

Lenguajes de Marcas: de HTML a XHTML 2.0

B. Cristina Pelayo García-Bustelo

Creación de Sitios Web con XHTML, CSS y JavaScript
Agaete 2007

- 1 Un poco de historia
- 2 Familia XHTML
- 3 Diferencias entre HTML y XHTML
- 4 XHTML: Elementos de marcado

SGML: Standard Generalized Markup Language

- Raíces de finales de los 60
- SGML es un sistema diseñado para crear formatos de documentos a través de su marcado estructural.
- SGML hace hincapié únicamente en la estructura del documento
- Este sistema ya tenía estructura de documento jerárquica y definiciones de tipos de documentos

Evolución del HTML

- HTML = HyperText Markup Language
- Usado para publicación de documentos en la Web desde 1990
 - Aplicación de SGML
 - Semántica genérica que permite representar documentos estructurados textuales o gráficos
- 1990: Tim Berners-Lee crea El Navegador WorldWideWeb
- HTML 2.0 estableció el estándar para las características básicas de HTML
 - Especificación de 1995
 - Tim Berners-Lee y Dan Connolly
 - Grupo de trabajo de HTML del IETF (Internet Engineering Task Force)

HTML 3.0

- Borrador en marzo de 1995
- Extensión de las características de HTML 2.0 para soportar tablas, flujos de texto alrededor de las imágenes y fórmulas matemáticas
- Gran diferencia entre las versiones
 - La estandarización y el desarrollo de toda la propuesta parecía poco manejable
 - Surge la especificación HTML 3.2

HTML 3.2

- Sustituye a HTML 2.0
- Recomendación para HTML en enero de 1997
- IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass, y Sun Microsystems
- Incluye soporte para tablas, applets, flujos de texto alrededor de las imágenes, subíndices o superíndices

HTML 4.0

- Recomendación para HTML en diciembre de 1997
- Opciones de soporte para multimedia
- Lenguajes de scripting
- Hojas de estilo
- Facilidades para la impresión
- Documentos más accesibles para usuarios con discapacidades
- Internacionalización

HTML 4.01 (Diciembre de 1999)

- Se incluyen correcciones a errores detectados en la versión anterior
- La recomendación establece que los agentes que soporten este lenguaje, deben admitir a las versiones anteriores
- La creación de nuevos elementos de marcado han llevado a grandes problemas de interoperabilidad

- 1 Un poco de historia
- 2 Familia XHTML
 - Primer paso: XHTML 1.0
 - Segundo paso: Modularización de XHTML
 - Tercer paso: XHTML 2.0
- 3 Diferencias entre HTML y XHTML
- 4 XHTML: Elementos de marcado

XHTML: eXtensible HyperText Markup Language

- Es el sucesor de HTML como un conjunto de módulos XML
- XHTML no identifica a un único lenguaje de marcas, sino que es el nombre de una familia de tipos de documentos.
- Semántica de elementos idéntica a la de HTML 4
- Es creada para mejorar la interoperabilidad
- Mejora las capacidades de transformación de la representación del contenido y un escalado de las funcionalidades

1 Un poco de historia

- SGML
- HTML

2 Familia XHTML

- Primer paso: XHTML 1.0
- Segundo paso: Modularización de XHTML
- Tercer paso: XHTML 2.0

3 Diferencias entre HTML y XHTML

4 XHTML: Elementos de marcado

- DOCTYPE, espacio de nombres y codificación
- Metadatos
- Módulos de XHTML
- Mecanismos de estructuración

XHTML 1.0: el primer paso

- Es una reformulación de HTML 4 en XML 1.0: se definen tres DTDs.
- Se garantiza la compatibilidad con HTML siguiendo unas normas
- Recomendación desde 26/01/2000, revisión el 01/08/2002

XHTML 1.0 Transitional

- Es el más parecido a HTML
- Permite la utilización de estructuras de marcado de presentación
- Además de otros elementos y atributos obsoletos (Pj: target)
- Uso de colores de fondo aplicado a celdas de tablas
- NO ESTA PERMITIDO SU USO (pensado para los diseñadores que realizan la transición)

Declaración XHTML 1.0 Transitional

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

XHTML 1.0 Frameset

- La DTD de los 90s
- Permite la utilización de marcos
- NO ESTA PERMITIDO SU USO

Declaración XHTML 1.0 Frameset

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


XHTML 1.0 Strict

- La DTD es “severa”
- Obliga a utilizar elementos y atributos de marcado de presentación
- SI ESTA PERMITIDO SU USO

