

Tema 1: Introducción a HTML5.



1. Breve historia de HTML5.
2. Líneas maestras de HTML5.
3. Las novedades de HTML5.
4. HTML5 es un lenguaje de etiquetas.
 - 4.1. Etiquetas.
 - 4.2. Atributos de etiqueta.
 - 4.3. Uso correcto de las etiquetas.
5. Etiquetas y atributos HTML 4.0 desaparecidos.
6. Los navegadores de nuestro estudio.
7. El documento HTML5 mínimo.
 - 7.1. El doctype y su importancia.
 - 7.2. La etiqueta HTML.
 - 7.2. El encabezado del documento.
8. El título del documento.
9. La codificación (charset) del documento.
10. El cuerpo del documento.
11. Modo de proceder.
12. Validación de código HTML5.
 - 12.1. El validador de W3C.
 - 12.2. El validador validator.nu
13. Prueba.

1.- Breve historia de HTML5.

Comenzamos esta historia en 1998 con la finalización de la especificación Html 4.0. Dejando aparte una ligera revisión en 1999 con Html 4.01, no ha habido nada nuevo en el horizonte de este primer lenguaje Web. El W3C (*World Wide Web Consortium*), organismo que gestiona los estándares de escritura en la red, ¿no anunció en su momento que Html, en su versión 4.0, no conocería ningún desarrollo posterior y, en breve, que Html estaba muerto?

El W3C editó, también en 1998, el estándar XML (*eXtensible Markup Language* o, en castellano, lenguaje de marcas extensible) para gestionar de forma estructurada los datos de tipo texto, donde el aspecto extensible del lenguaje permitía al usuario definir su propio lenguaje (comportándose como un metalenguaje) con sus distintos elementos. El W3C recomendaba encarecidamente, en la época, XML para expresar lenguajes de marcado específico.

Una de las primeras aplicaciones concretas de XML fue la reformulación de Html 4.0 según la sintaxis estricta y formalista de XML. Nació así Xhtml 1.0 en el año 2000. Si bien los desarrolladores saludaron unánimemente el rigor que Xhtml aportaba a la escritura de código, es justo señalar que la migración a Xhtml no ha supuesto el éxito esperado e incluso una gran parte de los diseñadores han permanecido fieles a Html 4.0, más laxo.

Muy rápidamente, el W3C puso en marcha el ambicioso proyecto de Xhtml 2.0, que debía suponer un punto y aparte respecto al pasado en materia de publicación en la red. Esta posición indujo a que Xhtml 2.0 no fuera compatible hacia atrás con todo lo que existía hasta la época, y por consiguiente con Html.

Esta ausencia de compatibilidad supuso un cierto descontento por parte de los fabricantes de software que desarrollaban los navegadores.

Todo ello tuvo como consecuencia la creación de un grupo disidente, el WHATWG (*Web Hypertext Application Technology Working Group*). Este grupo de trabajo nació como respuesta a la lentitud del desarrollo de estándares por parte de W3C y al carácter demasiado cerrado de su proceso interno de elaboración de especificaciones. En sus orígenes estaba compuesto por representantes de firmas como Mozilla, Opera y Apple con sus conocidos navegadores Firefox, Opera y Safari. Google no tardó en unirse a ellos. Su posición era diametralmente opuesta a la de Xhtml 2.0, trabajando de forma pragmática sobre la base de las implementaciones actuales y, por tanto, sobre la base de Html 4.0. Sus primeros esfuerzos se centraron en los Web Forms 2.0 para renovar los formularios y en el concepto de publicación en la web basado en Web Apps 1.0, es decir aplicaciones Web.

Paralelamente Xhtml 2.0, incluso tras un borrador (*working draft*) que aparece en julio de 2006, conoce un desarrollo particularmente laborioso y los rumores muestran numerosas disensiones en el seno de este grupo de

trabajo. Por otro lado, los navegadores antes citados ignoraron deliberadamente a Xhtml 2.0.

Parecía claro que Xhtml 2.0 tenía graves problemas. Sir Tim Berners-Lee, inventor de la Web y presidente del W3C, decide, a finales de 2006, reabrir un grupo de trabajo sobre Html que, de forma pragmática, retome la labor realizada por WHATWG. De este modo ha existido durante todo este tiempo un equipo de trabajo sobre Xhtml 2.0 y otro sobre Html5. Este último publicó el 23 de enero de 2008 un primer borrador de trabajo que después ha sufrido numerosas evoluciones.

Pero los días de Xhtml 2.0 estaban contados. A finales de 2009, el mismo Sir Tim Berners-Lee anuncia la disolución del grupo de trabajo sobre Xhtml 2.0 y el abandono definitivo de este último.

Html5 pasó a *candidate recommendation* (recomendación candidata) en diciembre de 2012 y se debería finalizar en el último trimestre de 2014.

El anuncio de Html5 ha causado un gran revuelo (un buzz, en el vocabulario moderno). Este interés se ha concretado rápidamente a diversos niveles. Google ha incorporado de forma temprana, desde la versión 5 de Chrome, elementos de Html5. También lo han hecho sus compañeros en el WHATWG con las últimas versiones de Firefox, Safari y Opera. Incluso Microsoft, a menudo rezagado en materia de adopción de novedades y estándares, ha incorporado Html5 en las versiones 9 y 10 de Internet Explorer. Por su lado, los teléfonos smartphone también están adoptando el estándar Html5: Apple con el iPhone, RIM con BlackBerry, Google con Google Phone, etc. ¿Se trata de una evolución o de una revolución?

De forma paralela está en curso toda una reformulación de Html5 según las reglas y la sintaxis de XML, bajo el nombre de Xhtml5.

De toda esta agitada actividad en torno a la gestación de Html5 nos quedaremos con:

- Que ya no se hablará de páginas Web sino del concepto más amplio de aplicaciones Web.
- Que Html5 es una evolución (en efecto muy importante) de Html 4.0.
- Que Html5 ha sido concebido directamente por los navegadores más innovadores del mercado como Firefox, Safari Opera y Google. Esto garantiza una adopción rápida de los estándares de Html5.
- Que cabe esperar de Html5 grandes cambios en la elaboración de formularios.
- Que, después de una década sin novedades, Html5 responde a la necesidad real de los diseñadores de renovar la interfaz de las aplicaciones Web.

2.- Líneas maestras de HTML5.

Antes de detallar las novedades de Html5, esbochemos algunas de las líneas maestras de esta nueva versión del lenguaje Html.

