente. Inicialmente aparecen desplegados únicamente los elementos principales de toda página HTML: <head> y <body>. A partir de ellos se puede ir desplegando el código de la página para encontrar cualquier elemento integrado en ella. En el ejemplo con el que vamos a describir la funcionalidad hemos seleccionado un elemento imagen: .



Cuando mantenemos el ratón encima del código de un elemento de la página, se sombrea dicho elemento para ayudarnos a identificarlo.

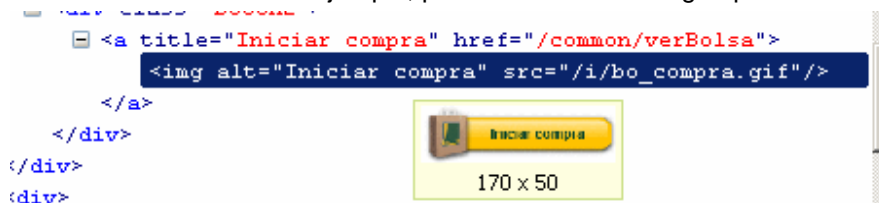


Por otro lado, seleccionando el elemento en el código, en el menú rápido se muestra la estructura jerárquica de los elementos que contienen al seleccionado, pudiendo seleccionar cualquiera de ellos, subiendo así en la jerarquía hasta el elemento inicial <html>.



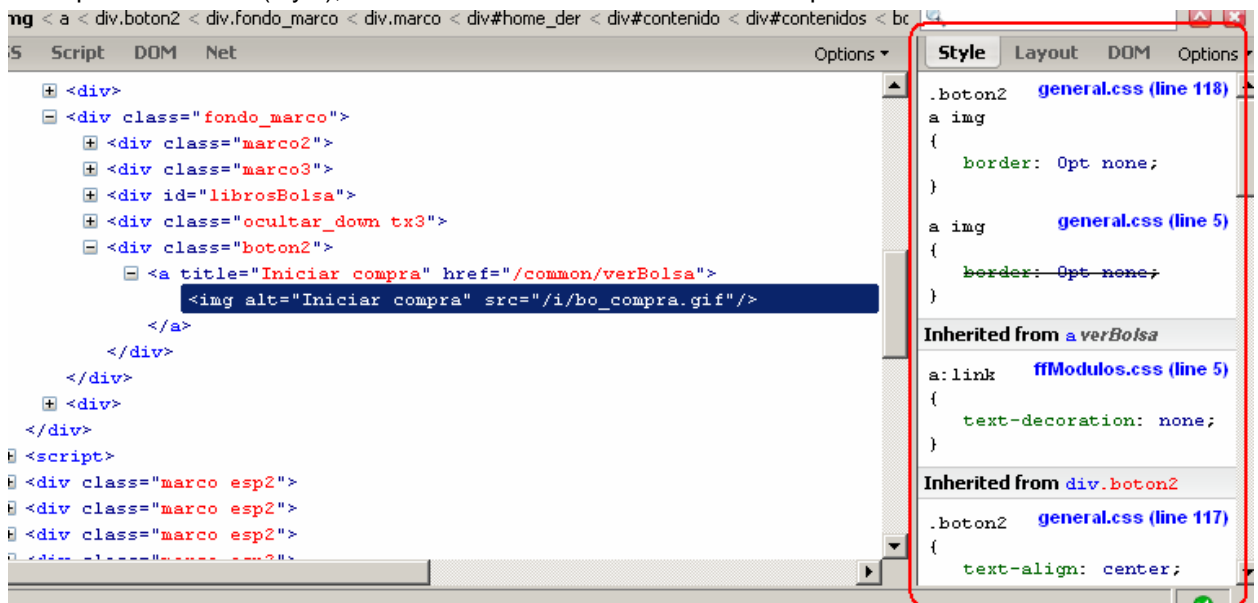
En el panel auxiliar, se muestra información más detallada del elemento seleccionado: estilos aplicados, colocación en la composición de la página y objetos DOM que le afectan. A continuación se muestra una breve explicación de las pestañas que aparecen en el panel auxiliar.

Si mantenemos el ratón sobre el elemento seleccionado, en función del tipo de elemento que es, aparece una información contextual. Por ejemplo, para un elemento `` aparece la imagen a la que hace referencia:



Estilos

En la pestaña estilos (style), se muestra información de los estilos aplicados al elemento seleccionado.



La información se muestra en orden de relevancia, es decir, apareciendo primero el estilo heredado del elemento más cercano jerárquicamente al seleccionado.

En el ejemplo puede verse el estilo del borde que se aplica al elemento seleccionado. Se puede ver también como la definición del estilo de borde que se da a los elementos `` dentro de un elemento `<a>`, no se tiene en cuenta, ya que dicha definición queda sobrescrita por la que figura unas líneas más abajo en la hoja de estilos (css) de la clase 'boton2'.

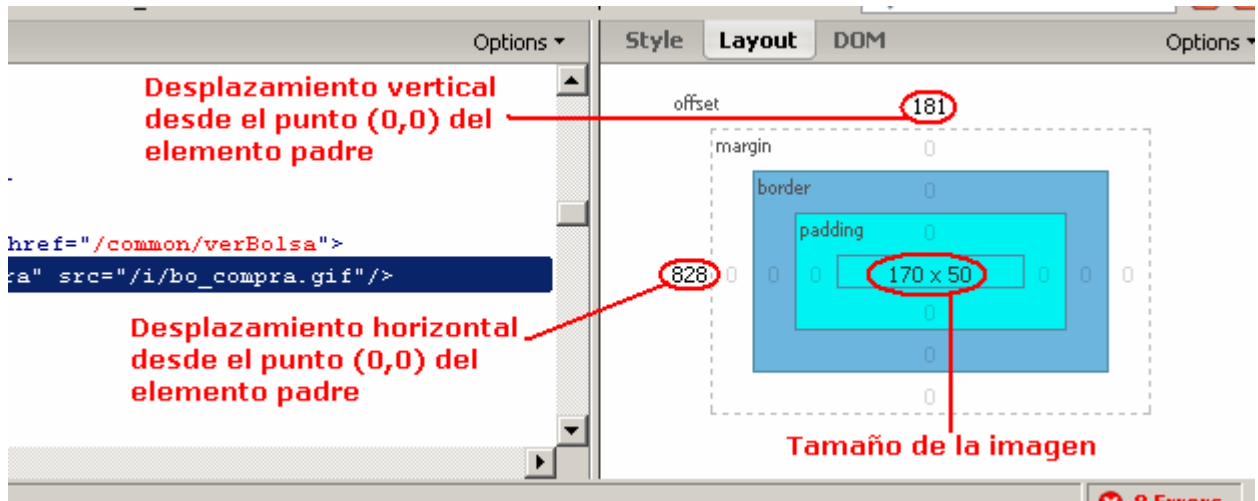
Se van mostrando todos los estilos aplicados, indicando la hoja de estilos en la que se definen e incluso el número de línea. Si se pincha en la css que define un estilo, Firebug presentará en la pestaña 'CSS' la línea donde se define el estilo.

Se muestran tachados los estilos que a pesar de ser aplicables al elemento seleccionado, son sobrescritos por estar definidos también en un elemento de relevancia mayor.