Declaración XHTML 1.0 Strict

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

1 Un poco de historia

- SGML
- HTML

2 Familia XHTML

- Primer paso: XHTML 1.0
- Segundo paso: Modularización de XHTML
- Tercer paso: XHTML 2.0

3 Diferencias entre HTML y XHTML

4 XHTML: Elementos de marcado

- DOCTYPE, espacio de nombres y codificación
- Metadatos
- Módulos de XHTML
- Mecanismos de estructuración

Segundo paso: Modularización de XHTML

- La descomposición de XHTML 1.0 en una colección de módulos abstractos que permiten especificar diferentes tipo de funcionalidad.
- También define una forma para diseñar módulos propios que pueden combinarse con los módulos de XHTML
- Recomendación del 10/04/2001

Cuatro Módulos centrales definidos

- Módulo de estructura: html, head, body, title
- Módulos de texto: abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var
- Módulo de hipertexto: a
- Módulo de listas: dl, dt, dd, ol, ul, li

Otros módulos

- Módulos de texto extendido: Presentation Module, Edit, Bi-directional Text
- Módulos de formularios: Basic Forms, Forms
- Módulo de tablas: Basic Tables, Tables
- Miscelanea: Image, Client-side Image Map, Server-side Image Map, Object, Frames, Target, Iframe, Intrinsic Events, Metainformation, Scripting, Style Sheet, Style Attribute, Link, Base
- Módulos obsoletos: Applet, Name Identification, Legacy

XHTML 1.1

- Es una recomendación desde 31/05/2001
- Realiza una reformulación de XHTML 1.0 Strict basado en módulos XHTML

- 1 Un poco de historia
 - SGML
 - HTML
- 2 Familia XHTML
 - Primer paso: XHTML 1.0
 - Segundo paso: Modularización de XHTML
 - Tercer paso: XHTML 2.0
- 3 Diferencias entre HTML y XHTML
- 4 XHTML: Elementos de marcado
 - DOCTYPE, espacio de nombres y codificación
 - Metadatos
 - Módulos de XHTML
 - Mecanismos de estructuración

XHTML 2.0

- Es la próxima generación de XHTML
- Objetivo: es utilizar tantas características de XML como sean posibles
- Integrando nuevas tecnologías: XFORM, XML Events, ...
- 26/07/2006: Publicado borrador de trabajo.
- 07/03/2007: Creación del grupo de trabajo.

- 1 Un poco de historia
- 2 Familia XHTML
- 3 Diferencias entre HTML y XHTML**
- 4 XHTML: Elementos de marcado

Documentos “bien formados”

- Documentos deben estar “bien formados” o gramaticalmente correctos
- Es un nuevo concepto introducido por XML
- Significa que todos los elementos deben tener etiquetas de cierre, deben ser escritos de manera especial y deben estar anidados.
- Aunque el solapamiento de elementos no está permitido en SGML, era tolerado en los navegadores existentes.

Ejemplo

- CORRECTO: elementos anidados

`<p>he aquí un párrafo enfatizado.</p>`

- INCORRECTO: elementos solapados

`<p>he aquí un párrafo</p> enfatizado`

Elementos y atributos en minúsculas

- Los documentos XHTML deben usar minúsculas para los nombres de todos los elementos y atributos HTML.
- Necesario porque XML es sensible a minúsculas y mayúsculas

Ejemplo

- CORRECTO:

`Un elemento de una lista`

- INCORRECTO

`Un elemento de una lista`

Todos los elementos (no vacíos) requieren etiquetas de cierre

- En HTML se podían omitir las etiquetas de cierre (era implícito)
- En XHTML es necesaria etiqueta de cierre para todos los elementos no marcados en el DTD como EMPTY

Ejemplo

- CORRECTO:

```
<p>he aquí un párrafo.</p>
```

```
<p>aquí hay otro párrafo.</p>
```

- INCORRECTO

```
<p>he aquí un párrafo.
```

```
<p>aquí hay otro párrafo.
```


Atributos entre comillas

- Todos los valores de atributos deben ir entrecomillados, incluso aquellos que son numéricos.

- CORRECTO:

```
<table rows="3">
```

- INCORRECTO:

```
<table rows=3>
```

Minimización de Atributos

- XML no soporta la minimización de atributos.
- Los pares atributo-valor deben escribirse en toda su extensión.