- **Html5 es una evolución de Html.** Retoma los grandes principios, y los modifica para mejorar ciertos aspectos. Ha sido diseñado para asegurar una compatibilidad hacia atrás con todo lo que se ha desarrollado hasta el presente, en términos de publicación sobre la Web. A pesar de que el código es menos formal que Xhtml, la exigencia de un código propio, respetuoso con las reglas fundamentales del etiquetado, continúa existiendo.
- El principio de **separación entre el contenido y la presentación sigue siendo conveniente e incluso se ve reforzado.** El abandono de toda una serie de etiquetas y de atributos de presentación es la mejor prueba de ello. Las hojas de estilo CSS son elementos inseparables de Html5. Se espera que, aprovechando la mayor potencia que proporcionan las nuevas hojas de estilo CSS3, la interfaz de usuario se transforme por completo en los próximos años. La simplificación del código y el deseo de evitar cualquier complicación inútil son dos de los principios que han regido la elaboración de Html5.
- La caza de **plug-ins** que enturbian el código y complican la tarea de los diseñadores, **sustituídos por una integración directa en el navegador.** Las nuevas etiquetas de audio y de vídeo son un buen ejemplo.
- El **reconocimiento de JavaScript** como socio de Html5 a la hora de la publicación en la red. También de los scripts clásicos usados a la hora de verificar la introducción correcta de datos o la validez de una dirección de correo electrónico en los formularios. Html5 evita estas codificaciones, redundantes, haciendo que los navegadores incluyan de forma nativa este tipo de funciones. Otro ejemplo es el código para el diseño 2D y muy pronto 3D (etiqueta <canvas>), mucho más próximo a JavaScript que a un lenguaje etiquetado.
- Html5 se convierte, a su vez, en una plataforma de interfaces de aplicación (API) que integran funcionalidades avanzadas como la geolocalización, el almacenamiento de datos en local, la consulta en modo desconectado, etc.

3.- Las novedades de HTML5.

Las novedades de Html5 y sobre todo del trinomio conceptual "Html5 + CSS3 + aplicaciones JavaScript" son numerosas.



Html5.

- Un nuevo doctype, simplificado y unificado.
- Supresión de etiquetas y atributos de presentación.
- Nuevas etiquetas **semánticas** o de organización.
- Nuevas etiquetas de audio y de vídeo que ya no necesitan más plug-ins dedicados.
- Diseño 2D y muy pronto 3D gracias a la nueva etiqueta <canvas>.
- Profusión de formularios innovadores como, por ejemplo, cursores o calendarios, y la integración de forma nativa en los navegadores de la validación de datos.
- Etc.

Hay que señalar que la integración de Html5 no es del todo completa aún. Todavía se esperan nuevos aportes, en particular para todo lo relacionado con los formularios.

CSS3

- Nuevos selectores.
- Bordes redondeados.
- Bordes decorados.
- Tipos de letra personalizados que permitirán a la tipografía encontrar, por fin, su sitio en las interfaces Web.
- Presentación de un texto en varias columnas.
- Sombras aplicadas al texto y al resto de elementos.
- Degradados de color.



- Múltiples fondos.
- Opacidad o transparencia.
- Transformaciones.
- Transiciones (sin JavaScript).
- El diseño del texto con guiones automáticos o el tratamiento de títulos largos.
- Los elementos gráficos como los filtros, las máscaras, el efecto espejo o la fusión de imágenes.

Estas nuevas especificaciones CSS3 ya están disponibles en las versiones más recientes de los navegadores. No cabe duda de que CSS3 revoluciona y va a revolucionar la forma de diseñar y de explotar las interfaces Web en los próximos meses y años.

Las API JavaScript.

- La geolocalización permite localizar al usuario (con su permiso) mediante coordenadas de longitud y latitud.
- Las súper cookies con *Web Storage* permiten guardar una cantidad mayor de datos en el navegador.
- Posibilidad de utilizar aplicaciones Web en modo sin conexión tras cachear los recursos necesarios.
- Nuevos elementos *Workers* que permiten ejecutar tareas de fondo en paralelo con el programa JavaScript en un entorno totalmente separado de la página.
- Nuevos elementos *Websockets* que permiten establecer una comunicación bidireccional asíncrona entre el navegador y el servidor.
- Opción deslizar-soltar (*drag/drop*) de forma nativa en el navegador.
- El atributo *ContentEditable* habilita la edición en línea del contenido de un elemento. Da la apariencia de un editor WYSIWYG básico que permite de este modo editar directamente el contenido en la página. Todas las modificaciones realizadas por el usuario se podrán procesar al finalizar la edición.
- Etc.



Estas API JavaScript son potentes y permiten realizar aplicaciones innovadoras tales como la geolocalización del usuario, el almacenamiento de datos en el navegador o la consulta de aplicaciones en modo desconectado en su smartphone. Se puede constatar que para explotar toda la potencia del concepto Html5, el conocimiento de JavaScript es indispensable.

Algunas de estas funciones ya están resueltas en frameworks JavaScript como Dojo y jQuery.

4.- HTML5 es un lenguaje de etiquetas.

Digno sucesor de las versiones precedentes de Html, y de Html 4.0 en particular, Html5 sigue siendo un lenguaje de etiquetas.

4.1.- Etiquetas.

Las etiquetas, también llamadas elementos, son comandos interpretados por los navegadores e informados entre los signos menor que (<) y mayor que (>). De este modo una etiqueta se escribe <etiqueta>.

Por regla general, a toda etiqueta de apertura le corresponde una etiqueta de cierre que marca el final del comando indicado por la etiqueta de apertura. La etiqueta de cierre retoma el mismo enunciado que la etiqueta de apertura, aunque precedida por una barra oblicua (/). De este modo a la etiqueta de apertura <etiqueta> le corresponde la etiqueta de cierre </etiqueta>. La sintaxis de una etiqueta es, por tanto:

`<etiqueta> ... </etiqueta>`

Para comprender el modo de funcionamiento de las etiquetas se propone el texto siguiente:

iEs <negrita>importante</negrita> aprender el lenguaje
<cursiva>Html5</cursiva>!

Esto puede interpretarse de la siguiente manera:

1. Escribir "iEs" de forma normal, puesto que no se ha especificado nada.
2. A continuación escribir la palabra (únicamente esta palabra) "importante" y ponerla en negrita.
3. Retomar la escritura normal para "aprender el lenguaje".
4. Escribir esta vez la palabra "Html5" en cursiva.
5. Y terminar por "!" en escritura normal.

Esto daría el resultado siguiente:

iEs **importante** aprender el lenguaje *Html5*!

El lenguaje Html no es ni más ni menos que esto. Cada vez que se indica una instrucción al navegador, por ejemplo insertar un título, empezar a escribir una tabla o realizar un vínculo a otra página, se aplica una etiqueta de apertura. La etiqueta de cierre señala al navegador que la instrucción se ha terminado.

Existe no obstante una excepción, pues Html5 ha heredado de Html las etiquetas únicas, también llamadas etiquetas vacías, que no tienen etiqueta de cierre.