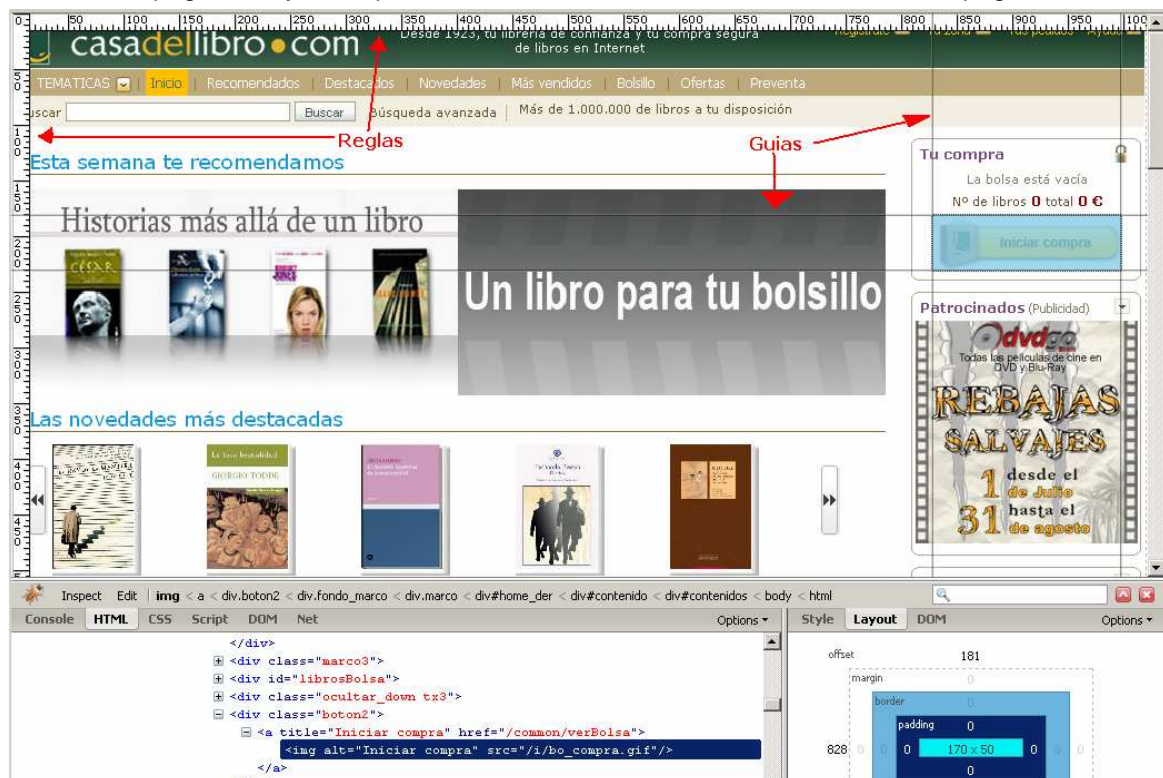
Ubicación en la composición de la página

En el panel auxiliar de la pestaña 'HTML' se muestra también información relativa a la ubicación dentro del diseño de la página del elemento seleccionado. Esta información se presenta en la pestaña 'Layout'.

En la siguiente figura se detalla el tipo de información que se puede obtener para un elemento.

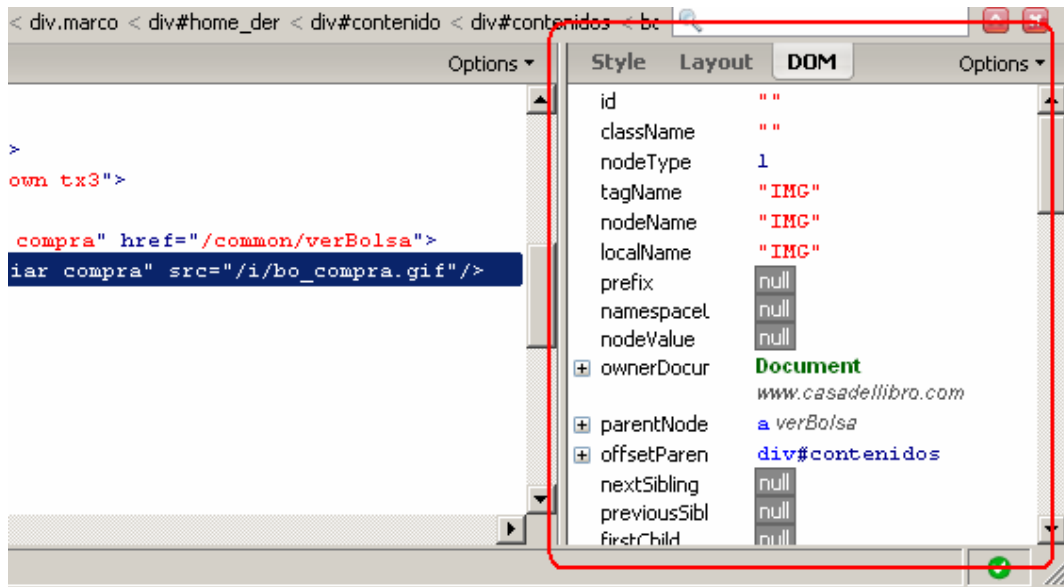


Además, pasando con el ratón por encima de las diferentes cajas que conforman la ubicación y tamaño del elemento en el contexto de la página HTML, aparecen unas guías y una regla (en pixels) sobre la visualización de la página, muy útiles para la tarea de alinear elementos de una misma página.



Información del objeto DOM

En la pestaña 'DOM', se muestra toda la información de forma jerarquizada, del objeto DOM creado por el navegador para el elemento seleccionado.



4.2.2. Búsqueda de código HTML

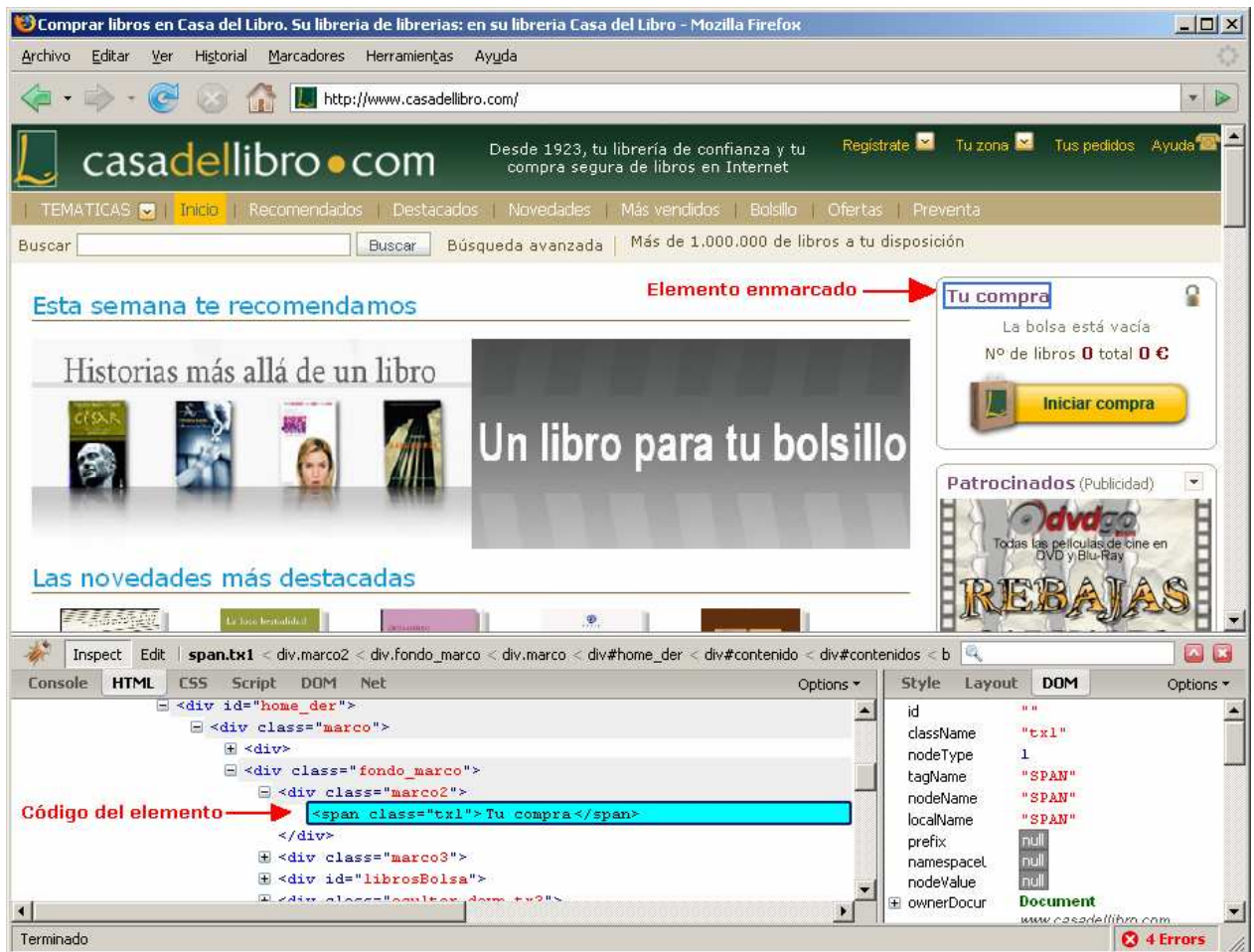
Otra de las tareas que facilita Firebug es la de encontrar código HTML.

