

Unidad Didáctica 2

Introducción a XHTML 1.0

1. Introducción.....	3
1.1. ¿Qué es HTML?.....	3
1.2. Breve historia de HTML.....	3
1.3. Especificación oficial.....	4
1.4. HTML y XHTML.....	4
1.5. HTML y CSS.....	5
2. Características básicas.....	6
2.1. Lenguajes de etiquetas.....	6
2.2. Estructura de un documento HTML.....	7
2.3. Etiquetas y atributos.....	8
2.4. Elementos HTML.....	11
2.5. Sintaxis de las etiquetas XHTML.....	12
3. Texto.....	15
3.1. Estructurar.....	15
3.2. Marcado básico de texto.....	16
3.3. Marcado avanzado de texto.....	18
3.4. Marcado genérico de texto.....	20
3.5. Espacios en blanco y nuevas líneas.....	21
3.6. Codificación de caracteres.....	23
4. Enlaces.....	26
4.1. URL.....	26
4.2. Enlaces relativos y absolutos.....	28
4.3. La etiqueta <a>.....	31
4.5. Otros tipos de enlaces.....	36
5. Metainformación.....	38
5.1. Estructura de la cabecera.....	38
5.2. Metadatos.....	39
5.3. DOCTYPE.....	41
6. Comentarios.....	42

7. Validación.....	42
------------------------------------	--------------------

1. Introducción

1.1. ¿Qué es HTML?

El propio **W3C** define el lenguaje HTML como *"un lenguaje reconocido universalmente y que permite publicar información de forma global"*. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica.

1.2. Breve historia de HTML

El origen de HTML se remonta a 1980, cuando el físico **Tim Berners-Lee**, trabajador del [CERN](http://www.cern.ch/) (*Organización Europea para la Investigación Nuclear*) propuso un nuevo sistema de *"hipertexto"* para compartir documentos.

Tras finalizar el desarrollo de su sistema de *"hipertexto"*, Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de *"hipertexto"* para Internet. Después de unir sus fuerzas con el ingeniero de sistemas **Robert Cailliau**, presentaron la propuesta ganadora llamada *WorldWideWeb (W3)*.

El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre *"HTMLTags"*

(<http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>) (*Etiquetas HTML*) y todavía hoy puede ser consultado online a modo de *reliquia informática*.

La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993 por parte del organismo [IETF](http://www.ietf.org/) (*Internet Engineering Task Force*). Aunque se consiguieron avances significativos (en esta época se definieron las etiquetas para imágenes, tablas y formularios) ninguna de las dos propuestas de estándar, llamadas HTML y HTML+ consiguieron convertirse en estándar oficial.

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar, el 22 de septiembre de ese mismo año, el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML.

A partir de 1996, los estándares de HTML los publica otro organismo de estandarización llamado [W3C](http://www.w3.org/) (*World Wide Web Consortium*). La versión HTML 3.2 se publicó el 14 de Enero de 1997 y es la primera recomendación de HTML publicada por el W3C. Esta revisión incorpora los últimos avances de las páginas web desarrolladas hasta 1996, como *applets* de Java y texto que fluye alrededor de las imágenes.

HTML 4.0 se publicó el 24 de Abril de 1998 (siendo una versión corregida de la publicación original del 18 de Diciembre de 1997) y supone un gran salto desde las versiones anteriores. Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o *scripts* en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios.

La última especificación oficial de HTML se publicó el 24 de diciembre de 1999 y se denomina HTML 4.01. Se trata de una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

Desde la publicación de HTML 4.01, la actividad de estandarización de HTML se detuvo y el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada [WHATWG](http://www.whatwg.org/) (<http://www.whatwg.org/>) (*Web Hypertext Application Technology Working Group*).

La actividad actual del WHATWG se centra en el estándar HTML 5.2.

De forma paralela a su actividad con HTML, W3C ha continuado con la estandarización de XHTML, una versión avanzada de HTML y basada en XML.

XHTML 1.0 es una adaptación de HTML 4.01 al lenguaje XML, por lo que mantiene casi todas sus etiquetas y características, pero añade algunas restricciones y elementos propios de XML.

1.3. Especificación oficial

El organismo [W3C](http://www.w3.org/) (<http://www.w3.org/>) (*World Wide Web Consortium*) elabora las normas que deben seguir los diseñadores de páginas web para crear las páginas HTML. Las normas oficiales están escritas en inglés y se pueden consultar de forma gratuita en las siguientes direcciones:

- [Especificación oficial de XHTML 1.0](http://www.w3.org/TR/xhtml1/) (<http://www.w3.org/TR/xhtml1/>)
- [Especificación HTML 5](https://www.w3.org/TR/html/) <https://www.w3.org/TR/html/>

1.4. HTML y XHTML

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML.

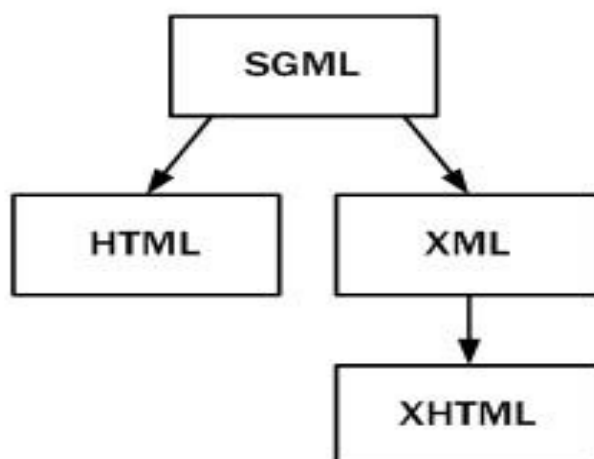


Figura 1.1. Esquema de la evolución de HTML y XHTML

1.5. HTML y CSS

Originalmente, las páginas HTML sólo incluían información sobre sus contenidos de texto e imágenes. Con el desarrollo del estándar HTML, las páginas empezaron a incluir también información sobre el aspecto de sus contenidos: tipos de letra, colores y márgenes.

La posterior aparición de tecnologías como JavaScript, provocaron que las páginas HTML también incluyeran el código de las aplicaciones (llamadas *scripts*) que se utilizan para crear páginas web dinámicas.

Incluir en una misma página HTML los contenidos, el diseño y la programación complica en exceso su mantenimiento. Normalmente, los contenidos y el diseño de la página web son responsabilidad de diferentes personas, por lo que es conveniente separarlos.

CSS es el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos. Una ventaja añadida de la separación de los contenidos y su presentación es que los documentos XHTML creados son más flexibles, ya que se adaptan mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, impresoras y dispositivos utilizados por personas discapacitadas.

2.Características básicas

2.1. Lenguajes de etiquetas

El proceso de transformación de caracteres en secuencias de números se denomina **codificación de caracteres** y cada una de las tablas que se han definido para realizar la transformación se conocen con el nombre de **páginas de código**. Una de las codificaciones más conocidas (y una de las primeras que se publicaron) es la codificación ASCII.

Una vez resuelto el problema de almacenar el texto simple, se presenta el reto de almacenar los contenidos de texto con formato. En otras palabras, ¿cómo se almacena un texto en negrita? ¿y un texto de color rojo? ¿y otro texto azul, en negrita y subrayado?

La solución que realmente se emplea para guardar la información con formato es mucho más sencilla: el archivo electrónico almacena tanto los contenidos como la información sobre el formato de esos contenidos. Si por ejemplo se quiere dividir el texto en párrafos y se desea dar especial importancia a algunas palabras, se podría indicar de la siguiente manera:

```
<parrafo> Contenido de texto con <importante>algunas  
palabras</importante> resaltadas de forma especial. </parrafo>
```

El proceso de indicar las diferentes partes que componen la información se denomina **marcar** (*markup* en inglés). Cada una de las palabras que se emplean para marcar el inicio y el final de una sección se denominan **etiquetas**.