- CORRECTO:

```
<dl compact="compact">
```

- INCORRECTO:

```
<dl compact>
```

Elementos vacíos

- Los elementos vacíos deben tener una etiqueta de cierre o terminar en `/>`

- CORRECTO:

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

- INCORRECTO:

```
<br><hr>
```

Espacios en blanco

- En los valores de atributos, se eliminan los espacios en blanco de encabezamiento y terminación
- Se sustituirán las secuencias de uno o más espacios en blanco (o saltos de línea) por un único espacio en blanco entre palabras

Elementos script y style

- En XHTML, los elementos script y style se declaran como elementos con contenido #PCDATA.
- “<” y & son tratados como comienzos de etiquetado
- Se engloba el contenido del script o style dentro de una sección marcada como CDATA, que se reconocen por XML como nodos del Modelo de Objetos del Documento (DOM)
- ALTERNATIVA: utilizar documentos externos de estilo y escritura de código
- Todos los símbolos “<” y & deben codificarse como < y &

Las exclusiones de SGML

- SGML permite impedir la anidación de elementos específicos (pj. <a>). Son las “exclusiones”
- Pero en XML no es posible determinar exclusiones.
- Pero algunos elementos NO deben anidarse:
 - a no puede contener otros elementos a.
 - pre no puede contener los elementos img, object, big, small, sub o sup.
 - button no puede contener los elementos input, select, textarea, label, button, form, fieldset, iframe o isindex.
 - label no puede contener otros elementos label.
 - form no puede contener otros elementos form.

Elementos con atributos *id* y *name*

- En HTML 4.0 estos atributos son utilizados como identificadores de fragmentos de información.
- En HTML 4.0 el atributo *name* se definía para los elementos: a, applet, frame, iframe, img, y map.
- En XML (y en XHTML) el atributo *id* son del tipo ID, y tan sólo puede haber un único atributo de tipo ID por elemento.
- En XHTML 1.0, el atributo name está formalmente prohibido.

- 1 Un poco de historia
- 2 Familia XHTML
- 3 Diferencias entre HTML y XHTML
- 4 XHTML: Elementos de marcado**
 - DOCTYPE, espacio de nombres y codificación
 - Metadatos
 - Módulos de XHTML
 - Mecanismos de estructuración

DOCTYPE

- Definición del tipo de documento (DTD)
- La declaración DOCTYPE informa a los servicios de validación y agentes de usuario qué DTD se ha aplicado.
- Además les informa de cómo procesar la página
- Elemento CLAVE de las páginas Web compatibles.
- Las páginas no se validan si no comienzan con un DOCTYPE correcto.

Ejemplo DOCTYPE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

El espacio de nombres

- Es una colección de tipos de elemento y de nombres de atributo asociados a un DTD.
- Se identifica el espacio de nombres haciendo referencia a su ubicación en línea.
- El atributo “xml:lang” indica el idioma del documento.
- Se añade después del DOCTYPE.

Ejemplo del Espacio de nombres

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

La declaración XML

- Precede a la declaración del DOCTYPE
- Especifica la versión de XML y el tipo de codificación de caracteres
- W3C recomienda empezar todos los documentos XML (incluidos los XHTML) con un prólogo XML

Ejemplo de declaración XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

Codificación de caracteres

- Los ordenadores manejan código binario: 0s y 1s (Bytes: Grupos de 8 bits)
- Numeros: Sistemas de codificación binaria, octal, hexadecimal...
- Caracteres: Código que asocia a cada carácter un número
 - ASCII (American Standard Code for Information Interchange): 7 bits (0-127)
- Extensiones de ASCII
 - ISO-8859-1 (iso-latin-1): 8 bits ASCII (0-127) + otros caracteres típicos de Europa occidental
 - Familia ISO-8859-X = Otros alfabetos europeos
 - ISO-8859-15 (iso-latin-9) Igual que iso-8859-1 + símbolo de €

UNICODE

- ISO-10646 (31 bits) Define un repertorio universal de caracteres (UCS). En continua revisión contiene más de 70.000 caracteres.
- UNICODE = Consorcio de empresas que define restricciones sobre la implementación de ISO-10646
- Varias codificaciones (UTF = Unicode Transformation Format)
 - UTF-8: Los primeros 127 códigos se presentan igual (compatible con ASCII). El resto se codifican en longitud variable. Relativamente Eficiente
 - UTF-16: Usa 16 bits para los caracteres más comunes, el resto con pares de 16 bits
 - UTF-32: Codificación directa en 32 bits (desperdicio de espacio)