Es conocido que cada vez el código HTML de una página es más complejo. Firebug ofrece dos formas muy sencillas de encontrar el código asociado a un elemento, bien de forma visual (Inspector), bien conociendo el nombre del elemento (panel de búsquedas).

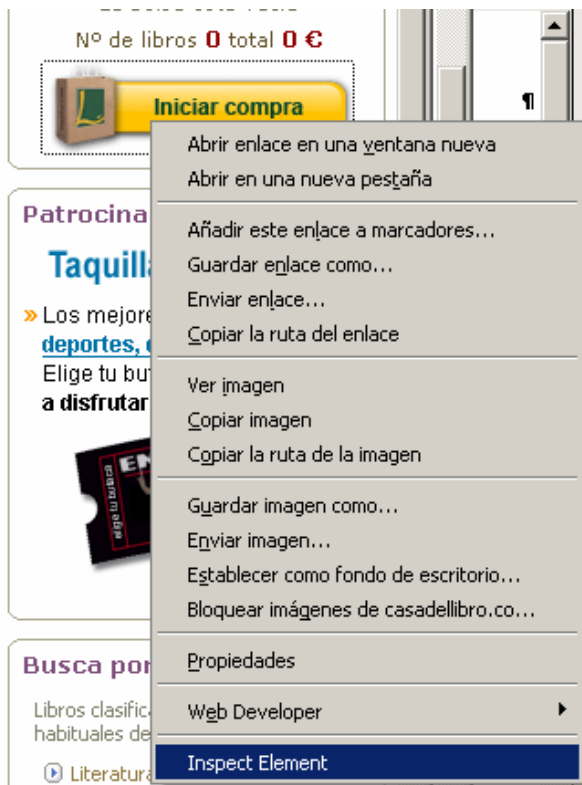
Inspector HTML

Una de las funcionalidades más innovadoras de la pestaña 'HTML' es la posibilidad de encontrar el código de un elemento pasando el ratón sobre dicho elemento. Para ello:

1. Pulsar el botón 'Inspect' que se encuentra en el menú rápido de Firebug.
2. Pasar el ratón sobre los distintos elementos de la página, se podrá ver como aparecen enmarcados los distintos elementos y como en la zona de edición del HTML aparece el código del elemento enmarcado.
3. Pinchar finalmente sobre un elemento enmarcado si se desea pasar a la edición del código de dicho elemento.



Otra forma todavía más rápida si cabe es pinchar con el botón derecho del ratón sobre un elemento del que se desea buscar el código. En el menú contextual que aparecerá, ir a la opción 'Inspect Element'.



Se desplegará la ventana de Firebug, si todavía no lo estaba, y en la pestaña HTML aparecerá el texto del elemento seleccionado.

```

+ <div class="ocultar_down tx3">
- <div class="boton2">
  - <a title="Iniciar compra" href="/common/verBolsa">
    
  </a>

```

Búsqueda rápida HTML

Otra de las opciones que ofrece Firebug es la de buscar un elemento si se conoce su nombre: id, class, tipo de elemento, etc. Para ello, en el panel de búsquedas introducir el texto que se desea buscar y pulsar 'Enter' sucesivamente hasta encontrar el elemento buscado.

4.2.3. Edición de código HTML

Como se irá viendo a lo largo del presente manual, Firebug facilita mucho el desarrollo de páginas web mediante el método de prueba y error. Es decir, permite modificar el código de una página para ver como quedaría, sin necesidad de cambiar los ficheros fuente y recargar la página de nuevo.

Por supuesto, una de las cosas que permite modificar es el código HTML. A continuación para explicar esta funcionalidad, se mostrarán dos formas de cambiar el título del enlace que contiene la imagen del ejemplo, es decir, la propiedad 'title' del elemento <a>.

```

+ <div class="ocultar_down tx3">
- <div class="boton2">
  - <a href="/common/verBolsa" title="Iniciar compra">
    
  /a>

```

Eso hace que al pasar con el ratón sobre la imagen aparezca un texto de ayuda que dice “Iniciar compra”, ver imagen:



Modificación desde ventana de código

El primer modo y quizá el más sencillo para los desarrolladores acostumbrados a editar HTML desde un editor de textos normal, es el de realizar la modificación desde la ventana de código de la pestaña HTML.

1. Hacer un sólo clic sobre el texto que se quiere editar. Firebug presentará el texto resaltado.

```

<div class="boton2">
  - <a href="/common/verBolsa" title="Iniciar compra">
    
  /a>

```

2. Editar el texto.

```

<div class="boton2">
  - <a href="/common/verBolsa" title="Comenzar compra">
    
  /a>

```

3. El cambio en la página HTML deberá haberse realizado automáticamente, sin necesidad de recargar la página.



También es posible realizar el mismo cambio pulsando el botón ‘Edit’ del menú rápido de la pestaña HTML.

1. Seleccionar el elemento que se desea editar:

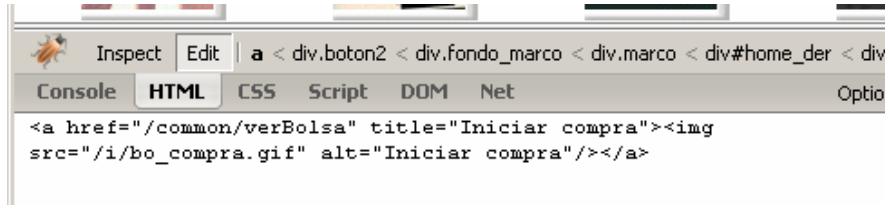
```

<div class="boton2">
  - <a href="/common/verBolsa" title="Iniciar compra">
    
  /a>

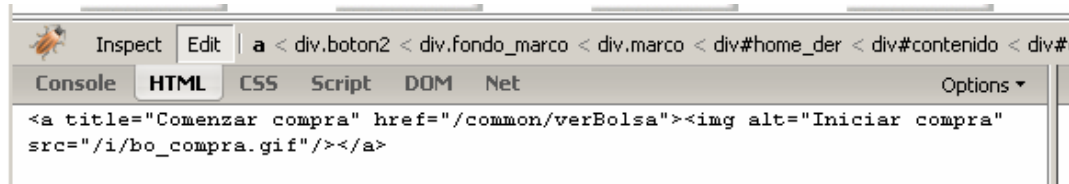
```

2. Pulsar el botón ‘Edit’ del menú rápido: . Firebug presentará en la ventana de edición el texto

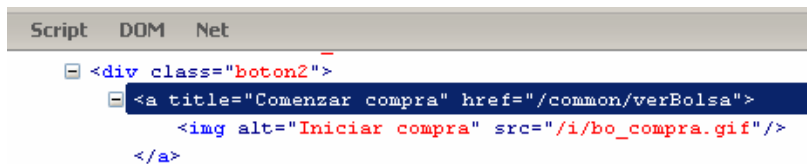
para editar, como si fuera un editor de textos normal.



3. Al modificar el texto, Firebug presentará los cambios inmediatamente sin necesidad de guardar ningún cambio ni de recargar la página.



