

UD 02 - UTILIZACIÓN DE LENGUAJES DE MARCAS EN ENTORNOS WEB (Parte 1 - HTML)

1. Del HTML al XHTML. Evolución y versiones

HTML es el lenguaje utilizado para crear la mayor parte de las páginas web. Es un estándar reconocido en todos los navegadores.

Evolución de sus versiones:

Origen: Sistema de hipertexto para compartir documentos electrónicos en 1980. La primera propuesta para convertir HTML en estándar se realizó en 1993. (Dos propuestas HTML y HTML+)

HTML 2.0 Primera versión oficial de HTML

HTML 3.2. Incorpora las applets de Java y texto alrededor de imágenes. 1997.

HTML 4.0. Hojas de estilos CSS y pequeños programas en webs. 1998.

HTML 4.01. Se detiene la actividad de estandarización hasta 2007.

Tras el estándar HTML 4.01 se detecta su incompatibilidad en XML y se crea el XHTML que combina la sintaxis de HTML 4.0 con la de XML.

XHTML 1.0. Primera versión (2000). Adaptación de HTML 4.01 añadiendo restricciones XML.

XHTML 1.1. Modularizar XHTML

Borrador XHTML 2.0.

HTML 5.0. Especificación actual de HTML. (Y sigue evolucionando...)

1.1. HTML5

Publicado en 2014. Especifica dos variantes de sintaxis para HTML: La variante HTML 5 y una variante conocida como sintaxis XHTML5 que debe ser servida como XML.

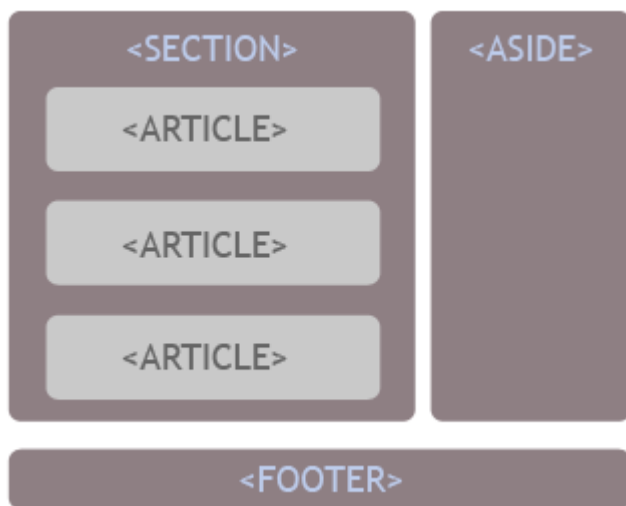
HTML5 ofrece características para desarrollo de páginas web más simples y para formas más complejas de manejo. La especificación HTML 5 se basa en XHTML 4.01 Strict pero no usa DTD.

Usa como base el Modelo de Objetos del Documento (el DOM, árbol creado por la estructura del documento) en vez de un conjunto determinado de reglas sintácticas.

En esta especificación se incluyen instrucciones detalladas de cómo los navegadores deben usar el HTML mal especificado.

Novedades principales:

- **Archivos multimedia:** Soporte integrado para el contenido multimedia gracias a los elementos `<audio>` y `<video>` ofreciendo la posibilidad de incluir contenido multimedia en HTML de forma nativa. El elemento `<canvas>` puede ser usado para dibujar gráficos.
- **Web semántica:** Se introducen etiquetas como `header`, `nav`, `section`, `aside`, `footer`, `article`, `hgroup`, `figure`... para hacer más descriptivo el lenguaje. Así el desarrollador puede definir mejor la importancia o finalidad de cada parte de la web y el código es más entendible para los buscadores.
- Los formularios también han sido mejorados con validaciones sin necesidad de Javascript, atributos nuevos y gran cantidad de elementos semánticos.



2. Estructura de un documento HTML

La estructura de HTML tiene un prólogo y un ejemplar.

Prólogo

Declaración del tipo de documento en la que se indica el tipo de documento a iniciar y la versión HTML utilizada para que lo interprete correctamente el navegador.