Por ejemplo, la etiqueta imagen .

Para asegurar la compatibilidad hacia atrás con Xhtml, estas etiquetas también pueden escribirse en Html5 con un signo de cierre.

De este modo, la etiqueta imagen también puede escribirse en Xhtml .

El espacio situado antes de la barra oblicua de cierre es importante por motivos de compatibilidad con los navegadores más antiguos.

Ejemplos

	b por <i>bold</i> que significa negrita
<i>	i por <i>italic</i> que significa cursiva
<p>	p por <i>paragraph</i> que significa párrafo
<div>	div por <i>division</i> que significa... división
<table>	<i>table</i> significa tabla
<form>	<i>form</i> significa formulario
	<i>img</i> para imagen
Etc.	

4.2. Atributos de etiqueta.

En ocasiones es preciso completar una instrucción con especificaciones más precisas en uno u otro dominio. Para hacer esto, el lenguaje Html5 dispone de los atributos de etiqueta. **El atributo se inserta en la etiqueta**, entre la palabra de instrucción y el signo > final.

La sintaxis completa de una etiqueta con un atributo es:

<etiqueta atributo="valor"> ... </etiqueta>

El atributo contiene siempre un valor, que se indica de forma complementaria al atributo mediante el signo igual (=) seguido del valor escrito entre comillas. **El respeto estricto a la sintaxis exige que no exista espacio antes o después del signo igual.**

Es posible usar varios atributos, separados por espacio, en una misma etiqueta:

<etiqueta atributo1="valor" atributo2="valor"> ... </etiqueta>

4.3. Uso correcto de las etiquetas.

He aquí algunas reglas simples que será preciso respetar a la hora de escribir etiquetas Html5.

- En Html5, **las etiquetas no son sensibles a las mayúsculas o minúsculas**. De este modo es indiferente escribir `<ETIQUETA>`, `<Etiqueta>` o `<etiqueta>`. Algunos verán en la escritura con mayúsculas un medio eficaz para distinguir el código Html del contenido del documento. No obstante, el uso generalizado escribe las etiquetas con minúsculas (como en Xhtml).
- **La regla general exige que toda etiqueta abierta `<etiqueta>` se cierre `</etiqueta>`**. La libertad que se adquirió en la escritura de Html 3.2 a causa del funcionamiento más o menos permisivo de los navegadores ya no está de actualidad. El rigor aportado por Html 4.0 estricto y Xhtml 1.0 debe seguir guiando su escritura.
- **Las etiquetas deben enlazarse correctamente**. Cuando se aplican varias etiquetas a un elemento, el orden de cierre de éstas es esencial. La primera etiqueta de cierre debe corresponderse con la última etiqueta de apertura que no haya sido cerrada. El siguiente ejemplo resultará más claro:
Es correcto: `<a><c>contenido</c>`
Es incorrecto: `<a><c>contenido</c>`
- **Los valores de los atributos deben informarse siempre entre comillas**. Aquí también sigue siendo necesario el rigor en el código.

5. Etiquetas y atributos HTML 4.0 desaparecidos.

Las diferencias, tanto a nivel de etiquetas como de atributos, respecto a Html 4.0 son numerosas e importantes.

5.1. A nivel de etiquetas.

Destacamos en primer lugar la desaparición por completo de los cuadros. Las etiquetas `<frame>`, `<frameset>` y `<noframes>` han desaparecido definitivamente de las herramientas a disposición de los diseñadores. Esto no es ninguna sorpresa puesto que estaban obsoletos desde hace varios años. Los programadores profesionales los evitaban como la peste después de que Google anunciara a los Webmasters que no garantizaba una indexación correcta de los sitios web que incluyeran cuadros. La etiqueta `<iframe>` como tal subsiste en Html5.

También se produce la desaparición de ciertas etiquetas de presentación como `<big>`, `<center>`, ``, `<strike>`, `<tt>` y `<u>`. También en este caso no hay grandes sorpresas puesto que el principio de separación entre contenido y presentación ya está bien establecido en el presente. Estas etiquetas de presentación ahora se tienen en cuenta en las hojas de estilo CSS.

5.2. A nivel de atributos.

Html5 apuesta firmemente por el principio de separación entre contenido y presentación. Lo que asombra es la cantidad de atributos afectados. En resumen, todos los atributos relativos a la alineación, la anchura, los fondos (con color o con imagen), los bordes y la numeración de listas desaparecen.

Todos estos atributos deben tenerse en cuenta en las hojas de estilo, que se convierten así en un elemento inseparable del código Html5.

De forma más detallada:

- `align` en las etiquetas `<caption>`, `<iframe>`, ``, `<input>`, `<object>`, `<legend>`, `<table>`, `<hr>`, `<div>`, `<hx>`, `<p>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` y `<tr>`.
- `alink`, `link`, `text` y `vlink` en la etiqueta `<body>`.
- `background` en la etiqueta `<body>`.
- `bgcolor` en las etiquetas `<table>`, `<tr>`, `<td>`, `<th>` y `<body>`.
- `border` en las etiquetas `<table>` y `<object>`.
- `cellpadding` y `cellspacing` en la etiqueta `<table>`.
- `frameborder` en la etiqueta `<iframe>`.
- `height` en las etiquetas `<td>` y `<th>`.

- `hspace` y `vspace` en las etiquetas `` y `<object>`.
- `marginheight` y `marginwidth` en la etiqueta `<iframe>`.
- `noshade` en la etiqueta `<hr>`.
- `nowrap` en las etiquetas `<td>` y `<th>`.
- `rules` en la etiqueta `<table>`.
- `size` en la etiqueta `<hr>`.
- `type` en las etiquetas ``, `` y ``.
- `valign` en las etiquetas `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` y `<tr>`.
- `width` en las etiquetas `<hr>`, `<table>`, `<td>`, `<th>`, `<col>`, `<colgroup>` y `<pre>`.

De este modo, ya no es posible fijar, en Html5, los bordes, la alineación, los fondos o el tamaño de las tablas sin recurrir a hojas de estilo CSS.

Con la desaparición de los atributos `color` o `bgcolor`, Html5 se define en blanco y negro.

Esta ausencia completa de atributos de presentación nos obliga a revisar nuestra estructura de aprendizaje tradicional de Html en la que se estudian por un lado las etiquetas Html y por otro las hojas de estilo CSS. Ahora, ya desde nuestros primeros pasos con Html, nos veremos obligados a incluir nociones elementales de hojas de estilo para la presentación.