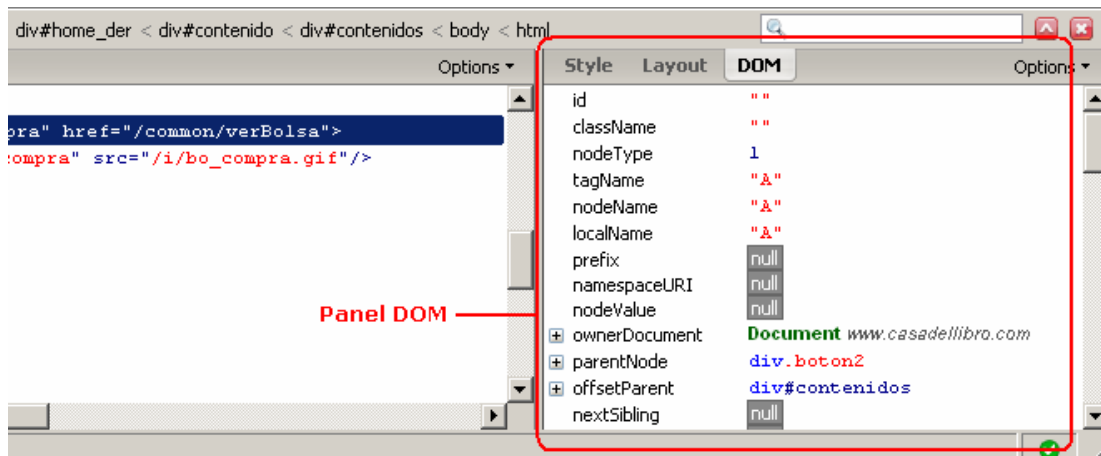
4. Para volver al modo normal pulsar de nuevo el botón 'Edit'. Se verá como los cambios también se mantienen en este modo.



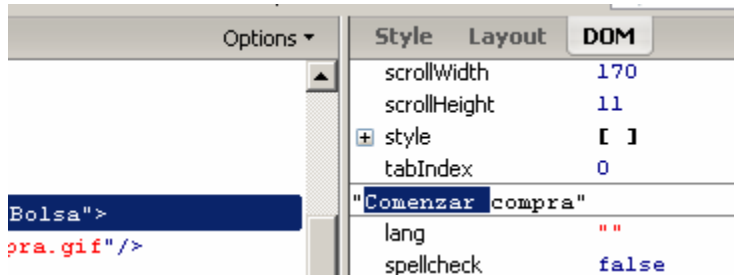
Modificación desde pestaña DOM

Otra forma de modificar las propiedades de los elementos de una página HTML, es ir al objeto DOM que define el elemento. Para ello:

1. Seleccionar el elemento que se desea editar sobre la ventana de código HTML.
 - <div class="boton2">
 -
 -
2. Ir a la pestaña DOM del panel auxiliar (no confundir con la pestaña DOM de la barra de pestañas) donde aparecerán las propiedades del elemento seleccionado.



3. Buscar la propiedad que se desea cambiar, en este caso: 'title', y modificarla.



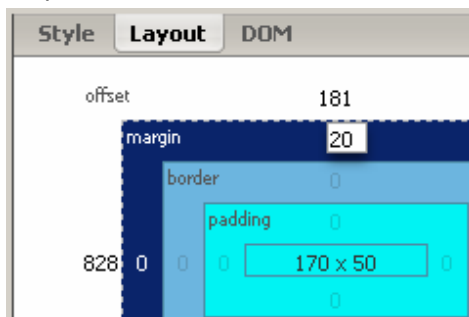
4. Los cambios se verán reflejados automáticamente, sin necesidad de recargar la página.



4.2.4. Modificación de las dimensiones de la caja de un elemento

Como se ha visto antes, en la pestaña Layout del panel auxiliar, aparecen las dimensiones que se tienen en cuenta para la ubicación de un elemento en una página HTML.

Es posible editar dichas dimensiones, únicamente haciendo clic sobre el dato que se desea alterar.



4.3 CSS

En la pestaña CSS se puede acceder fácilmente a la edición de estilos de la página, es muy intuitivo y tremendamente útil ya que todos los cambios se visualizan en tiempo real.

En el menú rápido aparece un botón desplegable con las diferentes hojas de estilos cargadas por la página.



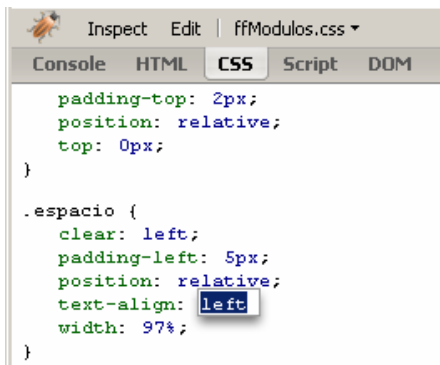
Seleccionar el fichero del que se quieren ver los estilos.

Para ver los estilos aplicados a un elemento en cuestión, ir a la pestaña HTML y actuar como se ha explicado en el punto 4.2.1.

4.3.1. Modificación de propiedades de un estilo

La modificación de los estilos de un elemento, es tan fácil como la modificación de una propiedad HTML. Vamos a ver un ejemplo con la alineación de uno de los textos de la página ejemplo.

1. Hacer un sólo clic sobre el estilo que se quiere editar. Firebug presentará el texto resaltado. En este caso es la propiedad text-align.



2. Modificar la propiedad. Firebug ofrece la posibilidad de utilizar las flechas de arriba y abajo para recorrer en orden alfabético todos los valores posibles que tiene la propiedad. Firebug dispone de un completo diccionario de palabras claves del estándar CSS, además de algunas propietarias de Mozilla.

La opción de recorrer las opciones posibles, se da tanto para los valores de una propiedad, como para el nombre clave de la propiedad.

Para eliminar una propiedad, simplemente modificarla dejándola en blanco. Firebug, la eliminará del estilo.

4.3.2. Modificación de estilos

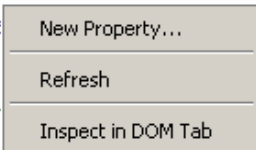
Es posible también añadir propiedades a un estilo. Puede hacerse de dos formas:

- Haciendo doble clic sobre el estilo al que se quiere añadir la propiedad. Se resaltará el hueco donde incluir la propiedad. Escribir en ese hueco la nueva propiedad.


```
.espacio {
  clear: left;
  padding-left: 5px;
  position: relative;
  text-align: left;
  width: 97%;
   undefined;
}
```

- Pinchando con el botón derecho del ratón y sobre el menú contextual que aparece seleccionar: 'New Property...'

```
.espacio {
  clear: left;
  padding-left: 5px;
  position: relative;
  text-align: left;
  width: 97%;
}
```



4.3.3. Deshabilitación de estilos

Cuando se pasa con el ratón por encima de la propiedad de un estilo, aparece el símbolo: , pinchando sobre él, se deshabilita la propiedad indicada, provocando los oportunos cambios en la visualización del HTML.

```
.espacio {
  clear: left;
  padding-left: 5px;
  position: relative;
   text-align: left;
  width: 97%;
}
```

4.4 Script

Como se ha comentado anteriormente en este manual, Firebug ofrece un gran número de funciones para ayudar a la depuración de código Javascript. Muchas de esas funciones se llevan a cabo en la pestaña 'Console' y ya han sido comentadas, pero otras se realizan en la pestaña 'Script' y se detallarán en este punto.

En la pestaña 'Script' se ofrece la posibilidad de poner puntos de ruptura (breakpoints), ver el valor de las variables en tiempo de ejecución, ejecutar paso a paso un conjunto de sentencias, etc. Así se consigue controlar la ejecución del código Javascript de una página.

4.4.1. Puntos de ruptura

Firebug da la posibilidad de poner puntos de ruptura, lo que permite al depurador parar en una línea específica. Cuando esto ocurre, pueden verse los valores de las variables, inspeccionar objetos, continuar la

ejecución paso a paso, etc.

Puntos de ruptura en una línea específica

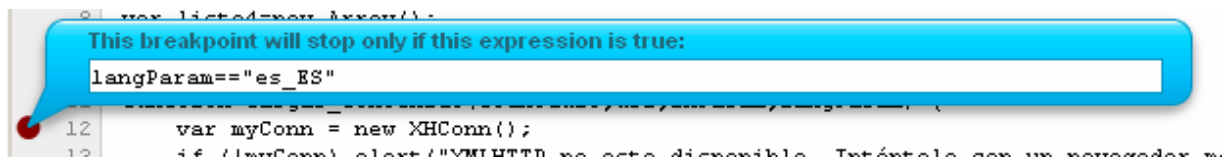
Para poner puntos de ruptura en un línea específica, únicamente pinchar con el ratón en la parte izquierda de la zona de código, donde aparece la numeración de la línea. Firebug indicará el punto de ruptura con un círculo.

```
9  var c=0;
10
11 function cargar_contenido(codModulo,url,inParam,langParam) {
12     var myConn = new XHConn();
13     if (!myConn) alert("XMLHTTP no esta disponible. Inténtalo con un navegador m...");
14     var peticion = function (oXML) {
15         var respuesta = oXML.responseText;
16         if (respuesta.indexOf("<")>0){
17             respuesta2=eval('(' + respuesta + ')');
18             c=respuesta2.lib.length;
```

Puntos de ruptura condicionales

En ocasiones, al desarrollador no le interesa que se pare la ejecución siempre, sino únicamente cuando se cumplen unas determinadas condiciones. Firebug permite poner puntos de ruptura condicionales, de tal forma que sólo parará la ejecución si al evaluar una condición especificada previamente el resultado es verdadero.