Ejemplo de definición de la codificación

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
<title>Ejemplo Codificación Caracteres </title>
<meta http-equiv="Content-Type"
  content="text/html; charset=UTF-8"/>
</head>
```

Metadatos

- Los elementos META se utilizan para especificar información sobre el propio documento
- La etiqueta META siempre se sitúa en el bloque “head”
- Se compone de un par nombre/valor
- Existen atributos estandar, obligatorios y opcionales

Metadatos: atributos requeridos

Atributo	Valor	Descripción
content	texto	Define meta información asociada a un http-equiv o name

Ejemplo: palabras claves del documento

```
<meta name="keywords" content="HTML, DHTML, CSS,  
XML, XHTML, JavaScript, VBScript" />
```

Metadatos: atributos opcionales

Atributo	Valor	Descripción
http-equiv	content-type expires refresh set-cookie	Relaciona el contenido del atributo con un encabezado HTTP
name	author description keywords generator revised otros	Relaciona el contenido del atributo con un nombre
scheme	texto	Define un formato que será utilizado para interpretar el valor del atributo

Ejemplo: metadatos

```
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"/>
<meta name="keywords"
  content="lenguaje, estándares,
  w3c, xhtml, css, euitio"/>
<meta name="description"
  content="Página web del Curso de Creación
  de Sitios Web con XHTML, CSS
  y JavaScript - Agaete 2007"/>
```

Los 28 módulos de XHTML (1/3)

- Módulo de estructura: html, head, body, title
- Módulos de texto: abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var
- Módulo de hipertexto: a
- Módulo de listas: dl, dt, dd, ol, ul, li
- Módulo de Objetos: object, param
- Módulo de Presentación: b, big, hr, i, small, sub, sup, tt

Los 28 módulos de XHTML (2/3)

- Módulo de Edición: del, ins
- Módulo de Texto Bidireccional: bdo
- Módulo de Formularios: button, fieldset, form, input, label, legend, select, optgroup, option, textarea
- Módulo de Tablas: caption, col, colgroup, table, tbody, td, tfoot, th, thead, tr
- Módulo de Imagen: img
- Módulo de Mapa de Imagen del lado Cliente: area, map
- Módulo de Mapa de Imagen del lado Servidor: Attribute ismap on img

Los 28 módulos de XHTML (3/3)

- Módulo de Eventos Intrínsecos: Events attributes
- Módulo de Metainformación: meta
- Módulo de Scripting: noscript, script
- Módulo de Hoja de Estilo: style element
- Módulo del Atributo Style (obsoleto): style attribute
- Módulo de Link: link
- Módulo de Base: base
- Ruby Annotation Module: ruby, rbc, rtc, rb, rt, rp

Módulos de estructura

- *html*: elemento principal de un documento XHTML
- *head*: representa la cabecera del documento, define el título, las hojas de estilo, los metadatos, etc.
- *title*: título de la página, se utiliza para la barra del agente de usuario y por los robots de búsqueda.
- *body*: contenido del documento XHTML.

Ejemplo

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xml:lang="es">
<head>
  <title>Página Personal</title>
</head>
<body>
  ...
</body>
</html>
```

Módulos de texto

- Este módulo contiene todos los elementos contenedores y atributos de texto básico.
- *h1*, *h2*, *h3*, *h4*, *h5* y *h6*: representan “títulos” deben seguir la secuencia lógica y el elemento *h1* debe aparecer una única vez en un documento dado que representa el título del contenido general del documento. (Ejemplo: titulos.html)
- *p*: representa un párrafo de texto (Ejemplo: parrafo.html)
- *pre*: representa un párrafo preformateado, mantiene los saltos de línea, los espacios en blanco, los tabuladores. (Ejemplo: textoPre.html)

Módulos de texto

- *blockquote*: es un bloque que contiene una cita, los agentes de usuario suelen “sangrar” el contenido. Tiene un atributo “cite=URI” para indicar el sitio web de la cita.
- *q*: se utiliza para marcar una cita pero no produce ningún tipo de efecto visual en el agente de usuario.(Ejemplo: `blockquote.html`)
- *cite*: se utiliza para citas dentro de un párrafo de texto.
- Ejemplo: `citas.html`

Módulos de texto

- *abbr*: representa abreviaturas
- *acronym*: representa acrónimos
- Los acrónimos y abreviaturas se utilizan habitualmente, pero hay que explicar lo que significan, para ello se define un atributo *title* que define su significado al pasar el ratón por encima del texto, aparece un rectángulo de ayuda con su significado.
- Ejemplo: `abbrAcronym.html`

Módulos de texto

- Para dar énfasis a un párrafo existen dos etiquetas:
 - Etiqueta *strong*: los navegadores muestran el texto en negrita
 - Etiqueta *em*: los navegadores muestran el texto en cursiva
 - Ejemplo: `enfasis.html`
- Etiqueta *address*: se utiliza para indicar una dirección, generalmente se muestra en cursiva.