6. Los navegadores de nuestro estudio.

Tanto los navegadores de oficina como los navegadores móviles han hecho grandes esfuerzos para integrar Html5 sin esperar a la finalización prevista para el cuarto trimestre de 2014. Se puede afirmar que Html5 ha sido globalmente bien integrado en los navegadores actuales. Sin embargo, esta integración todavía no es completa. Podemos citar, por ejemplo, los nuevos formularios de Html5 y algunos CSS3 experimentales.

Es importante destacar que los diferentes navegadores han cumplido con las recomendaciones de W3C. El cumplimiento de los estándares web ha alcanzado niveles sin precedentes en la historia de la web. Esto puede hacer pensar que Html5 será una importante evolución de las aplicaciones Web actuales y futuras.

Internet Explorer.

Las relaciones de Microsoft con respecto a la Web han sido siempre complejas. El paso de los años no ha cambiado nada. A pesar de todo, Internet Explorer, recordemos, el navegador por defecto de todos los sistemas operativos de Windows, sigue siendo el líder de todas las versiones de navegadores de escritorio.

Nuevo y bonito, Internet Explorer 10 es una evolución importante sobre todo en el ámbito de las hojas de estilo CSS3 y en las API JavaScript. Pero está reservado a los usuarios de Windows 8 y 7 y no para internautas que todavía utilicen Windows Vista o XP.

Hay quien dirá que queda Internet Explorer 9. Pero éste está reservado sólo a los usuarios de Windows 7 y Vista. Hay que reconocer que aunque Internet Explorer 9 asimila Html5 y una parte de las especificaciones CSS3, parece ya antiguo respecto a la competencia, particularmente en materia de hojas de estilo y de aplicaciones JavaScript.

Hay que subrayar que los excelentes navegadores de la competencia (Google Chrome y Firefox) están disponibles para todas las versiones de Windows.

En lo que respecta a Internet Explorer 8, se puede considerar como obsoleto lo que respecta a los avances del concepto Html5.

También nos podemos preguntar si la opción de publicar una nueva versión de un navegador cada dos o tres años se adapta a la ola de cambios a la que se enfrenta la web.

Google Chrome.

Aparecido en septiembre de 2008, Google Chrome no ha cesado de incrementar su cuota de mercado. Se puede considerar que es, a día de hoy, el más innovador de los navegadores disponibles en la integración de Html5 y de las hojas de estilo CSS3. No es casual que se haya convertido en el

navegador favorito de los desarrolladores de aplicaciones Web. Las versiones se suceden a un ritmo sostenido gracias a su sistema de actualización automática, el usuario siempre está seguro de tener la versión más reciente y la más puntera de las tecnologías modernas, tanto a nivel de HTML5 como CSS3 y de las API JavaScript. Se recomienda para este estudio. Para su información, se ha utilizado la versión 25 para esta actualización.

Firefox.

Muchos desarrolladores tienen la impresión de que Firefox se ha estancado y que esto repercute en su cuota de mercado entre los internautas. Naturalmente, Firefox sigue siendo un navegador excelente. A pesar de un bajada confusa en el número de versiones (estamos en la versión 19) y de un sistema de actualizaciones con un buen rendimiento, deja al usuario con la desagradable sensación de que no evoluciona a la velocidad de su competencia directa, que es Google Chrome. ¿La comunidad Mozilla tiene la suficiente envergadura para manejar sus múltiples proyectos, como la versión móvil de Firefox y un OS Firefox, para intentar competir con el sistema Android?

Safari.

Safari es el navegador de las plataformas Macintosh y de la familia Apple. En su versión 6 en este sistema operativo, es sin duda un excelente navegador que está a la última en lo que respecta a la integración de HTML5 y las hojas de estilo CSS3 y no tiene nada que envidiar a su competencia en cuanto a innovación y rendimiento. En junio de 2009 apareció una versión para Windows que, a pesar de sus evidentes cualidades, no ha seducido a los internautas. Su cuota de mercado es tan marginal que se habla de la posibilidad que no haya versión 6 para Windows. Hay muchas posibilidades de que la versión 5.1 de Safari sea la última versión para Windows.

Opera.

Se trata de un navegador desconocido y su cuota de mercado no se conoce. Es un navegador notable con numerosas cualidades y a menudo innovador en el ámbito de la integración de HTML5 y de CSS3, por ejemplo, en materia de los nuevos formularios HTML5. Afortunadamente para este agradable navegador, la versión móvil de Opera tiene un cierto éxito y se posiciona como el líder de los navegadores móviles no nativos en el universo de los smartphones.

7. El documento HTML5 mínimo.

7.1. El doctype y su importancia.

Todo documento Html debe comenzar por un doctype. Html5 propone un doctype único y simplificado.

<!DOCTYPE html>

Para darse cuenta de la situación, basta con compararlo con un doctype de Html 4.0.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

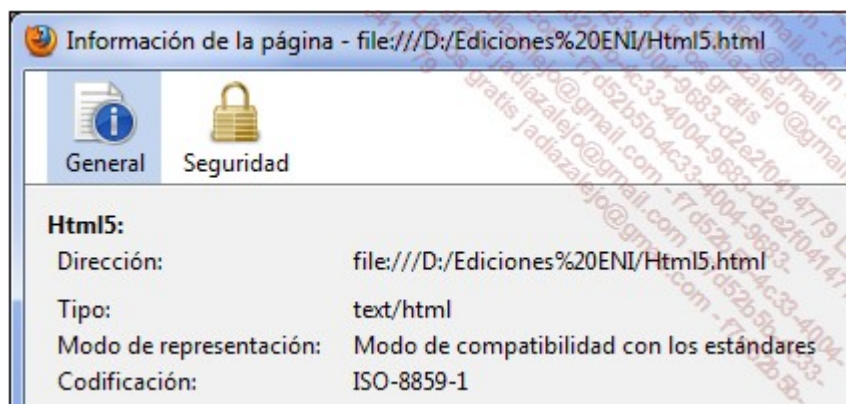
¡Claramente mucho más corto y, al fin, posible de memorizar!

El doctype, también llamado DTD, sirve para indicar al navegador a qué reglas de escritura obedece el código fuente de la página Html o Xhtml. Lo usará para mostrar la página según los estándares de W3C. Todos los navegadores aplican la misma normativa, por lo que cabe esperar una visualización idéntica en los distintos navegadores.

En ausencia de un doctype el navegador no sabe según qué reglas debe procesar la página. Se conforma con acogerse a reglas genéricas para renderizar a toda costa la página Html. Esto puede suponer diferencias respecto a la visualización en los distintos navegadores. Este modo, incompleto, se llama modo de compatibilidad o *quirks mode*.

Puede verificar fácilmente en su navegador en qué modo (quirks o compatible con los estándares) se encuentra el navegador.

En Firefox 3.6+, acceda al menú **Herramientas - Información de la página** - pestaña **General - Modo de representación**.