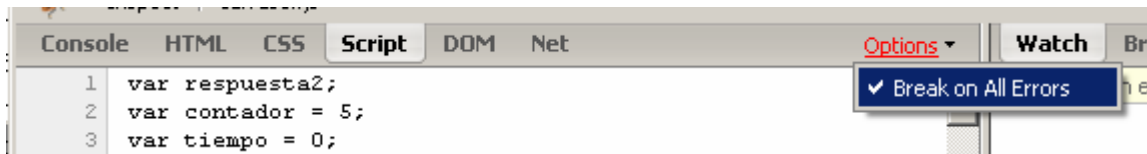
Para poner un punto de ruptura condicional, realizar la misma acción que en el punto anterior, pero pinchando en la línea de código con el botón derecho del ratón. Aparecerá un cuadro de diálogo en el que introducir la condición.



Parar ejecución si existe error

Firebug también da la posibilidad de parar automáticamente en caso de error, para realizar la depuración del código que ha dado el problema.

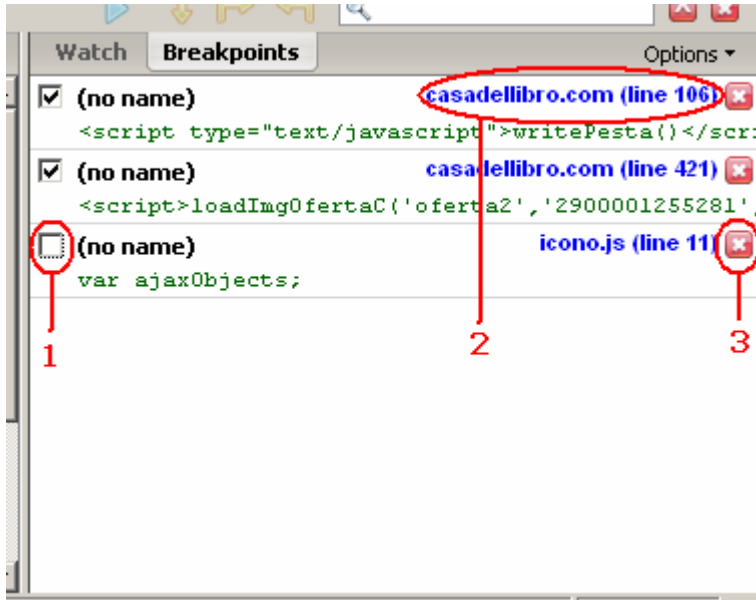
Para activar esta opción, a la derecha de la barra de pestañas está el botón 'Options'. Pulsando en él aparece un menú contextual con la opción 'Break on All Errors' (parar en los errores). Se mostrará un tico a la izquierda de la opción si está activada, si no lo está, pinchar en la opción 'Break on All Errors' para activar.



Si durante la navegación existe un error, Firebug desplegará su ventana de ejecución (si no estuviera desplegada) y mostrará la línea en la que ocurrió el error.

Gestión de puntos de ruptura

Como la mayoría de los depuradores gráficos, Firebug ofrece la posibilidad de visualizar todos los puntos de ruptura que hay definidos para una página web. En este caso, Firebug los muestra en el panel auxiliar, en la pestaña 'Breakpoints'.



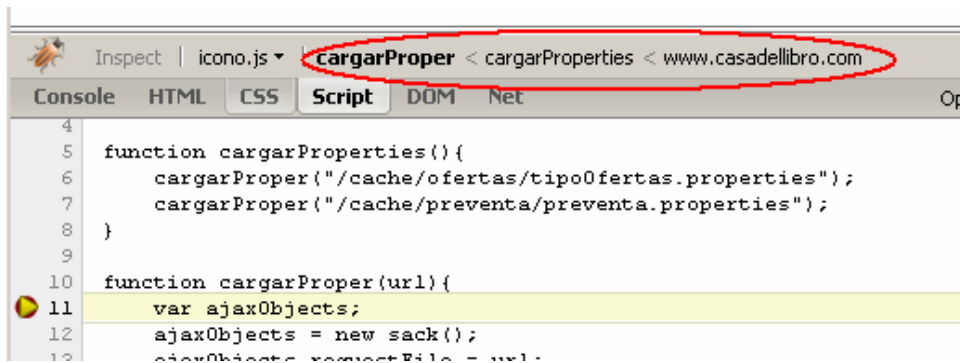
Las acciones que se pueden realizar con cada punto de ruptura definido son:

- Deshabilitarlo. De esta forma, sigue definido, pero la ejecución no se parará en ese punto. Para ello desmarcar el cuadro de comprobación (checkbox) que aparece a la izquierda de la definición (1).
- Ir a la línea de código en el que está definido. Para ello pinchar en el enlace en el que indica la línea (2).
- Eliminar el punto de ruptura. Para ello pinchar en el símbolo con la cruz que aparece a la derecha de la línea de la definición (3).

Pila de llamadas

Cuando se detiene la ejecución de un script, Firebug muestra la pila de llamadas hasta llegar a ese punto.

Tal como muestra la figura, en la barra de menú rápido aparecen las funciones que se han llamado hasta llegar al punto en el que se ha parado la ejecución. Pinchando en cualquiera de ellas, lleva al punto (de la función seleccionada) en el que se originó la llamada.



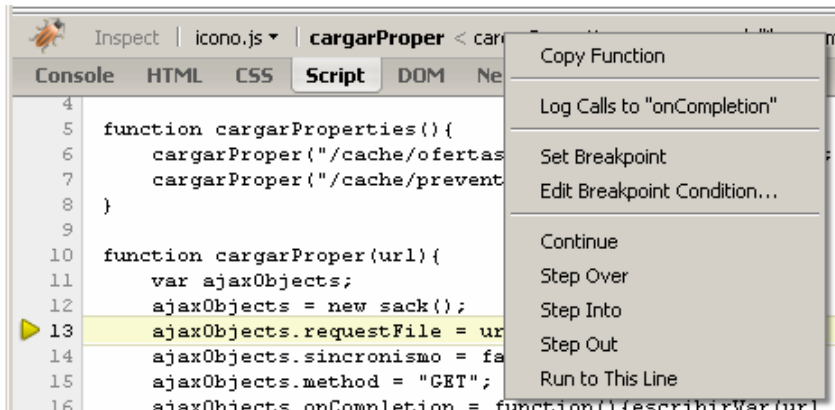
4.4.2. Ejecución paso a paso

Como cualquier depurador, Firebug ofrece la posibilidad de ejecutar paso a paso el código. Para ello pueden elegirse tres métodos.

- Utilizando los botones para tal fin dispuestos en el menú rápido.







- Utilizando el menú contextual que aparece al pinchar con el ratón sobre la zona de código.



- Utilizando las teclas de función (F8, F10 y F11).

Y como cualquier depurador ofrece las siguientes posibilidades:

- Ejecutar hasta la siguiente línea: (, Step Over y F10).
- Ejecutar hasta la primera línea de la función llamada: (, Step Into y F11).
- Ejecutar hasta la siguiente línea de la función llamante: (, Step Out. No tiene tecla función)
- Continuar (, Continue y F8)
- Ejecutar hasta una línea dada. Para ello posicionarse en la línea deseada, abrir el menú contextual pinchando con el botón derecho del ratón y acceder a la opción 'Run on This Line'.

4.4.3. Inspección de expresiones

Como cualquier depurador, Firebug ofrece la posibilidad de ver los valores de las variables en tiempo de ejecución. También ofrece varias formas de hacer esto.