Módulo de hipertexto

- Etiqueta *a*: representa a los enlaces
- Atributo *href*: contiene la URL a donde queremos enlazar
- Atributo *title*: descripción del sitio web al que enlazamos. Al igual que con las etiquetas *abbr* y *acronym*, el texto aparecerá al pasar el ratón por encima del link.
- Ejemplo: enlaces.html

Módulo de listas

- Existen tres tipos de listas: ordenadas, sin ordenar (o de viñetas) y de definición
- Lista ordenadas: etiqueta *ol* (“ordered list”) designa a toda la lista y cada elemento se indica mediante *li*
- Listas no ordenadas (viñetas): etiqueta `` (“unordered list”) designa a toda la lista y cada elemento se indica mediante *li*
- Listas de definición: pensadas para diccionarios, glosarios, etc. *dl* para la lista completa, *dt* el término a definir y *dd* la definición
- Ejemplo: listas.html

Módulo de imagen

- Las imágenes son un elemento importante a la hora de hacer más atractiva una web. Se pueden usar tres formatos de imagen: GIF, JPEG y PNG.
- Etiqueta **img** con atributos:
 - *src*: define la imagen que se desea mostrar teniendo en cuenta la ruta.
 - *alt*: descripción alternativa a la imagen que veremos cuando se pase el ratón sobre la misma y cuando no ha sido posible cargar dicha imagen. También se utiliza por navegadores de texto.
 - *height* y *width*: establecen el alto y ancho de la imagen (píxeles o porcentaje).
 - *longdesc*: permite establecer un enlace a una URL con una descripción de la imagen.
 - Los atributos obligatorios son “**src**” y “**alt**”
- Ejemplo: imagen.html

Módulo de tablas

- Sólo se deben utilizar para mostrar información de forma tabular
- Etiquetas:
 - *table* : Crea la tabla
 - *caption* : Título de la tabla
 - *tr* : Crea una fila de la tabla
 - *td* : Crea una celda de datos
 - *th* : Crea una celda de encabezado
 - *summary* : resumen de la tabla para que sea accesible
- Ejemplo: tabla.html

Mecanismos de estructuración

- Los elementos **“div”** y **“span”** junto a los atributos **“id”** y **“class”** constituyen un mecanismo genérico para añadir estructura a los documentos.
- Estos elementos definen el contenido en línea (**“span”**) o en bloques (**“div”**) pero no realizan imposiciones de presentación al contenido.

Ejemplo

```
<div id = "contenido">  
  ...  
</div>  
<p> <span class ="cliente"> ... </span> </p>
```

Atributos “id” y “class”

- *id*: asigna un nombre exclusivo a un elemento del documento
- *class*: se puede utilizar repetidamente en el mismo documento

Usos del atributo “id”

- 1 Utilizarse como selector de una hoja de estilo
- 2 Ancla de destino
- 3 Hacer referencia para un elemento de una secuencia de comandos basado en DOM
- 4 Nombre de un objeto declarado
- 5 Para identificar campos al extraer datos de una página a una base de datos.

Elemento “div”

- Un “div” es una unidad de marcado para crear bloques de estructura de un documento.
- Pero no debe sustituir a los elementos válidos (y más adecuados), por ejemplo `<p>` para un párrafo y no `<div class=”parrafo”>`

Referencias

- Oficina española del W3C: <http://www.w3c.es>
- Traducción de la norma XHTML 1.0: El Lenguaje de Etiquetado Hipertextual Extensible Una Reformulación de HTML 4.0 en XML1.0
<http://www.sidar.org/recur/desdi/traduc/es/xhtml/xhtml11.htm>
- Técnicas HTML para las Pautas de Accesibilidad al Contenido de la Web 1.0:
http://www.discapnet.es/web_accesible/tecnicas/html/WCAG10-HTML-TECHS_es.html