El doctype debe situarse en la primera línea del archivo Html. Si hubiera cualquier otra cosa, incluso un simple espacio o una línea en blanco, ciertos navegadores consideran que la página no tiene el doctype y la mostrará en quirks mode.

7.2. La etiqueta HTML.

La etiqueta `<html>` indica al navegador que se trata de un documento Html.

La etiqueta `<html>` es el elemento más alto o el elemento raíz del documento. Se declara justo debajo de la declaración del doctype.

```
<!DOCTYPE html>
<html lang="es">
...
</html>
```

La etiqueta `<html>` puede incluir el atributo `lang="es"` que especifica que el documento está en lengua española. Esta información es muy valiosa para los motores de búsqueda como Google y para los programas de síntesis de voz usados por personas con dificultades visuales.

En la dirección www.w3schools.com/tags/ref_language_codes.asp podrá encontrar una lista completa de las demás lenguas referenciadas.

7.3. El encabezado del documento.

La etiqueta `<head> ... </head>` que se posiciona justo después del doctype y de la etiqueta `<html>` contiene lo que llamamos el encabezado del documento.

En este encabezado del documento se sitúa toda una serie de información relativa al propio documento, independientemente de su contenido.

Esta información puede ser de naturaleza muy diversa:

- **El título del documento.** La etiqueta `<title> ... </title>` que se explica en el siguiente punto.
- Declaraciones o llamadas a hojas de estilo CSS.
- Funciones o vínculos a archivos JavaScript.
- Información dirigida a los motores de búsqueda para mejorar la indexación y el posicionamiento de la página.
- Información dirigida a los navegadores.
- Meta-etiquetas que retoman la codificación, la descripción de la página, las palabras clave asociadas a la página, el nombre del autor, menciones de copyright, etc.

Con intención de simplificar, Html5 especifica valores por defecto para el atributo tipo de estos scripts, estilos y elementos `link`. Salvo si necesita un valor diferente al definido por defecto, podrá omitir en Html5 este atributo *type*.

JavaScript es el lenguaje de script predefinido.

Así, en Html 4.0, era preciso escribir:

```
<script type="text/javascript">
...
</script>
```

Y

```
<script type="text/javascript" src="archivo.js"></script>
```

En Html5, basta con escribir:

```
<script>
...
</script>
```

Y

```
<script src="archivo"></script>
```

CSS es ahora el lenguaje predefinido para las hojas de estilo.

Así, en Html 4.0, era preciso escribir:

```
<style type="text/css">
...
</style>
```

Y

```
<link rel="stylesheet" type="text/css" href="archivo.css">
```

En Html5, basta con escribir:

```
<style>
...
</style>
```

Y

```
<link rel="stylesheet" href="archivo.css">
```

Nuestro documento Html5 mínimo queda así:

```
<!DOCTYPE html>
<html lang="es">
<head>
...
</head>
...
</html>
```

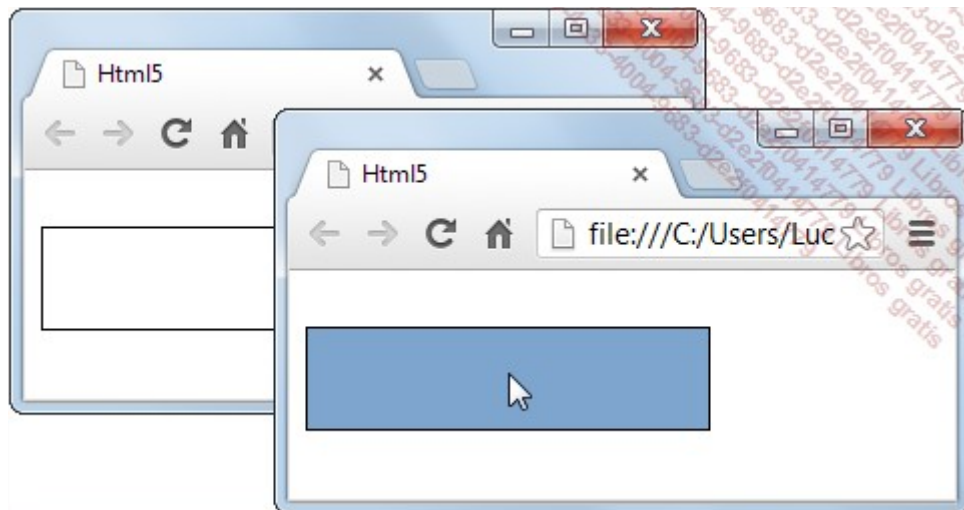
Dejando aparte el título de la página (véase a continuación), ningún elemento comprendido entre las etiquetas `<head> ... </head>` se muestra en el navegador.

8. El título del documento.

La etiqueta `<title>`, incluso en el encabezado del documento, es la única etiqueta obligatoria del mismo.

La etiqueta `<title> ... </title>` le asigna un título a su página.

El título se visualiza en la barra de título, situada en la parte superior izquierda de cualquier navegador y/o en las pestañas abiertas por el mismo.



La etiqueta `<title>` tiene especial importancia en lo que respecta al algoritmo de posicionamiento de Google. Velaremos por tener, para los documentos Web, un título atractivo y sintético.

El documento Html5 se completa de la siguiente forma:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
</head>
...
</html>
```

Si accidentalmente hubiera un error en las etiquetas `<title> ... </title>`, se mostraría una simple página en blanco.

9. La codificación (charset) del documento.

La noción del juego de caracteres usado (*charset*) es una noción esencial en el desarrollo de páginas Web. Este concepto está no obstante plagado de trampas, principalmente en términos de interoperabilidad. ¿Ha recibido alguna vez un correo electrónico en el que algunos caracteres habían sido reemplazados por otros símbolos cabalísticos como o vÃ©rificaciÃ³n?

El juego de caracteres determina la forma en que los caracteres de un alfabeto se convierten en bytes en un archivo informático (y viceversa). Ciertos métodos de codificación son específicos de un entorno informático en un idioma o alfabeto dados, otros son multiplataforma y multiidioma. Ejemplos de codificación: ASCII, ISO-8859-1, ISO-8859-15, UTF-8, etc.

ASCII	Juego de caracteres básico, aunque relativamente antiguo, que permite el intercambio de textos en inglés a nivel mundial. Se escogió en el origen de la web pero como el inglés no tiene acentos, había que codificar los caracteres acentuados y especiales con entidades del tipo ´ (para la é) o € (para el €).
ISO-8859-1	Extensión de ASCII, ISO-8859-1, también llamado Latin-1 o Europa occidental, es la codificación que se utiliza con frecuencia en nuestra zona mundial. Añade numerosos caracteres latinos así como otros signos de puntuación o símbolos. Actualmente, su utilización tiende a decrecer en favor de Unicode
ISO-8859-15	Versión actualizada de ISO-8859-1 que incluye, entre otros, el símbolo €.
UTF-8	UTF-8 es un formato de codificación de caracteres (charset) creado para codificar el conjunto de caracteres internacionales de Unicode de diferentes idiomas, a diferencia de los formatos más antiguos como iso-8859-1 que requiere una forma de codificación para aceptar, por ejemplo, letras

	acentuadas o caracteres especiales. Su principal ventaja para la web es que permite representar los miles de caracteres de los diferentes idiomas y permite crear sitios Web que puedan ser multilingües (español, chino o árabe...) sin tener que crear múltiples versiones del sitio según la codificación de cada país.
--	--

Desde finales de 2011, el W3C, siguiendo en esto a la IETF (*Internet Engineering Task Force*) que edita los estándares de Internet, **recomienda el charset UTF-8 para la edición de páginas Web** en gran medida por su universalidad.

Los navegadores incluyen una función de detección automática del juego de caracteres. Aun así, **dejar la codificación al criterio del navegador supone en ocasiones una puerta abierta a errores de interpretación**. Por ello es importante definir el charset que se utiliza en el encabezado del documento.