Aunque existen algunas excepciones, en general las etiquetas se indican por pares y se forman de la siguiente manera:

- Etiqueta de apertura: carácter <, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >
- Etiqueta de cierre: carácter <, seguido del carácter /, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >

2.2. Estructura de un documento HTML

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.

El cuerpo contiene todo lo que el usuario ve en su pantalla y la cabecera contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

A continuación se muestra el código HTML de una página web muy sencilla:

```
<html>
  <head>
    <title>Título documento HTML</title>
    <!--codificación de caracteres -->
    <!-- resto del contenido de la cabecera -->
  </head>
  <body>
    <!-- contenido del cuerpo -->
    <p>El lenguaje HTML es <strong>tan sencillo</strong> que
    prácticamente se entiende sin estudiar el significado de sus etiquetas
    principales.
    </p>
  </body>
</html>
```

Volviendo al código HTML del primer ejemplo, es importante conocer las tres etiquetas principales de un documento HTML (<html>, <head>, <body>):

- <html>: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta <html>(con una sola excepción que se verá más adelante). En el interior de la etiqueta <html>se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta <html> se ignora.
- <head>: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta <title>, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte

superior izquierda de la ventana del navegador .

- `<body>`: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` no contiene más que unas pocas.

2.3. Etiquetas y atributos

HTML define 91 etiquetas que los diseñadores pueden utilizar para *marcar* los diferentes elementos que componen una página:

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, isindex, kbd, label, legend, li, link, map, menu, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var.

De todas las etiquetas disponibles, las siguientes se consideran **obsoletas** y no se pueden utilizar: applet, basefont, center, dir, font, isindex, menu, s, strike, u.

A pesar de que se trata de un número de etiquetas muy grande, no es suficiente para crear páginas complejas. Algunos elementos como las imágenes y los enlaces requieren cierta información adicional para estar completamente definidos.

La etiqueta `<a>` por ejemplo se emplea para incluir un enlace en una página. Utilizando sólo la etiqueta `<a>` no es posible establecer la dirección a la que apunta cada enlace. Como no es viable crear una etiqueta por cada enlace diferente, la solución consiste en personalizar las etiquetas HTML mediante cierta información adicional llamada **atributos**.

No todos los atributos se pueden utilizar en todas las etiquetas. Por ello, cada etiqueta define su propia lista de atributos disponibles. Además, cada atributo también indica el tipo de valor que se le puede asignar. Si el valor de un atributo no es válido, el navegador ignora ese atributo.

Aunque cada una de las etiquetas HTML define sus propios atributos, algunos de los atributos son comunes a muchas o casi todas las etiquetas. De esta forma, es habitual explicar por separado los atributos comunes de las etiquetas para no tener que volver a hacerlo cada vez que se explica una nueva etiqueta.

Los atributos comunes se dividen en cuatro grupos según su funcionalidad:

1) **Atributos básicos:** se pueden utilizar prácticamente en todas las etiquetas HTML. Estos atributos sólo son realmente útiles cuando se trabaja con CSS y con Javascript.

Atributo	Descripción
id = "texto"	Establece un identificador único a cada elemento dentro de una página HTML
class = "texto"	Establece la clase CSS que se aplica a los estilos del elemento
style = "texto"	Establece de forma directa los estilos CSS de un elemento
title = "texto"	Establece el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

Respecto al valor de los atributos id y class, sólo pueden contener guiones medios (-), guiones bajos (_), letras y/o números, pero no pueden empezar por números. Además, los navegadores distinguen mayúsculas de minúsculas y no se recomienda utilizar letras como ñ y acentos, ya que no es seguro que funcionen correctamente en todas las versiones de todos los navegadores.

2) **Atributos para internacionalización:** los utilizan las páginas que muestran sus contenidos en varios idiomas o aquellas que quieren indicar de forma explícita el idioma de sus contenidos:

Atributo	Descripción
lang = "codigo de idioma"	Indica el idioma del elemento mediante un código predefinido
xml:lang = "codigo de idioma"	Indica el idioma del elemento mediante un código predefinido
dir	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

En las páginas XHTML, el atributo xml:lang tiene más prioridad que lang y es obligatorio incluirlo siempre que se incluye el atributo lang.

Como la palabra internacionalización es muy larga, se suele sustituir por la abreviatura i18n (el número 18 se refiere al número de letras que existen entre la letra i y la letra n de la palabra internacionalización).

3) **Atributos de eventos:** sólo se utilizan en las páginas web dinámicas creadas con JavaScript.

Atributo	Descripción
onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup	Permiten controlar los eventos producidos sobre cada elemento de la página

Atributo	Descripción
accesskey = "letra"	Establece una tecla de acceso rápido a un elemento HTML
tabindex = "numero"	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32.767
onfocus, onblur	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

Cada vez que el usuario pulsa una tecla, mueve su ratón o pulsa cualquier botón del ratón, se produce un evento dentro del navegador. Utilizando JavaScript y los atributos anteriores, es posible responder de forma adecuada a cada evento.

4) **Atributos para los elementos que pueden obtener el foco:**

Los elementos de las páginas web también pueden obtener el foco de la aplicación (en este caso, el foco del navegador) y HTML define algunos atributos específicos para controlar cómo se seleccionan los elementos.

Cuando se pulsa repetidamente la tecla del tabulador sobre una página web, el navegador selecciona de forma alternativa todos los elementos de la página que se pueden seleccionar (principalmente los enlaces y los elementos de formulario). El atributo tabindex permite alterar el orden en el que se seleccionan los elementos, por lo que es muy útil cuando se quiere controlar de forma precisa cómo se seleccionan los campos de un formulario complejo.

Por su parte, el atributo accesskey permite establecer una tecla para acceder de forma rápida a cualquier elemento. Aunque la tecla de acceso rápido se establece mediante HTML, la combinación de teclas necesarias para activar ese acceso rápido depende del navegador. En el navegador Internet Explorer se pulsa la tecla ALT + la tecla definida; en el navegador Firefox se pulsa Alt + Shift + la tecla definida; en el navegador Opera se pulsa Shift + Esc + la tecla definida; en el navegador Safari se pulsa Ctrl+ la tecla definida.

2.4. Elementos HTML

Además de etiquetas y atributos, HTML define el término **elemento** para referirse a las partes que componen los documentos HTML.

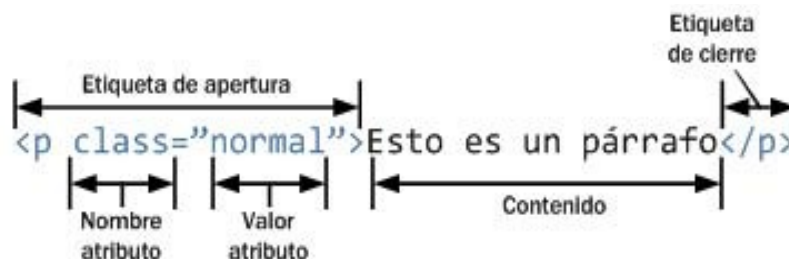


Figura 2.1. Esquema de las partes que componen un elemento HTML

Aunque en ocasiones se habla de forma indistinta de "elementos" y "etiquetas", en realidad un elemento HTML es mucho más que una etiqueta, ya que está formado por:

- Una etiqueta de apertura.
- Cero o más atributos.
- Texto encerrado por la etiqueta.
- Una etiqueta de cierre.

El texto encerrado por la etiqueta es opcional, ya que algunas etiquetas de HTML no pueden encerrar ningún texto.

Por otra parte, el lenguaje HTML clasifica a todos los elementos en dos grupos: elementos **en línea** (*inline elements*) y elementos de **bloque** (*block elements*).