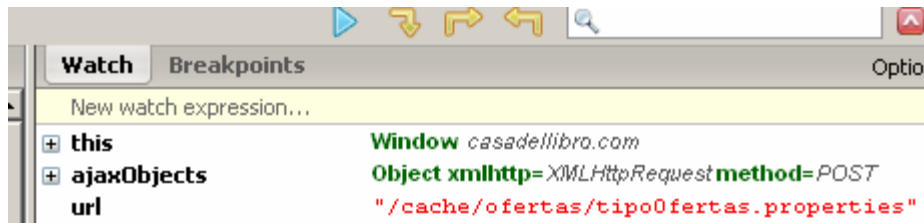
- Si la variable u objeto a inspeccionar está cerca (y en una línea anterior) del punto en el que está detenida la ejecución, simplemente con pasar con el ratón por encima de dicha variable, Firebug mostrará una etiqueta de ayuda (tooltip) con el valor de la variable. Ver imagen:

```

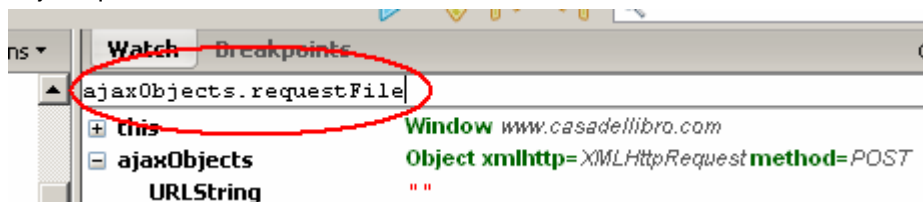
9
10 function cargarProper(ur] {
11   var ajaxObjects;  "/cache/preventa/preventa.properties"
12   ajaxObjects = new sack();


```

- En la pestaña 'Watch' del panel auxiliar, aparece un cuadro de texto que pone 'New watch expression...':



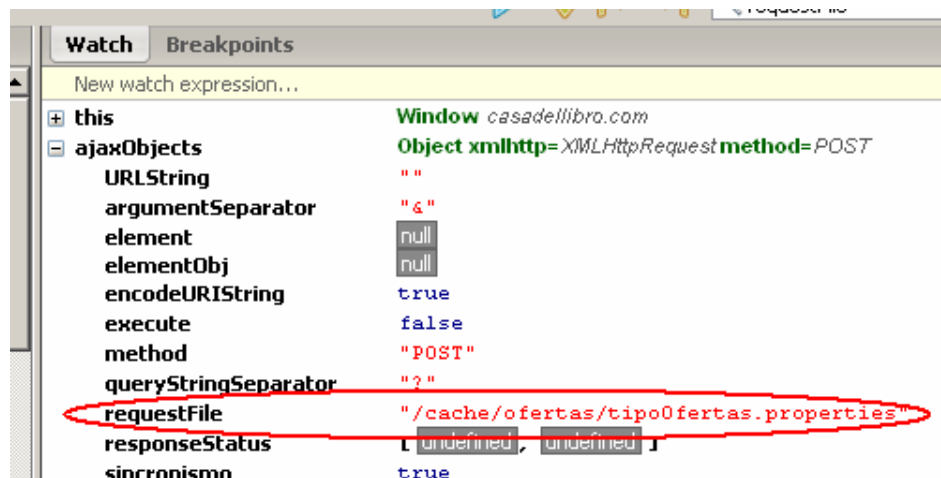
Al pinchar encima con el ratón se pondrá en modo edición. Escribir la variable o propiedad del objeto que se desee estudiar.



Firebug mostrará la expresión y su valor. A la derecha de la línea con la expresión aparecerá el símbolo: . Pinchando en él se elimina la expresión de la lista.

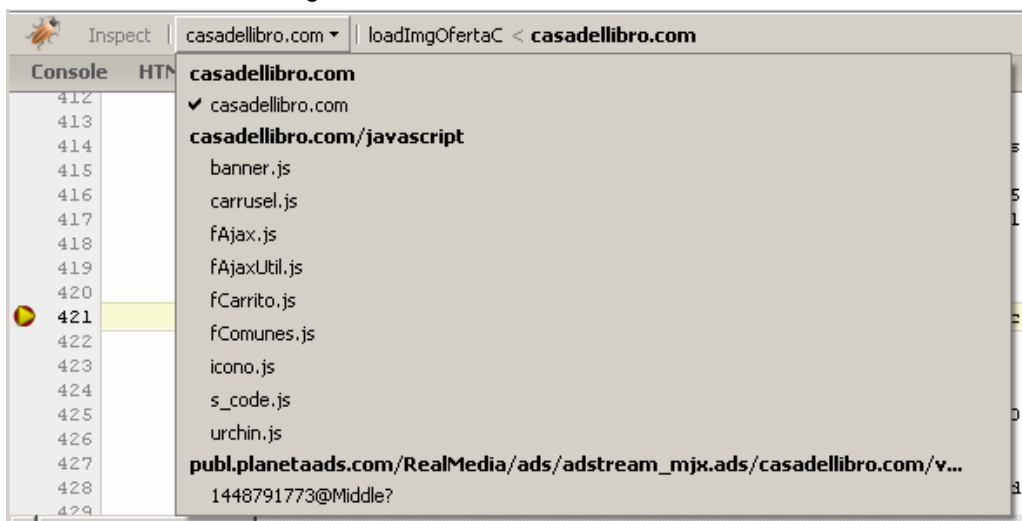


- Por último, se puede observar una variable, también buscándola en la lista de objetos que aparece en la pestaña 'Watch'. Por ejemplo, para buscar: `ajaxObjects.requestFile`, se despliega el objeto `ajaxObjects` y se busca entre sus propiedades que aparecen ordenadas alfabéticamente (con las mayúsculas primero).



4.4.4. Gestión de scripts

En la zona de código de la pestaña 'Script', aparece el código de un determinado script cargado por el sitio web que se está visitando. En el menú rápido, Firebug muestra un desplegable con todos los scripts cargados por la página, incluso da la posibilidad de ver el código Javascript que se encuentra directamente en la página HTML, el definido entre los tags `<script></script>`, pudiendo también en este caso, poner puntos de ruptura en dichas líneas de código.



Búsqueda de código en un script

Con el panel de búsquedas puede buscarse cualquier texto en el script seleccionado. No busca en todos los scripts, sino únicamente en el seleccionado.

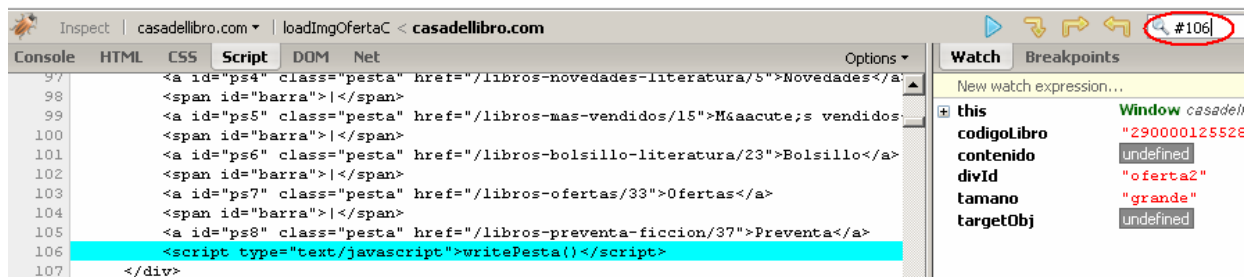


La búsqueda la realiza del mismo modo que lo hace Firefox, inmediatamente después de cada pulsación de tecla.

Para búsquedas sucesivas, pulsar Enter cuando el foco está en el panel de búsquedas.

Ir a un línea específica de un script

Para ir a una determinada línea de un script, estando seleccionado dicho script, en el panel de búsquedas, escribir: '#nº de línea'. Por ejemplo, para ir a la línea 106 se escribirá '#106'.