```
<meta charset="UTF-8">
```

La siguiente sintaxis, que recuerda a Xhtml 1.0, también está aceptada:

```
<meta charset="UTF-8" />
```

De este modo no sólo se ha simplificado el doctype en Html5. Recordemos la forma que tomaba la declaración de la codificación en Html 4.0:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

Nuestro documento Html5 mínimo es, ahora:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
...
</html>
```

La especificación Html5 recomienda (lógicamente) a los autores el uso de UTF-8. Pero nada impide mantener el formato iso-8859-1 o iso-8859-15 más europeo. No obstante se arriesga a parecer algo anticuado...

10. El cuerpo del documento.

Las etiquetas `<body> ... </body>` definen lo que llamamos el cuerpo del documento. Es entre estas etiquetas donde se situará el código Html5 que se usará para elaborar el contenido de la página.

Esta parte del documento Html se visualizará en la ventana del navegador y por tanto la que se presentará al internauta.

En Html5, los atributos dedicados a la presentación del cuerpo del documento en la etiqueta `<body>` se abandonan dejando paso al uso de hojas de estilo.

Estos atributos que determinaban el color de fondo (`bgcolor`), la imagen de fondo (`background`) o el color del texto (`text`) ya no tienen lugar en Html5.

Para finalizar, nuestro documento Html5 mínimo tiene el siguiente aspecto:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
</body>
</html>
```

11. Modo de proceder.

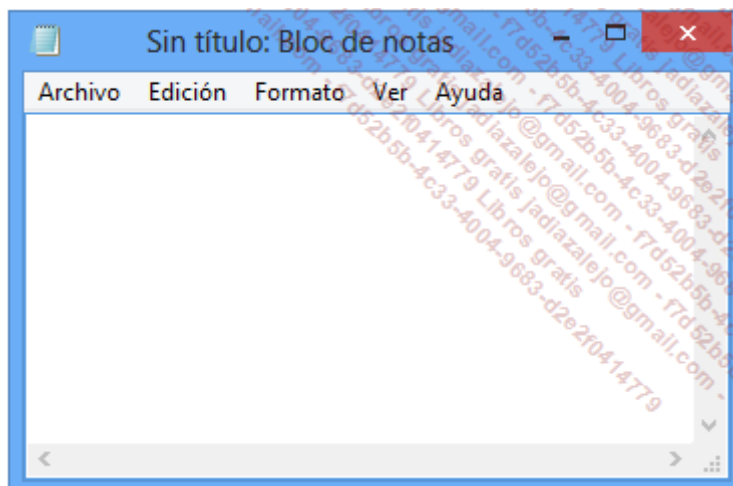
Los lenguajes usados en la Web tienen la ventaja de que no necesitan una herramienta de desarrollo dedicada. Para escribir el código fuente, no hace falta un programa caro! Con un sencillo editor de texto, incluido en todos los sistemas operativos, tendremos suficiente.

Lo fundamental es tener el texto en bruto sin ningún tipo de formato.

En Windows, puede usar el Bloc de notas o Notepad. Recordamos que se accede a través del menú de Inicio en **Todos los programas - Accesorios - Bloc de notas**.

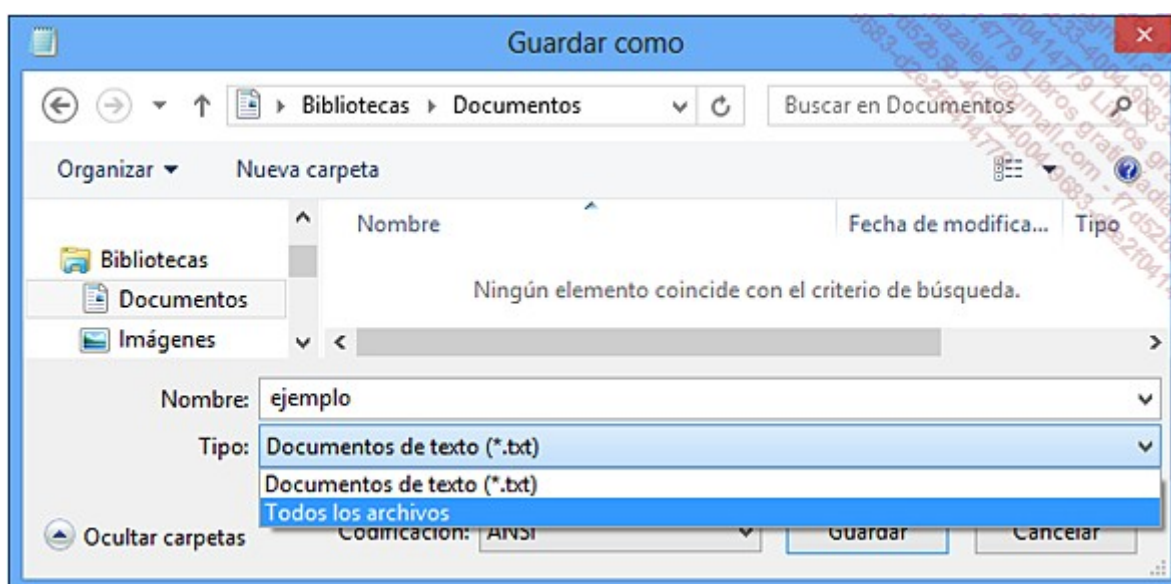
El equivalente del bloc de notas para los Macintosh es TextEdit. Hay numerosos editores de texto gratuitos como Notepad ++ (<http://notepad-plus-plus.org>) para Windows y para Mac, textWrangler (www.barebones.com/products/textwrangler/).

Si su editor de texto le propone diferentes opciones para el formato UTF-8, utilice "UTF sin BOM" o "UTF-8" without BOM".



Este editor de texto, algo rudimentario, es no obstante "perfecto" para codificar texto bruto (sin ningún tipo de formato).

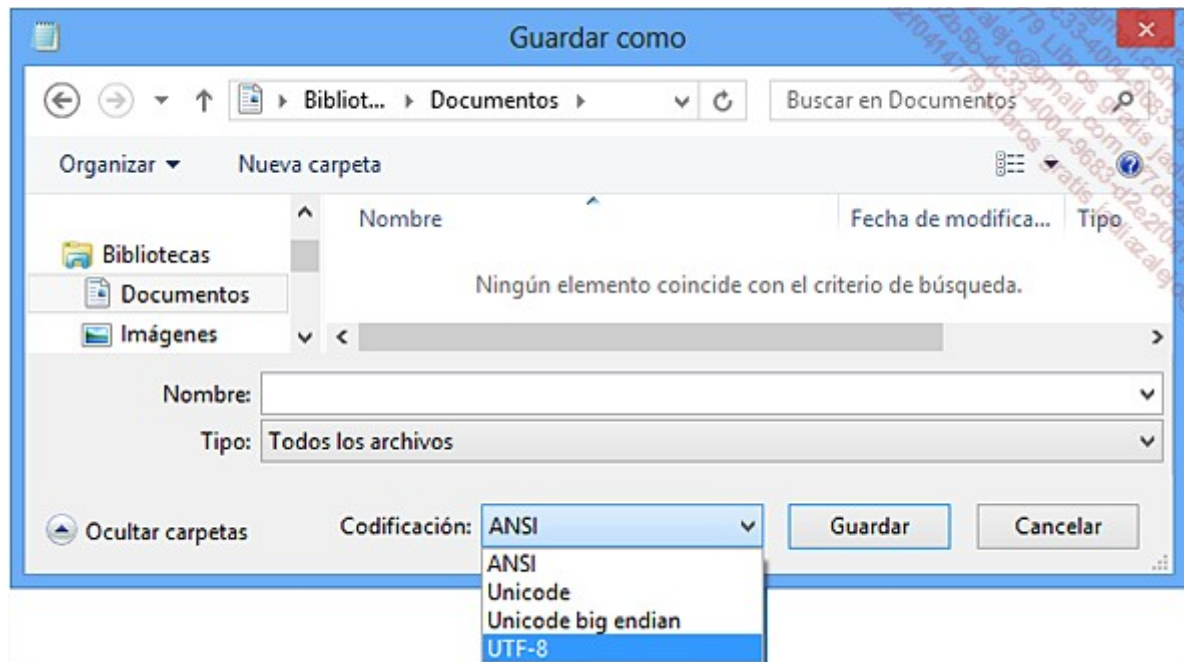
Es importante respetar un procedimiento correcto a la hora de guardar los archivos. Tras haber hecho **Archivo - Guardar como...**, tenga precaución de activar en la zona **Tipo** la opción **Todos los archivos**.