La principal diferencia entre los dos tipos de elementos es la forma en la que ocupan el espacio disponible en la página.

- Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea.
- Por su parte, los elementos en línea sólo ocupan el espacio necesario para mostrar sus contenidos.

Si se considera el siguiente ejemplo:

```
<html>
  <head>
    <title>Ejemplo de elementos en línea y elementos de bloque </title>
  </head>
  <body>
    <p>Los párrafos son elementos de bloque.</p>
    <a href="http://www.google.com">Los enlaces son elementos en
    línea</a>
    <p>Dentro de un párrafo, <a href="http://www.google.com">los
    enlaces</a> siguen siendo elementos en línea.</p>
  </body>
</html>
```

La mayoría de elementos de bloque pueden contener en su interior elementos en línea y otros elementos de bloque. Los elementos en línea sólo pueden contener texto u otros elementos en línea.

Los elementos en línea definidos por HTML son: a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Los elementos de bloque definidos por HTML son: address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul.

Los siguientes elementos también se considera que son de bloque: dd, dt, frame-set, li, tbody, td, tfoot, th, thead, tr.

Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: button, del, iframe, ins, map, object, script.

2.5. Sintaxis de las etiquetas XHTML

A continuación se muestran las cinco restricciones básicas que introduce XHTML respecto a HTML en la sintaxis de sus etiquetas:

1. Las etiquetas se tienen que cerrar de acuerdo a como se abren:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a>un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML 4.01):

```
<p>Este es un párrafo con <a>un enlace</p></a>
```

2. Los nombres de las etiquetas y atributos siempre se escriben en minúsculas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<P>Este es un párrafo con <A HREF="http://www.google.com">un enlace</A></P>
```

3. El valor de los atributos siempre se encierra con comillas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a href=http://www.google.com>un enlace</a></p>
```

4. Los atributos no se pueden comprimir:

Ejemplo correcto en XHTML:

```
<dl compact="compact">...</dl>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<dl compact>...</dl>
```

Este tipo de atributos en los que el nombre coincide con su valor no son muy habituales.

5. Todas las etiquetas deben cerrarse siempre:

La mayoría de etiquetas HTML encierran un contenido de texto entre la etiqueta de apertura y la etiqueta de cierre. Sin embargo, algunas etiquetas especiales llamadas "*etiquetas vacías*" no necesitan encerrar ningún texto.

La etiqueta `
` por ejemplo, se utiliza para indicar el comienzo de una nueva línea.,. Por sus características, la etiqueta `
` nunca encierra ningún contenido de texto. En lugar de abrir y cerrar de forma consecutiva la etiqueta (`
</br>`) se puede utilizar la sintaxis `
`.

Ejemplo correcto en XHTML:

```
<br></br>
```

```
<br/>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

Además de estas cinco restricciones básicas, XHTML incluye otros cambios más avanzados respecto a HTML:

- 1 Antes de acceder al valor de un atributo, se eliminan todos los espacios en blanco que se encuentran antes y después del valor. Además, se eliminan todos los espacios en blanco sobrantes dentro del valor de un atributo.
- 2 El código JavaScript debe encerrarse entre unas etiquetas especiales (<![CDATA[y]]>) para evitar que el navegador interprete de forma errónea caracteres como & y <.
- 3 Las páginas XHTML deben prescindir del atributo name para identificar de forma única a los elementos. En su lugar, siempre debe utilizarse el atributo id.

3. Texto

La mayor parte del contenido de las páginas HTML habituales está formado por texto, llegando a ser más del 90% del código de la página. Por lo que es importante conocer los elementos y etiquetas que define HTML para el manejo del texto.

HTML define etiquetas para **estructurar** el contenido en secciones y párrafos y define otras etiquetas para **marcar** elementos importantes dentro del texto.

3.1. Estructurar

3.1.1. Párrafos

La etiqueta `<p>`, permite definir los párrafos que forman el texto de una página.

La siguiente tabla recoge la definición formal de la etiqueta `<p>`:

<code><p></code>	Párrafos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	Bloque
Descripción	Delimita el contenido de un párrafo de texto

3.1.2. Secciones

Es habitual que las páginas se dividan en diferentes secciones jerárquicas. Los títulos de sección se utilizan para delimitar el comienzo de cada sección de la página.

HTML permite crear secciones de hasta seis niveles de importancia. De esta forma, aunque una página puede definir cualquier número de secciones, sólo puede incluir seis niveles jerárquicos.

Las etiquetas que definen los títulos de sección son `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`. La etiqueta `<h1>` es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. La importancia del resto de etiquetas es descendiente, de forma que la etiqueta `<h6>` es la que se utiliza para delimitar las secciones menos importantes de la página.

A continuación se muestra la definición formal de la etiqueta `<h1>`, siendo idéntica la definición del resto de etiquetas referidas a los títulos de sección:

<h1>	Sección (titular) de nivel 1
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	Bloque
Descripción	Define los títulos de las secciones de mayor importancia de la página.

Las etiquetas <h1>, ..., <h6> definen títulos de sección, no secciones completas. Por este motivo, no es necesario encerrar los contenidos de una sección con su etiqueta correspondiente. Solamente se debe encerrar con las etiquetas <h1>, ..., <h6> los títulos de cada sección.

Evidentemente, el aspecto que los navegadores aplican por defecto a los títulos de sección se puede modificar utilizando las hojas de estilos de CSS.

3.2. Marcado básico de texto

HTML proporciona varias etiquetas para *marcar* cada uno de los diferentes tipos de texto.

- ◆ Etiquetas **** (cursiva) y **** (negrita). (vigentes en HTML 5)

	Énfasis
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	En línea
Descripción	Realza la importancia del texto que encierra

	Énfasis más acentuado
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	En línea
Descripción	Realza con la máxima importancia el texto que encierra

HTML permite indicar de forma clara el texto que ha sido eliminado y el texto que ha sido añadido a un determinado texto original.

- ♦ Las etiquetas utilizadas son **<ins>** (subrayar) y **** (tachar): (vigentes en HTML 5)

<ins>	Inserción
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> ▪ <code>cite = "url"</code> - Indica la URL de la página en la que se puede obtener más información sobre el motivo por el que se realizó la modificación. ▪ <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio
Tipo de elemento	Bloque y en línea
Descripción	Se emplea para marcar una modificación en los contenidos originales consistente en la inserción de un nuevo contenido

	Borrado
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> ▪ <code>cite = "url"</code> - Indica la URL de la página en la que se puede obtener más información sobre el motivo por el que se realizó la modificación. ▪ <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio
Tipo de elemento	Bloque y en línea
Descripción	Se emplea para marcar una modificación en los contenidos originales consistente en el borrado de cierto contenido

Por defecto, el texto eliminado (marcado con la etiqueta ****) se muestra tachado de forma que el usuario pueda identificarlo fácilmente como un texto que formaba parte del texto original y que ya no tiene validez.

El texto insertado (marcado con la etiqueta **<ins>**) se muestra subrayado, de forma que el usuario pueda identificarlo como un texto nuevo que no formaba parte del texto original.

- ♦ etiqueta **<blockquote>** (vigente en HTML 5)

Por otra parte, en muchos tipos de páginas (artículos, noticias) es habitual citar literalmente un texto externo. HTML define la etiqueta **<blockquote>** para incluir citas textuales en las páginas web.

<blockquote>	Citas
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none">▪ cite = "url" - Indica la dirección de la página web original de la que se extrae la cita
Tipo de elemento	Bloque
Descripción	Se emplea para indicar que el texto que encierra es una cita textual de otro texto externo

Para indicar de forma clara que el texto es una cita externa, los navegadores muestran por defecto el texto del elemento <blockquote> con un gran margen en la parte izquierda.

En XHTML para validaciones hay que tener en cuenta que este elemento tiene que contener 1 o más elementos de bloque.