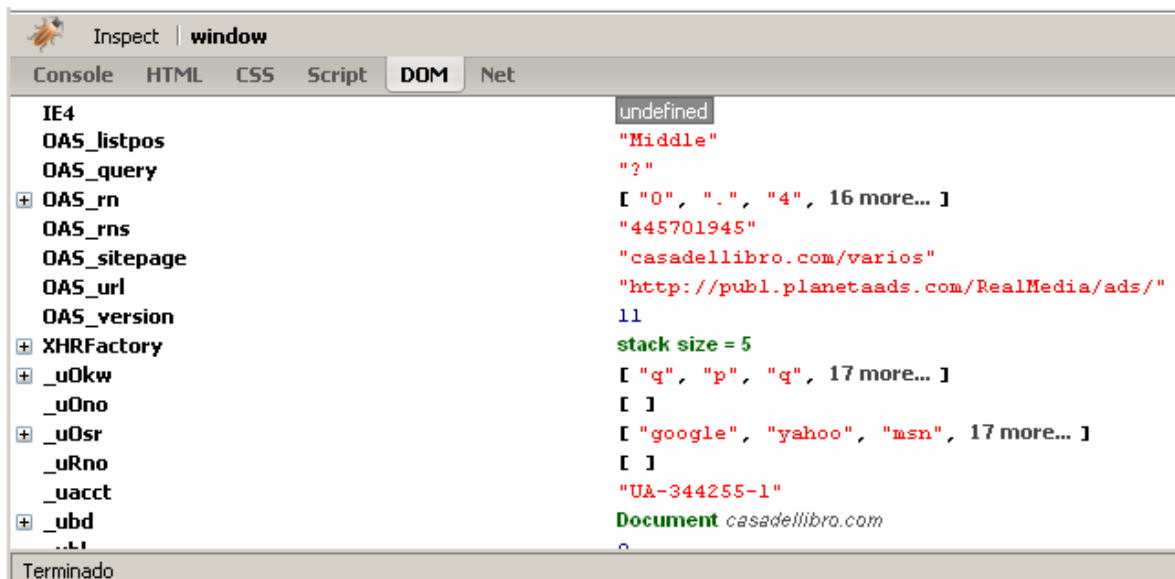


Firebug, además de ir a la línea, la resaltará temporalmente para facilitar su localización.

4.5 DOM

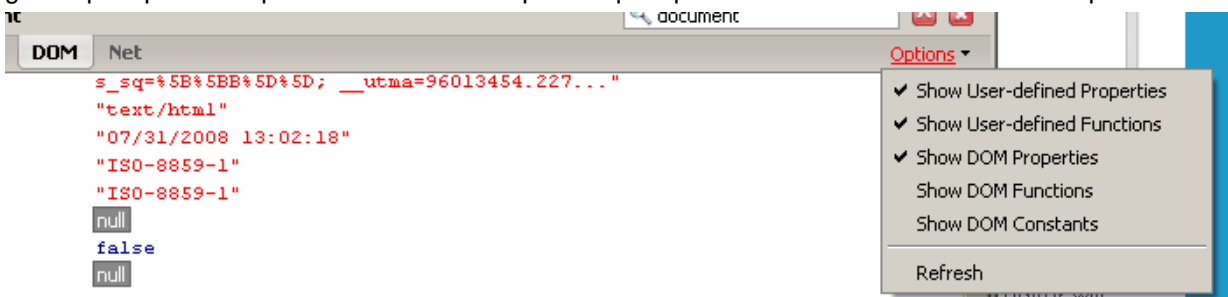
Firebug también ofrece la posibilidad de visualizar de una forma ordenada y de modificar al vuelo los objetos DOM creados por el navegador al cargar una página web. Los objetos DOM (Document Object Model) son la interfaz que crea un navegador para que a través de scripts se permita el acceso a ellos y a sus propiedades, tanto de visualización como de modificación en alguna de ellas.

En la pestaña 'DOM' se muestran todos los objetos creados a partir del objeto window (que es el padre de todos ellos).



Existen objetos DOM estándar, como: window, document, etc. y objetos definidos por el desarrollador de la página web. Firebug, muestra en negrita y en primer orden en la lista los definidos por el desarrollador y sin negrita los objetos estándar. Además distingue por colores entre objetos y propiedades (en negro), funciones (en verde) y constantes (en negro y mayúsculas).

Además ofrece la posibilidad de mostrar o esconder los objetos en base a su origen, seleccionándolos del desplegable que aparece al pinchar en el menú 'Options' que aparece a la derecha de la barra de pestañas.

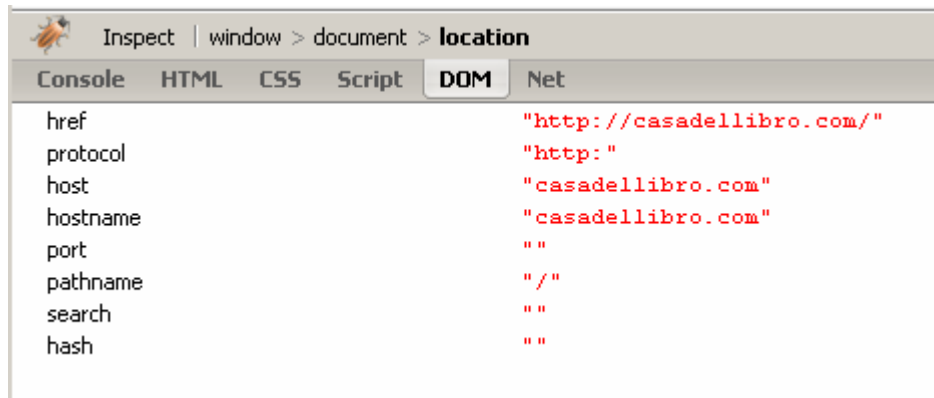


4.5.1. Visualización de la información

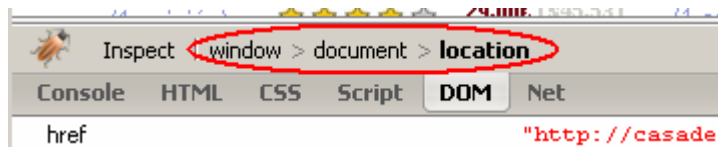
Existen muchas clases diferentes de objetos (Array, Texto, Fecha, Tipo Document, numéricos, etc.). Firebug intenta en la línea en la que visualiza el objeto mostrar la mayor información posible para cada uno de esos tipos, ya que no es lo mismo mostrar el valor de una variable de tipo texto que el valor de todos los elementos de un array.



Además, inicialmente Firebug muestra todos los objetos DOM derivados del objeto raíz 'window' en la zona de edición de la pestaña 'DOM', pero es posible que toda la zona la ocupe la información de un único objeto y sus propiedades, para ello, simplemente pinchar con el ratón en la columna de la derecha en la fila correspondiente al objeto. Firebug, utilizará toda la zona de edición para mostrar la información del objeto. Ver en la imagen el ejemplo al pinchar sobre: 'window.document.location'.



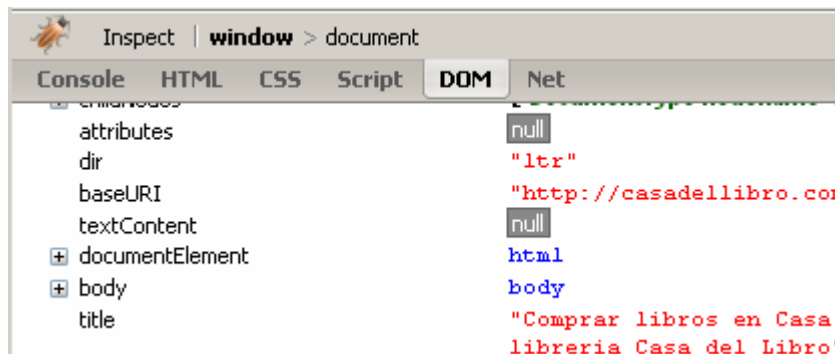
Además al ir accediendo a diferentes objetos, Firebug mostrará toda la jerarquía de objetos hasta llegar al seleccionado, pudiendo pinchar en cualquiera de ellos para cambiar de nuevo la vista de la zona de edición.