En el marco de este libro dedicado a HTML5, los archivos tendrán como extensión *.htm o *.html. El uso de la extensión *.htm parece estar generalizándose.

IMPORTANTE!!!!

Antes de salir del editor de texto, hay que tratar la codificación. De hecho, no sirve de nada especificar un charset en UTF-8, si el documento no está codificado en UTF-8. En la zona **Codificación**, ponga la opción **UTF-8**.



Para visualizar su archivo, basta con abrir su navegador e indicar la ruta del archivo concreto.

Esta operación parece muy sencilla. No obstante, con la tendencia de una interfaz más y más simplificada de los navegadores, no es del todo intuitiva. En Firefox, Safari y Opera, hay que activar la barra de menús, encontrar el archivo con **Archivo - Abrir** un archivo. En Internet Explorer 9 y Google Chrome, donde la interfaz es todavía más reducida, no hay más alternativa que usar el atajo de teclado [Ctrl] O.

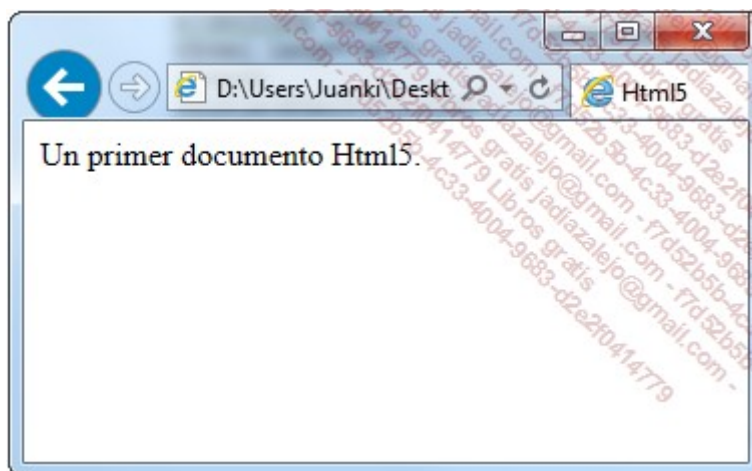
En caso de realizar modificaciones al código fuente inicial, no olvide actualizar la página para que el navegador tenga en cuenta los cambios.

Ejemplo

Llegados a este paso, contamos con todas las herramientas para crear nuestro primer documento Html5.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="iso-8859-1">
</head>
<body>
    Un primer documento Html5.
</body>
</html>
```

Que se muestra en Internet Explorer 10:



12. Validación de código HTML5.

Si bien Html5 es menos formalista que Html 4.0 estricto o que Xhtml 1.0, sigue siendo indispensable validar el código fuente.

Las ventajas de un código fuente perfecto y por tanto de la validación son:

- Comprensión y verificación en profundidad de Html5 por parte de programadores iniciados (y avanzados).
- Asegurar que la página se mostrará siempre de forma correcta en la mayoría de los navegadores. Una página inválida puede llevar a dos navegadores distintos a interpretarla de forma muy diferente.
- Demostración de su propia capacidad y profesionalidad a la hora de realizar un código de calidad, conforme a los estándares de Html5.
- Los programas de síntesis vocal y otras ayudas técnicas dirigidas a personas con problemas de accesibilidad se basan en un código válido a la hora de procesar las páginas de la red.

Si el documento, sometido a un validador, no es correcto, se le proveerá un análisis de los problemas que le permitirá corregir su código. Las primeras validaciones pueden ser muy frustrantes pero suponen un excelente aprendizaje del código fuente. No obstante el informe de errores es, con frecuencia, muy técnico.

Para realizar su función de verificación y validación, los validadores precisan que se defina el doctype del documento (véase la sección El documento Html5 mínimo - La codificación (charset) del documento en este capítulo).