```
<blockquote>
  <p>Here is a long quotation here is a long quotation.</p>
</blockquote>
```

3.3. Marcado avanzado de texto

◆ Etiquetas <abbr> y <acronym>

La etiqueta <abbr> marca las abreviaturas de un texto y la etiqueta <acronym> se emplea para marcar las siglas o acrónimos del texto.

Abbr (soportado en HTML 5 también)

acronym (no soportado en HTML 5) en su lugar se usa abbr.

Ejemplo:

```
<p>La <abbr title="World Health Organization">WHO</abbr> se fundó en 1948. </p>
```

Su definición es la siguiente:

<abbr>	Abreviaturas
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none">▪ <code>title = "texto"</code> - Indica el significado completo de la abreviatura
Tipo de elemento	En línea
Descripción	Se emplea para marcar las abreviaturas del texto y proporcionar el significado de esas abreviaturas

<acronym>	Acrónimos o siglas
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none">▪ <code>title = "texto"</code> - Indica el significado completo del acrónimo o sigla
Tipo de elemento	En línea
Descripción	Se emplea para marcar las siglas o acrónimos del texto y proporcionar el significado de esas siglas

La mayoría de navegadores muestran por defecto un borde inferior punteado para todos los elementos `<abbr>` y `<acronym>`. Al posicionar el puntero del ratón sobre la palabra subrayada, el navegador muestra un pequeño recuadro (llamado *tooltip* en inglés) con el valor del atributo `title`.

♦ **Etiqueta `<dfn>`** (vigente en HTML 5)

Por otra parte, en ocasiones resulta útil incluir la definición de una palabra extraña o cuyo uso está restringido a un entorno muy determinado. HTML incluye la etiqueta `<dfn>` para proporcionar al usuario la definición de todas las palabras para las que se considere apropiado. La definición formal de esta etiqueta se muestra a continuación:

<dfn>	Definición
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none">▪ <code>title = "texto"</code> - Indica el significado completo del término
Tipo de elemento	En línea
Descripción	Se emplea para marcar las definiciones de ciertos términos y proporcionar el significado de esos términos

El siguiente ejemplo muestra cómo se utiliza la etiqueta `<dfn>` para incluir la definición completa de una palabra cuyo uso no es habitual fuera de los ámbitos

médicos y psicológicos:

<p>Con estos síntomas, podría tratarse de un caso de <dfn title="Imagen o sensación subjetiva, propia de un sentido, determinada por otra sensación que afecta a un sentido diferente">sinestesia</dfn></p>

◆ Etiqueta <cite>

(En HTML 4.01) Sirve para marcar un texto como una citación. En ocasiones, no está clara la diferencia entre <cite> y <blockquote>. El elemento <cite> marca el autor de la cita (persona, documento, etc.) y <blockquote> marca el contenido de la propia cita.

<cite>	Cita
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	En línea
Descripción	Se emplea para marcar una cita o una referencia a otras fuentes

(En HTML 5) define el título de un trabajo (libro, programa de tv, canción, película).

3.4. Marcado genérico de texto

◆ Etiqueta

El estándar HTML/XHTML incluye una etiqueta llamada que se emplea para marcar cualquier elemento que no se puede marcar con las otras etiquetas definidas.

La etiqueta sólo se puede utilizar para encerrar contenidos y etiquetas en línea. Cuando se quieren estructurar elementos de bloque, se utiliza la etiqueta <div>, tal y como se verá en capítulos posteriores.

Ejemplo:

<p>el color azul es mi favorito</p>

3.5. Espacios en blanco y nuevas líneas

HTML considera *espacio en blanco* a los espacios en blanco, los tabuladores, los retornos de carro y el carácter de nueva línea (ENTER o Intro). HTML ignora todos los espacios en blanco *sobrantes*, es decir, todos los espacios en blanco que no son el espacio en blanco que separa las palabras.

3.5.1. Nuevas líneas

Para incluir una nueva línea en un punto y forzar a que el texto que sigue se muestre en la línea inferior, se utiliza la etiqueta `
`.

	Nueva línea
Atributos comunes	básicos
Atributos específicos	-
Tipo de elemento	En línea y etiqueta vacía
Descripción	Fuerza al navegador a insertar una nueva línea

3.5.2. Espacios en blanco

Para incluir espacios en blanco adicionales, se debe sustituir cada nuevo espacio en blanco por el texto ` `;

3.5.3. Texto preformateado

- ♦ Etiqueta `<pre>` (vigente en HTML 5)

En ocasiones, es necesario mostrar los espacios en blanco de un texto que no se puede modificar. Se trata de un caso habitual cuando una página web debe mostrar directamente el texto generado por alguna aplicación.

En estos casos, se puede utilizar la etiqueta `<pre>`, que muestra el texto tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas. La definición formal de la etiqueta se muestra a continuación:

<code><pre></code>	Texto preformateado
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	Bloque
Descripción	Muestra el texto que encierra tal y como está escrito (respetando los espacios en blanco)

Los elementos `<pre>` son *especiales*, ya que los navegadores les aplican las siguientes reglas:

- Mantienen todos los espacios en blanco (tabuladores, espacios y nuevas líneas)
- Muestra el texto con un tipo de letra especial, denominado "*de ancho fijo*", ya que todas sus letras son de la misma anchura
- No se ajusta la longitud de las líneas (las líneas largas producen un *scroll* horizontal en la ventana del navegador)

- ♦ Etiqueta `<code>` (vigente en HTML 5)

Otra etiqueta relacionada con `<pre>` es la etiqueta `<code>`, que se utiliza para mostrar código fuente de cualquier lenguaje de programación. La definición formal de `<code>` es la siguiente:

<code>	Código fuente
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo de elemento	En línea
Descripción	Delimita el texto considerado un fragmento de código fuente

En la mayoría de páginas web, no tiene sentido utilizar la etiqueta `<code>`. Sin embargo, en muchas páginas web técnicas que incluyen listados de programas, trozos de código o etiquetas HTML, lo correcto es emplear la etiqueta `<code>`.

Al igual que sucede con los elementos `<pre>`, el texto encerrado por la etiqueta `<code>` se muestra con un tipo de letra especial de ancho fijo. Por el contrario, el elemento `<code>` no respeta los espacios en blanco ni las líneas, por lo que su comportamiento es similar a la etiqueta `<p>`. La última diferencia es que `<code>` es un elemento en línea, mientras que `<pre>` es un elemento de bloque.

3.6. Codificación de caracteres

Una consideración importante directamente relacionada con el texto de las páginas HTML es la codificación de los caracteres y la inserción de caracteres *especiales*. Algunos de los caracteres que se utilizan habitualmente en los textos no se pueden incluir directamente en las páginas web:

- Los caracteres que utiliza HTML para definir sus etiquetas (`<`, `>` y `"`) no se pueden utilizar libremente.
- Los caracteres propios de los idiomas que no son el inglés (ñ, á, ç, ò, ï, etc.) pueden ser problemáticos dependiendo de la codificación de caracteres utilizada.

Tendremos en cuenta el juego de caracteres en dos momentos:

- Al grabar el archivo de texto, hay que elegir el conjunto apropiado. En el cuadro de diálogo *GUARDAR COMO...* de la aplicación que usemos para escribir el documento. *gedit* de *Linux* nos ofrece dos opciones: ISO-8859-15 y UTF-8 (UNICODE). *Notepad* de *Windows* ofrece una tercera opción: ANSI.