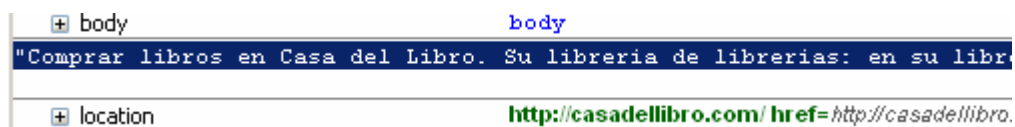
4.5.2. Modificación de propiedades

Además de visualizar el valor de todos los objetos y sus propiedades, Firebug permite también editarlos y mostrar los cambios 'al vuelo'. Por ejemplo, para modificar el título de una página habrá que editar la propiedad: window.document.title. Para ello:

1. Desplegar el objeto 'document' y buscar la propiedad 'title'.



2. Hacer doble clic sobre la columna izquierda de la fila en la que está definida la propiedad o pinchar con el botón derecho del ratón sobre la fila y en el menú contextual que aparece, seleccionar 'Edit Property...'



3. Editar la propiedad y pulsar Enter.

Firebug mostrará los cambios inmediatamente.



Al recargar la página de nuevo, Firebug restablecerá los cambios, volviendo a los valores originales.

Escritura automática de objetos y propiedades

La caja de texto de edición de propiedades en Firebug viene a ser una pequeña línea de comandos javascript, por tanto, pueden utilizarse sentencias javascript para modificar el valor de una determinada propiedad de un objeto DOM. Cualquier cosa que se escriba en la caja de edición, se evaluará en el contexto actual y su resultado se le asignará a la propiedad que se está editando.

Firebug ofrece una funcionalidad muy útil para encontrar una propiedad o un objeto si por ejemplo no se recuerda exactamente su nombre pero sí por que letra empieza. Permite recorrer las propiedades de un objeto utilizando el tabulador.

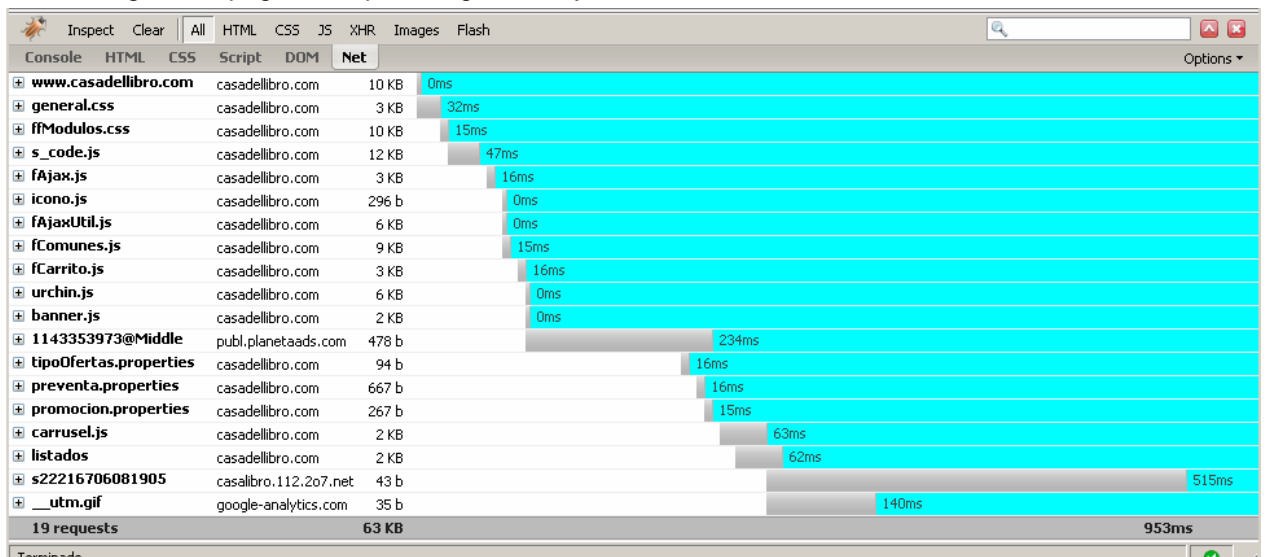
Puede verse un ejemplo actualizando el título de una página con la URL de la misma. Para seguir el ejemplo, hacer:

1. Escribir: "document.loc".
2. Pulsar el tabulador hasta que se complete a: "document.location"
3. Añadir un punto: "."
4. Pulsar el tabulador hasta que se complete a "document.location.href"

Esta facilidad es tremendamente útil para buscar nombres de variables o funciones en el complejo entramado de una página web.

4.6 NET

Por último, la pestaña NET ofrece una vista del tiempo de carga de los diferentes elementos de una página web: código de la página, scripts, imágenes, hojas de estilo, ficheros XML.



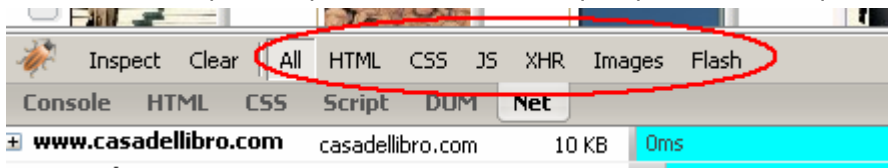
Request Name	Source	Size	Load Time
www.casadellibro.com	casadellibro.com	10 KB	0ms
general.css	casadellibro.com	3 KB	32ms
ffModulos.css	casadellibro.com	10 KB	15ms
s_code.js	casadellibro.com	12 KB	47ms
fAjax.js	casadellibro.com	3 KB	16ms
icono.js	casadellibro.com	296 b	0ms
fAjaxUtil.js	casadellibro.com	6 KB	0ms
fComunes.js	casadellibro.com	9 KB	15ms
fCarrito.js	casadellibro.com	3 KB	16ms
urchin.js	casadellibro.com	6 KB	0ms
banner.js	casadellibro.com	2 KB	0ms
1143353973@Middle	publ.planetaads.com	478 b	234ms
tipoOfertas.properties	casadellibro.com	94 b	16ms
preventas.properties	casadellibro.com	667 b	16ms
promocion.properties	casadellibro.com	267 b	15ms
carrusel.js	casadellibro.com	2 KB	63ms
listados	casadellibro.com	2 KB	62ms
s22216706081905	casalibro.112.2o7.net	43 b	515ms
__utm.gif	google-analytics.com	35 b	140ms
19 requests		63 KB	953ms

Cada fichero se muestra en una línea en la zona de visualización, indicando cuando comenzó a descargarse, en relación al resto de ficheros de la página y cuanto tiempo tardó.

Esta visión, muestra como los ficheros nunca se descargan en paralelo, sino uno detrás de otro. Esto puede ayudar al desarrollador a definir el orden de descarga, para que el usuario no tenga que esperar demasiado para visualizar la primera parte de la página.

4.6.1. Filtrado por tipo de fichero

Firebug ofrece la posibilidad de filtrar por tipo de fichero. Algunas veces, al desarrollador sólo le interesa conocer las prestaciones de las imágenes o de los ficheros javascript, en ese caso con los botones que aparecen en el menú rápido, se puede seleccionar de qué tipo de ficheros queremos ver la información.



4.6.2. Control de las peticiones HTTP

Las cabeceras de las peticiones HTTP ofrecen información muy útil, como: el tipo mime del fichero descargado, el tipo del servidor web, las directrices para las cachés con respecto a dicho fichero, etc. Firebug, simplemente pinchando con el ratón sobre el fichero en la pestaña 'Net', muestra las cabeceras tanto de la petición como de la respuesta, así como los datos de la respuesta.



Esa misma información podía conseguirse para las peticiones AJAX desde la pestaña 'Console', pero en esta pestaña puede conseguirse para cualquier tipo de fichero descargado desde un servidor web, incluso las peticiones AJAX.

Comprobación de la carga desde la caché

No todas las peticiones HTTP son iguales, unas veces, la petición no llega a lanzarse a la red, sino que la respuesta la da la caché del navegador. Firebug, muestra en un gris más claro las peticiones a las cuales ha respondido la caché del navegador.

Console	HTML	CSS	Script	DOM	Net
+ www.iberia.com			iberia.com		14 KB 140ms
+ iberia.css			iberia.com		16 KB 125ms
+ etiquetas.css			iberia.com		3 KB 94ms
+ commons.js			iberia.com		30 KB 93ms