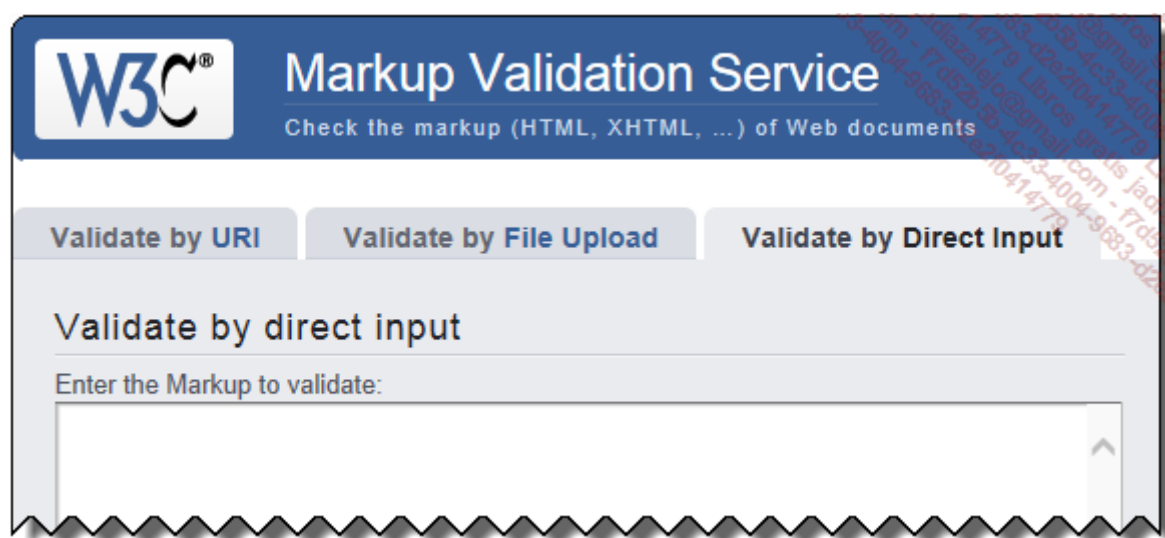
Los validadores Html5 (en línea) disponibles son:

- El propio validador de W3C.
- El validador validator.nu.

12.1. El validador de W3C.

W3C presenta un validador de código (validator.w3c.org). Lo cual es garantía de calidad.

No obstante la validación de Html5 se anuncia todavía en un estado experimental.



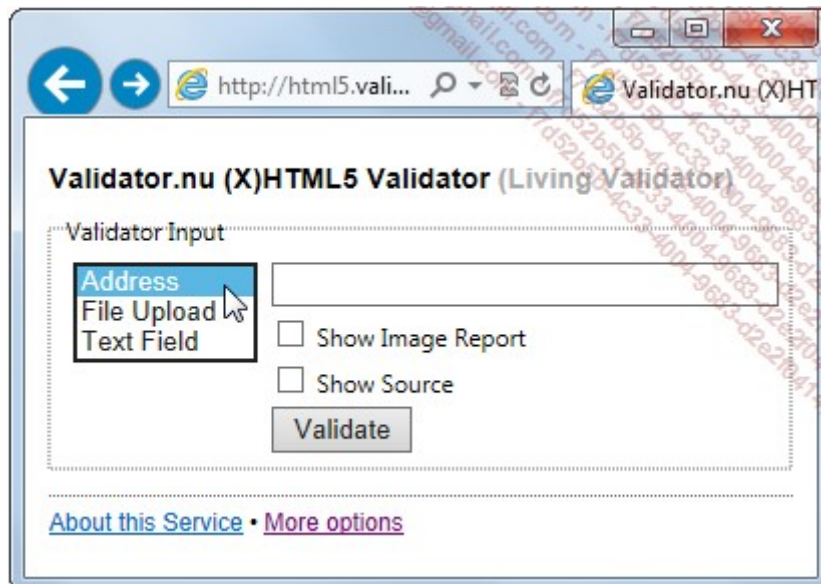
Destaquemos:

- Que está en inglés.
- Que ofrece la validación de un archivo a partir de su dirección http, un archivo que esté alojado en su ordenador o código que insertemos en una zona de texto.
- Que es muy potente.
- Que sus notas y explicaciones son muy técnicas o, a veces, incluso sibilinas.
- Que es "la" referencia de los profesionales.

En la validación de su documento HTML5 solo con la etiqueta de charset `<meta charset="utf-8">`, el validador de W3C (todavía en desarrollo) retorna dos avisos relativos al charset. Si le molestan, es posible evitarlos añadiendo en el encabezado, la etiqueta `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`.

12.2. El validador validator.nu

El sitio Web validator.nu proporciona, también en estado experimental, un validador de Html5 (html5.validator.nu/).



Destaquemos que fue el primero en proporcionar un validador de Html5.

Podemos subrayar:

- Que está en inglés.
- Que su interfaz es muy sencilla.
- Que ofrece la posibilidad de validar un archivo a partir de su dirección http, un archivo que esté alojado en su ordenador o código que insertemos en una zona de texto.
- Que parece de excelente calidad.

Para los amantes de las extensiones de Firefox, el módulo *Html5 Validator* le permite, tras su instalación, validar su propia página a través del menú contextual de validator.nu.

13. Prueba.

No nos hemos resistido a la tentación de hacer una prueba a partir de un código html5 erróneo.

El código erróneo es:

```
1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4. <title>Html5</title>
5. <meta charset="UTF-8">
6. </head>
7. <body>
8. <p align="center">
9. <s>Html5</s>
10. </p>
11. <table width="200px">
12. <tr>
13. <td>1</td><td>2</td>
14. </tr>
15. <tr>
16. <td>3</td><td>4</td>
17. </tr>
18. </table>
19. </body>
20. </html>
```

Los errores están:

- En la línea 8, donde el uso del atributo align ha sido suprimido (obsoleto) en Html5.
- En la línea 9, donde el uso de la etiqueta de subrayado <s> ya no existe en Html5.
- En la línea 11, donde el uso del atributo width en la etiqueta <table> ha sido suprimido en Html5.

El validador de W3C detecta sin ninguna dificultad estos errores.

Line 8, Column 18: <p align="center">	The align attribute on the p element is obsolete. Use CSS instead.
Line 9, Column 3: <s>Html5</s>	The s element is obsolete. Use CSS instead.
Line 11, Column 21: <table width="200px">	The width attribute on the table is obsolete. Use CSS instead.

La visualización de este código erróneo nos proporciona una buena justificación de la utilidad de una validación del código. En efecto, todos los navegadores de nuestro estudio muestran el término Html5 tachado cuando lo que habíamos previsto era un subrayado gracias a la etiqueta <s>.