- Dentro del documento tendremos que informar del conjunto que hemos usado. Para ello incluiremos un elemento *META* al principio de la cabecera (*HEAD*) de la siguiente forma:

`<META http-equiv="Content-Type" content="text/html; charset=UTF8">`

Indicando UTF-8 como la codificación de caracteres que hemos usado.

La solución a la primera limitación consiste en sustituir los caracteres reservados de HTML por unas expresiones llamadas entidades HTML o referencias a entidades de caracteres y que representan a cada carácter:

Entidad	Carácter	Descripción	Traducción
<	<	less than	signo de menor que
>	>	more than	signo de mayor que
&	&	ampersand	ampersand
"	"	quotation mark	comillas
 	(espacio en blanco)	non-breaking space	espacio en blanco
'	'	apostrophe	apóstrofo

De esta forma, si se considera el siguiente texto:

Los caracteres <, >, " y & pueden dar problemas con los textos en HTML

Para mostrar correctamente el texto anterior en una página HTML, se debe sustituir cada carácter especial por su entidad HTML:

<p>Los caracteres <, >, " y & pueden dar problemas con los textos en HTML</p>

Por otra parte, los caracteres propios de los idiomas diferentes al inglés también pueden ser problemáticos. El motivo es que desde que se crea una página web hasta que llega al navegador del usuario, intervienen numerosos procesos:

- El diseñador crea la página web con su editor HTML.
- Si se trata de una aplicación dinámica, el programador recorta la página HTML del diseñador y la mezcla con el resto del código de la aplicación (por ejemplo PHP).
- El servidor web almacena las páginas HTML estáticas o el código de la aplicación web y sirve las páginas solicitadas por los usuarios.
- El usuario solicita y visualiza las páginas web a través de su navegador.

Si en todos los procesos anteriores se utiliza la misma codificación de caracteres, los caracteres propios de los idiomas se pueden escribir directamente:

<p>Este párrafo contiene caracteres acentuados y se almacena en formato UTF-8</p>

La palabra párrafo del ejemplo anterior incluye la letra á. Si el editor HTML del diseñador utiliza la codificación UTF-8, el entorno de desarrollo del programador también utiliza UTF-8, el servidor web sirve las páginas con esa codificación y el navegador del usuario es capaz de visualizar las páginas con formato UTF-8, el texto

anterior se verá correctamente en el navegador del usuario.

Sin embargo, muchas veces no es posible que todos los procesos involucrados utilicen la misma codificación de caracteres. Por limitaciones técnicas o por decisiones de los diseñadores y programadores, los textos pueden pasar de codificación UTF-8 a codificación ISO-8859 en cualquier momento. Si se produce este cambio sin realizar una conversión correcta, el navegador del usuario mostrará caracteres extraños en todos los acentos y en todas las letras como la ñ.

La solución más sencilla para asegurar que todos estos caracteres potencialmente problemáticos se van a visualizar correctamente en el navegador del usuario consiste en sustituir cada carácter problemático por su entidad HTML:

Entidad	Carácter	Descripción oficial
ñ	ñ	latin letter n with tilde
Ñ	Ñ	latin capital n letter with tilde
á	á	a acute
é	é	e acute
í	í	i acute
ó	ó	o acute
ú	ú	u acute
Á	Á	A acute
É	É	E acute
Í	Í	I acute
Ó	Ó	O acute
Ú	Ú	U acute
€	€	euro

Así, el párrafo de texto del ejemplo anterior, se podría escribir de la siguiente manera:

<p>Este párrafo contiene caracteres acentuados y se almacena en formato UTF-8</p>

Si se utilizan las entidades HTML en vez de los caracteres problemáticos, es indiferente pasar de una codificación de caracteres a otra diferente. En la Wikipedia se puede consultar [la lista completa de las 252 entidades HTML definidas](http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references) ([http://en.wikipedia.org/wiki/ List_of_XML_and_HTML_character_entity_references](http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references))

4. Enlaces

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.

Una característica que no se suele tener en cuenta en los enlaces es que están formados por dos extremos y un sentido. En otras palabras, el enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de los dos extremos se llaman "*anchors*" en inglés, que se puede traducir literalmente como "anclas".

4.1. URL

El acrónimo URL (del inglés *Uniform Resource Locator*) hace referencia al identificador único de cada recurso disponible en Internet. Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

La URL de un recurso tiene dos objetivos principales:

- Identificar de forma única a ese recurso
- Permitir localizar de forma eficiente ese recurso

Si se accede a la página principal de Google, la dirección que muestra el navegador es:

`http://www.google.com`

La cadena de texto `http://www.google.com` es la URL completa de la página principal de Google.

`http://www.aulavirtual.es/xhtml/capitulo4.html`

`http://www.aulavirtual.es/comments/webs2012?page=5#42`

Las partes que componen una URL son:

- Protocolo (`http://`): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan `http://`. Las páginas web *seguras* (por ejemplo las de los bancos y las de los servicios de email) utilizan `https://` (se añade una letra s).
- Servidor (www.aulavirtual.es): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- Ruta (`/xhtml/capitulo4.html` o `/comments/webs2012`): *camino* que se debe seguir,

una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

- Consulta (?page=5): información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el carácter ? y contiene una sucesión de palabras separadas por = y &
- Sección (#42): permite que el navegador se posicione automáticamente en una sección de la página web. Siempre comienza con el caracter #

Como las URL utilizan los caracteres :, =, & y / para separar sus divisiones, estos caracteres están reservados y no se pueden utilizar libremente. Además, algunos caracteres no están reservados pero pueden ser problemáticos si se utilizan en la propia URL.

Si es necesario incluir estos caracteres reservados y especiales en una URL, se sustituyen por combinaciones de caracteres seguros. Esta sustitución se denomina *codificación* de caracteres y el servidor realiza el proceso inverso (*decodificación*) cuando le llega una URL con los caracteres codificados.

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%27	~	%7E
(espacio en blanco)	%20		%7C

Por otra parte, aunque desde hace tiempo ya es posible incluir en las URL caracteres de otros idiomas que no sean el inglés, aún no es completamente seguro utilizar estos caracteres en las URL. Si se utilizan letras como ñ, á, é o ç, es posible que algunos navegadores no las interpreten de forma correcta.

La solución consiste en codificar todos los caracteres que no existen en inglés. La siguiente tabla muestra la codificación de los caracteres más utilizados:

Carácter original	Carácter codificado	Carácter original	Carácter codificado
ñ	%F1	Ñ	%D1
á	%E1	Á	%C1
é	%E9	É	%C9

í	%ED	Í	%CD
ó	%F3	Ó	%D3
ú	%FA	Ú	%DA
Ç	%E7	Ç	%C7

Teniendo en cuenta las dos tablas anteriores de codificación de caracteres, es fácil crear las URL correctas sin caracteres problemáticos:

```
<!-- URL problemática -->  
http://www.ejemplo.com/estaciones/otoño.html
```

```
<!-- URL correcta -->  
http://www.ejemplo.com/estaciones/oto%F1o.html
```

```
<!-- URL problemática -->  
http://www.ejemplo.com/ruta/nombre página.html
```

```
<!-- URL correcta -->  
http://www.ejemplo.com/ruta/nombre%20p%E1gina.html
```

4.2. Enlaces relativos y absolutos

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos.

En esa página, cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "enlaces externos". Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "enlaces internos".

Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo.

- Las **URL absolutas** incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.
- Las **URL relativas** prescinden de algunas partes de las URL para hacerlas más breves. Como se trata de URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado. En concreto, para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace.

Las URL relativas se construyen a partir de las URL absolutas y prescinden de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado. Aunque las URL relativas pueden ser difíciles de entender para los que comienzan con HTML, son tan útiles que todos los sitios web las utilizan.

A continuación se muestran los cuatro tipos diferentes de URL relativas:

1) El origen y el destino del enlace se encuentran en el mismo directorio

Si desde una página web se quiere enlazar un recurso que se encuentra en el mismo directorio del servidor, la URL relativa puede prescindir de todas las partes de la URL absoluta salvo el nombre del recurso enlazado.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el mismo directorio
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
URL relativa	pagina2.html

2) El destino del enlace se encuentra cerca de su origen y en un nivel superior

En este caso, el recurso que se enlaza no está en el mismo directorio que el origen del enlace pero sí que está *cerca* y en algún directorio superior. Para indicar al navegador que debe subir un nivel, se incluyen dos puntos y una barra (../) en la ruta del recurso enlazado.

De la misma forma, si el destino se encuentra un par de niveles por encima, se debe incluir ../ dos veces seguidas:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta2
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/pagina2.html
URL relativa	../pagina2.html

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio llamado ruta4 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta4/pagina2.html
URL relativa	../../ruta4/pagina2.html

Además de subir niveles, también se puede entrar en otros directorios para obtener los recursos:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta1
URL absoluta	http://www.ejemplo.com/ruta1/pagina2.html
URL relativa	../../pagina2.html

Si se intentan subir más niveles de los que es posible, el navegador ignora todos los ../ sobrantes.

3) El destino del enlace se encuentra cerca de su origen y en un nivel inferior

Este caso es muy similar al anterior, pero más sencillo.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta6 que está dentro del directorio ruta5 y que a su vez está dentro del directorio ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html
URL relativa	ruta4/ruta5/ruta6/pagina2.html

De la misma forma, se pueden indicar varios directorios seguidos para que el navegador descienda jerárquicamente por la estructura de directorios:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html
URL relativa	ruta4/pagina2.html

4) El origen y el destino del enlace se encuentran muy alejados

Cuando el origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor) las URL relativas se pueden complicar en exceso. Aunque es posible utilizar ../ para subir por la jerarquía de directorios y se puede entrar en cualquier directorio indicando su nombre, las URL relativas que se obtienen son demasiado largas y complicadas.

En estos casos, lo más sencillo es indicar la ruta completa hasta el recurso enlazado

comenzando desde la raíz del servidor web. Por lo tanto, estas URL relativas sólo omiten el protocolo y el nombre del servidor.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se guarda en un directorio llamado ruta7 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta7/pagina2.html
URL relativa	/ruta7/pagina2.html

Cuando la URL relativa comienza por /, el navegador considera que es la ruta completa comenzando desde la raíz del servidor, por lo que sólo le añade el protocolo y el nombre del servidor origen.

4.3. La etiqueta <a>

Los enlaces en HTML se crean mediante la etiqueta <a> .

<a>	Enlaces
Atributos comunes	básicos, í18n, eventos y foco
Atributos específicos	<ul style="list-style-type: none">▪ name = "texto" - Permite nombrar al enlace para que se pueda acceder desde otros enlaces▪ href = "url" - Indica la URL del recurso que se quiere enlazar▪ hreflang = "codigo_idioma" - Idioma del recurso enlazado▪ type = "tipo_de_contenido" - Permite "avisar" al navegador sobre el tipo de contenido que se enlaza (imágenes, archivos, etc.) para que pueda prepararse en caso de que no entienda ese contenido▪ rel = "tipo_de_relacion" - Describe la relación del documento actual con el recurso enlazado▪ rev = "tipo_de_relacion" - Describe la relación del recurso enlazado con el documento actual▪ charset = "tipo_de_charset" - Describe la codificación del recurso enlazado
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

◆ **href:** (vigente en HTML 5)

Se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante href.

Ej 1: Crear un enlace que apunte a la página principal de Google :

```
<a href="http://www.google.com">Página principal de Google</a>
```

Ej 2: El siguiente enlace apunta a una imagen, que se mostrará en el navegador cuando el usuario pinche sobre el enlace:

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg">Imagen interesante para un fondo de escritorio</a>
```

Ej 3: De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc.

```
<a href="http://www.ejemplo.com/informe.pdf">Descargar informe completo [PDF]</a>  
<a href="http://www.ejemplo.com/informe.doc">Descargar informe completo [DOC]</a>
```

◆ **name:** (no soportado en HTML 5)

Permite definir enlaces dentro de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces de tipo "Saltar hasta la segunda sección", "Volver al principio de la página", etc.

Este tipo de enlaces son especiales, ya que la URL de la página siempre es la misma para todas las secciones y por tanto, debe añadirse otra parte a las URL para identificar cada sección.

```
<a name="primera_seccion"></a> <p>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus.  
Praesent condimentum mattis ligula.</p>
```

...

```
<a name="segunda_seccion"></a> <p>Pellentesque malesuada. In in lacus. Phasellus  
erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat  
varius, facilisis eget, dolor.</p>
```

...

El atributo name permite crear "enlaces vacíos" que hacen referencia a secciones dentro de una misma página. Una vez definidos los "enlaces vacíos", es posible crear un enlace que apunte directamente a una sección concreta de una página:


```
<!-- Enlace normal a la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html">Enlace a la página 1</a>
```

```
<!-- Enlace directo a la segunda sección de la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html#segunda_seccion">Enlace a la sección 2 de la página 1</a>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el símbolo # seguido del nombre de la sección a la que se apunta. Cuando el usuario pincha sobre uno de estos enlaces, el navegador accede a la página apuntada por la URL y baja directamente a la sección cuyo nombre se indica después del símbolo #.

También es posible utilizar este tipo de enlaces con URL relativas en una misma página. El siguiente ejemplo añade enlaces de tipo "Volver al inicio de la página" en varias secciones:

```
<p><a name="inicio"></a> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
```

...

```
<p><a href="#inicio">Volver al inicio de la página</a> </p>
```

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p><a href="#inicio">Volver al inicio de la página</a> ...
```

En HTML 5 este atributo queda obsoleto y en su lugar se hace uso del atributo id de cualquier elemento:

El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p><a href="#inicio">Volver al inicio de la página</a> ...
```

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p><a href="#inicio">Volver al inicio de la página</a> ...
```

El nombre de la sección que se indica después del símbolo # puede utilizar el valor de los atributos id de cualquier elemento. De hecho, se recomienda utilizar los atributos id de los elementos ya existentes en la página en vez de crear "enlaces vacíos" de tipo ``.

♦ **Idioma del enlace (hreflang)** (vigente en HTML 5)

El enlace puede indicar al navegador el idioma del recurso que se enlaza. Para establecer el valor del idioma, se utiliza un código estandarizado de dos letras. Además del idioma genérico, también se puede indicar una variación idiomática. Ejemplo de códigos de idioma más utilizados:

Código	Idioma	Variación idiomática
en	Inglés	-
en-US	Inglés	Estados Unidos
es	Español	-
es-ES	Español	España
es-AR	Español	Argentina

Otros códigos utilizados son: fr (francés), de (alemán), it (italiano), nl (holandés), el (griego), pt(portugués), ar(árabe), he(hebreo), ru(ruso), zh(chino), ja(japonés).

La lista completa de códigos de idioma está definida en [el estándar ISO 639](http://xml.coverpages.org/iso639a.html) (<http://xml.coverpages.org/iso639a.html>) .

♦ **Tipo de contenido (type)** (vigente en HTML 5)

Se utiliza para notificar al navegador sobre el tipo de contenido que se enlaza. Se indica mediante una cadena de texto cuyos posibles valores también están estandarizados. Los valores de los contenidos más utilizados son los siguientes: "text/html" (páginas HTML), "image/png" (imágenes con formato PNG), "image/gif" (imágenes con formato GIF), "text/css" (hojas de estilo CSS), "application/rss+xml" (archivos RSS).

La lista completa de tipos de contenido se define en [los estándares RFC 2045 y RFC 2046](http://www.iana.org/assignments/media-types/) (<http://www.iana.org/assignments/media-types/>) .

♦ **Tipo de relación (rel y rev)** (rel vigente en HTML 5 y rev no)

Los enlaces pueden proporcionar información adicional muy útil para los navegadores y para los motores de búsqueda como Google. Los atributos rel y rev permiten indicar la relación que la página actual tiene con la página a la que

se enlaza (atributo rel) y la relación que tiene la página enlazada con la página actual (atributo rev).

Los tipos de relación definidos son los siguientes:

- alternate- Indica que es una versión alternativa al documento actual (puede ser una versión en otro idioma o una versión preparada para otro medio, como una impresora o un dispositivo móvil)
- stylesheet - Indica que se ha enlazado una hoja de estilos
- start- Indica que se trata del primer documento de una colección de documentos (por ejemplo el primer capítulo de un libro)
- next- Indica que es el documento que sigue al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)
- prev- Indica que es el documento que precede al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)
- contents - Indica que el recurso enlazado es el documento que contiene la tabla de contenidos de la colección de documentos (por ejemplo, el índice de un libro).
- bookmark - Establece el enlace actual como un "marcador" o "favorito". Un marcador es un enlace que constituye un punto de entrada muy importante dentro del documento.

La especificación oficial de HTML define la [lista completa de tipos de relaciones](http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-links) (<http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-links>) que se pueden utilizar.

◆ **Codificación de caracteres (charset)** (no soportado en HTML 5)

Además del idioma, tipo de contenido y relación del recurso que se enlaza, los enlaces también pueden indicar la codificación de caracteres que utiliza la página web enlazada.

Los valores que se pueden utilizar también están estandarizados y las codificaciones más utilizadas son UTF-8 y ISO-8859-1, aunque existen decenas de códigos definidos (ISO-10646-UCS-2, IBM852, Big5-HKSCS, windows-1252, HZ-GB-2312).

El organismo IANA publica la [lista completa de codificaciones de caracteres disponibles](http://www.iana.org/assignments/character-sets) (<http://www.iana.org/assignments/character-sets>) .

Los ejemplos anteriores de enlaces básicos se pueden rehacer utilizando algunos de los atributos definidos por la etiqueta <a>:

```
<a href="http://www.google.com" hreflang="en" type="text/html" charset="UTF-8">Página principal de Google</a> <a href="http://www.ejemplo.com/fondo_escritorio.jpg" type="image/jpg">Imagen interesante para un fondo de escritorio</a>
```

4.5. Otros tipos de enlaces

HTML define las etiquetas `<script>` y `<link>` para enlazar recursos que se deben cargar automáticamente. Cuando el navegador encuentra alguna de estas dos etiquetas, descarga los recursos enlazados y los aplica a la página web.

La etiqueta `<script>` (vigente en HTML 5) tiene dos modos de funcionamiento, ya que se emplea tanto para insertar un bloque de código JavaScript en la página como para enlazar un archivo JavaScript externo.

Aunque la etiqueta `<script>` permite enlazar código de varios lenguajes de programación, el uso habitual de `<script>` consiste en enlazar un archivo JavaScript externo:

```
<head>
<script type="text/javascript" src="http://www.ejemplo.com/js/ inicializar.js"></script>
</head>
```

<script>	Código ejecutable
Atributos comunes	-
Atributos específicos	<ul style="list-style-type: none">▪ <code>src = "url"</code> - Indica la dirección del archivo que contiene el código▪ <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de código que se incluye (normalmente JavaScript)▪ <code>defer = "defer"</code> - El código no va a modificar el contenido de la página web▪ <code>charset = "tipo_de_charset"</code> - Describe la codificación del código enlazado
Tipo de elemento	Bloque y en línea (también puede ser una etiqueta vacía)
Descripción	Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)

El atributo `type` utilizado habitualmente para los archivos JavaScript es `"text/javascript"`. El atributo `src` es equivalente al atributo `href` de los enlaces creados con la etiqueta `<a>`. La URL indicada en el atributo `src` puede ser absoluta o relativa y externa o interna.

Además de enlazar un archivo JavaScript externo, la misma etiqueta `<script>` también permite incluir en la página web un bloque de código JavaScript:

```
<html>
<head>
  <script type="text/javascript">
    //<![CDATA[
```

```
    window.onload = function() { alert("La página se ha cargado completamente"); }  
  //]]>  
</script> </head>  
<body> ... </body>  
</html>
```

Cuando se incluye código JavaScript en la propia página XHTML, se debe insertar dentro de una sección especial llamada CDATA. Para ello, el código JavaScript se debe encerrar entre `<![CDATA[y]]>`. Cuando el navegador encuentra una sección de este tipo, no procesa su contenido como si fuera XHTML y por tanto no tiene en cuenta los posibles errores de validación de XHTML.

La etiqueta `<script>` (tanto cuando enlaza, como cuando incluye directamente el código) puede aparecer en cualquier parte del documento HTML, aunque normalmente se incluye dentro de la cabecera de la página (`<head>...</head>`).

La segunda etiqueta de XHTML para enlazar recursos es `<link>`, que permite enlazar y relacionar la página con otros recursos externos.

Al contrario que `<script>`, la etiqueta `<link>` solamente se puede incluir dentro de la cabecera del documento. Se pueden añadir tantas etiquetas `<link>` como sean necesarias, pero siempre dentro de `<head>...</head>`.

El atributo `media` hace referencia al medio para el que es válida la relación con el recurso enlazado. Los medios disponibles también están estandarizados, siendo los más comunes `screen` para los contenidos mostrados en pantalla, `print` para las impresoras y `handheld` para los dispositivos móviles.

El uso habitual de la etiqueta `<link>` (vigente en HTML 5) es el de enlazar las hojas de estilos CSS utilizadas por las páginas web:

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

<link>	Enlazar recursos
Atributos comunes	básicos, id y eventos
Atributos específicos	<ul style="list-style-type: none">Los siguientes con el mismo significado que para la etiqueta "a": <code>charset</code>, <code>href</code>, <code>hreflang</code>, <code>type</code>, <code>rel</code> y <code>rev</code><code>media</code> = "tipo_de_medio" - Indica el medio para el que debe aplicarse la relación
Tipo de elemento	Etiqueta vacía
Descripción	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos

5. Metainformación

Estos datos adicionales siempre están relacionados con la propia página, por lo que se denominan *metainformación* o *metadatos*. La metainformación siempre se incluye en la sección de la cabecera, es decir, dentro de la etiqueta `<head>`.

5.1. Estructura de la cabecera

La cabecera típica de una página HTML completa presenta la siguiente estructura:

```
<head>
  <!-- Zona de etiquetas META -->
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

  <!-- Zona de título -->
  <title>El título del documento</title>

  <!-- Zona de recursos enlazados (CSS, RSS, JavaScript) -->
  <link rel="stylesheet" href="#" type="text/css" media="screen" />

  <link rel="stylesheet" href="#" type="text/css" media="print" />

  <link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="#" />

  <script src="#" type="text/javascript"></script> </head>
```

<head>	Cabecera
Atributos comunes	¡18n
Atributos específicos	<ul style="list-style-type: none">▪ <code>profile = "url"</code> - Especifica la URL del perfil o perfiles que utilizan los metadatos▪ <code>lang = "codigo_de_idioma"</code> - Especifica el idioma principal de los contenidos de la página
Tipo de elemento	-
Descripción	Define la cabecera del documento HTML

La etiqueta `<title>` establece el título de la página. Los navegadores muestran este título como título de la propia ventana del navegador. Los buscadores utilizan este título como título de sus resultados de búsqueda. Las páginas XHTML deben tener definido un título y sólo uno, por lo que todas las páginas web deben incluir obligatoriamente una etiqueta `<title>`.

Por último, la etiqueta `<head>` permite definir en el atributo profile la URL de un

documento externo que contiene el perfil que siguen los metadatos de la cabecera. Los blogs creados con el programa WordPress incluyen por ejemplo el siguiente perfil en su cabecera:

```
<head profile="http://gmpg.org/xfn/11"> ...  
</head>
```

<title>	Título del documento
Atributos comunes	i18n
Atributos específicos	<ul style="list-style-type: none">▪ lang = "codigo_de_idioma" - Especifica el idioma principal del título de la página
Tipo de elemento	-
Descripción	Define el título del documento HTML

El documento <http://gmpg.org/xfn/11> es un perfil que define atributos adicionales para establecer la relación entre sitios web.

5.2. Metadatos

Una de las partes más importantes de la metainformación de la página son los metadatos, que permiten incluir cualquier información relevante sobre la propia página. La etiqueta empleada para la definición de los metadatos es <meta>.

<meta>	Metadatos
Atributos comunes	i18n
Atributos específicos	<ul style="list-style-type: none">▪ name = "texto" - El nombre de la propiedad que se define (no existe una lista oficial de propiedades)▪ content = "texto" - El valor de la propiedad definida (no existe una lista de valores permitidos)▪ http-equiv = "texto" - En ocasiones, reemplaza al atributo "name" y lo emplean los servidores para adaptar sus respuestas al documento▪ scheme = "texto" - Indica el esquema que se debe emplear para interpretar el valor de la propiedad
Tipo de elemento	-
Descripción	Permite definir el valor de los metadatos que forman la metainformación del documento

Los metadatos habituales utilizan solamente los atributos name y content para definir el nombre y el valor del metadato:

```
<meta name="autor" content="Juan Pérez" />
```

No obstante, algunas etiquetas <meta> muy utilizadas hacen uso del atributo http-equiv. Este atributo se utiliza para indicar que el valor establecido por este metadato puede ser utilizado por el servidor al entregar la página al navegador del usuario. El siguiente metadato indica al servidor que el contenido de la página es código HTML y su codificación de caracteres es UTF-8:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

El atributo scheme no suele utilizarse, aunque permite proporcionar información de contexto para que el navegador interprete correctamente el valor del metadato. En el siguiente ejemplo, el atributo scheme indica al navegador que el valor del metadato hace referencia al código ISBN:

```
<meta scheme="ISBN" name="identificador" content="789-1392349610">
```

Aunque no existe una lista oficial con los metadatos que se pueden definir, algunos de ellos se utilizan en tantas páginas que se han convertido prácticamente en un estándar. A continuación se muestran los metadatos más utilizados:

Definir el autor del documento:

```
<meta name="author" content="Juan Pérez" />
```

Definir el programa con el que se ha creado el documento:

```
<meta name="generator" content="WordPress 2.8.4" />
```

Definir la codificación de caracteres del documento:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

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

Definir el copyright del documento:

```
<meta name="copyright" content="web.es" />
```

Definir el comportamiento de los buscadores:

```
<meta name="robots" content="index, follow" />
```

Definir las palabras clave que definen el contenido del documento:

```
<meta name="keywords" content="diseño, css, hojas de estilos, web, html" />
```


Definir una breve descripción del sitio:

```
<meta name="description" content="Artículos sobre diseño web, usabilidad y accesibilidad" />
```

La etiqueta que define la codificación de los caracteres (http-equiv="Content-Type") se emplea prácticamente en todas las páginas y las etiquetas que definen la descripción (description) y las palabras clave (keywords) también son muy utilizadas.

5.3. DOCTYPE

El estándar XHTML deriva de XML, por lo que comparte con él muchas de sus normas y sintaxis. Uno de los conceptos fundamentales de XML es la utilización del DTD o *Document Type Definition* ("Definición del Tipo de Documento").

El conjunto de normas, obligaciones y restricciones que se deben seguir al crear un documento de un determinado tipo, se recogen en su correspondiente DTD. El estándar XHTML define el DTD que deben seguir las páginas y documentos XHTML. En este documento se definen las etiquetas que se pueden utilizar, los atributos de cada etiqueta y el tipo de valores que puede tener cada atributo.

En realidad, la versión 1.0 del estándar de XHTML define tres DTD diferentes. Para indicar el DTD utilizado al crear una determinada página, se emplea una etiqueta especial llamada doctype. La etiqueta doctype es el único elemento que se incluye fuera de la etiqueta <html> de la página. De hecho, la declaración del doctype es lo primero que se debe incluir en una página web, antes incluso que la etiqueta <html>.

Como se verá más adelante, para que una página XHTML sea correcta y válida es imprescindible que incluya el correspondiente doctype que indica el DTD utilizado. A continuación se muestran los tres DTD que se pueden utilizar al crear páginas XHTML:

XHTML 1.0 Estricto

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Se trata de la variante con las normas más estrictas y las restricciones más severas. Las páginas web que incluyan este doctype, no pueden utilizar atributos relacionados con el aspecto de los contenidos, por lo que requiere una separación total de código HTML y estilos CSS.

XHTML 1.0 Transitorio

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Se trata de una variante menos estricta que la anterior, ya que permite el uso de algunos atributos HTML relacionados con el aspecto de los elementos.

XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Esta última variante la utilizan las páginas que están formadas por *frames*, una práctica completamente desaconsejada y que hoy en día sólo utilizan los sitios web obsoletos.

Por otra parte, además del DOCTYPE apropiado, también es necesario que las páginas web indiquen el namespace asociado. Un namespace en un documento XML permite diferenciar las etiquetas y atributos que pertenecen a cada lenguaje.

Si en un mismo documento se mezclan etiquetas de dos o más lenguajes derivados de XML (XHTML y SVG por ejemplo) y que tienen el mismo nombre, no se podría determinar a qué lenguaje pertenece cada etiqueta y por tanto, no se podría interpretar esa etiqueta o ese atributo. Los namespaces se indican mediante una URL.

El namespace que utilizan todas las páginas XHTML (independientemente de la versión y del DOCTYPE) es <http://www.w3.org/1999/xhtml> y se indica de la siguiente manera:

```
<html xmlns="http://www.w3.org/1999/xhtml"> ... </html>
```

De esta forma, es habitual que las páginas XHTML comiencen con el siguiente código:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
```

Aunque el código anterior es mucho más complicado que una simple etiqueta `<html>`, es imprescindible para que las páginas XHTML creadas sean correctas y superen satisfactoriamente el proceso de validación.

6. Comentarios

La sintaxis de los comentarios es la siguiente:

```
<!-- texto -->
```

7. Validación

La validación es el proceso que asegura que un documento escrito en un determinado lenguaje (por ejemplo XHTML) cumple con las normas y restricciones de ese lenguaje. Las normas y restricciones de los documentos escritos en XML (y en sus lenguajes derivados, como XHTML) se definen en el DTD o *Document Type Definition*

(*"Definición del Tipo de Documento"*).

Además de los validadores disponibles en herramientas como Dreamweaver y de los validadores gratuitos disponibles en Internet (el organismo W3C ha creado una herramienta que se puede utilizar gratuitamente a través de Internet: <http://validator.w3.org/>).

Si dispones del navegador Firefox, puedes instalar un validador mediante un complemento llamado [HTML Validator](https://addons.mozilla.org/es-ES/firefox/addon/249) (<https://addons.mozilla.org/es-ES/firefox/addon/249>). Tras su instalación, la primera vez que se reinicia Firefox se muestra una ventana en la que se solicita al usuario que configure el tipo de validación que se va a realizar.

Las opciones para elegir son:

- HTML Tidy (que ofrece ayuda para resolver los errores y es mejor para HTML)
- SGML Parser (ofrece menos ayuda, pero es el mismo que el validador del W3C)
- Serial (que realiza las dos validaciones de forma seguida).