En HTML 4.0 hay tres prólogos distintos:

HTML 4.0 Strict. DTD usada por defecto en HTML 4.0. No se permiten los elementos declarados como deprecated por otras versiones o recomendaciones.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 //EN" "http://www.w3.org/TR/REC-html40/strict.dtd">
```

HTML 4.0 Transitional. Permite el uso de todos los elementos de HTML 4.0 Strict y los deprecated.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">
```

HTML 4.0 Frameset. Variante de HTML 4.0 Transitional para documentos que usan frames. En ellos el body es reemplazado por un frameset.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">
```

En **HTML 5.0** se simplifica el prólogo enormemente:

```
<!DOCTYPE html>
```

Ejemplar

Está delimitado por las etiquetas `<html>` y `</html>`.

El ejemplar se divide en:

- **Cabecera.** Delimitada por `<head>` `</head>`. Información sobre el título de la página, el autor, palabras clave. Es obligatorio definir el título del documento con `<title>` `</title>`. En la cabecera puede utilizarse también el elemento `<META>` que puede forzar a que la página activa se cargue cada cierto tiempo (indicado en segundos mediante el atributo CONTENT). Debe usarse con precaución porque puede sobrecargar el servidor.

```
<HEAD>
  <TITLE> Título de la página </TITLE>
  <META HTTP-EQUIV="Refresh" CONTENT="10">
</HEAD> ``
```

El elemento `BASE HREF="URL"` define la información a prefijar a todo URL incompleto en el documento. Por ejemplo, el URL relativo `"/html/test.html"` corresponderá de hecho a `"URL/html/test.html"`.

- **Cuerpo**: Contiene información que se va a presentar en pantalla. Tiene las etiquetas `<body>` y `</body>` salvo en HTML 4.0 Frameset donde se sustituyen por `<frameset>` y `</frameset>`

3. Identificación de etiquetas y atributos de HTML

- El documento HTML tiene etiquetas y atributos.
- La cantidad de etiquetas de HTML **está limitada** a las existentes en el lenguaje (a diferencia de XML).
- La información adicional se añade en los atributos (que tienen un conjunto de valores definidos).
- Si no es un valor válido, el navegador lo ignora. Muchos de los atributos son comunes a muchas etiquetas.

3.1. Clasificación de los atributos comunes según su funcionalidad

Podemos distinguir:

Atributos básicos: Se pueden usar en casi todas las etiquetas HTML

`name`: Asignar nombre al elemento HTML

`title`: Asignar título al elemento HTML, mejorando la accesibilidad (navegadores lo muestran cuando se pasa por encima)

`id`: Identificar al elemento HTML de forma única

`style`: Aplicar estilo al elemento HTML

`class`: Aplicar al elemento HTML definido en CSS.

Los atributos `id` y `class` solo pueden contener guiones medios, guiones bajos, letras o números pero no pueden empezar por números. Se distingue entre mayúsculas y minúsculas., No se recomiendan caracteres no ingleses, acentos...

Atributos para internacionalización: Para webs multiidioma

`dir`: Dirección del texto (ltr, de izquierda a derecha valor por defecto; rtl, de derecha a izquierda)

`lang`: Código de idioma.

`xml:lang`: Código según recomendación RFC 1766

En XHTML `xml:lang` tiene más prioridad que `lang` y es obligatorio incluirlo siempre que se incluya `lang`

Atributos de eventos y atributos para los elementos que pueden obtener el foco: No lo cubre la asignatura.

3. Elementos HTML

El elemento HTML está formado por etiqueta de apertura, cero o más atributos, texto encerrado por la etiqueta (opcional), etiqueta de cierre.

Podemos distinguir entre:

- **Elemento en línea**: Solo ocupa el espacio necesario para mostrar sus contenidos (pueden ser texto u otros elementos en línea). Ej.: Enlaces, texto en negrita.
- **Elemento 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 allí. El contenido puede ser texto, elementos en línea u otros elementos de bloque. Ej.: Encabezados, párrafos

Hay elementos que pueden tener un comportamiento en línea o en bloque según las circunstancias.

3.1. Estructura básica del documento

`html`

`head`

`body`: Permite definir formatos del documento que se aplican a los elementos de la página de forma global (color del fondo, márgenes, color de enlaces)

3.2. Sección de cabecera

Contenedores

Elemento	Descripción
-----	-----
<code>title</code>	Título del documento

`script`	Script referenciado o incrustado. Se puede agregar dentro de `head` y de `body`
`style`	Estilo aplicado usando CSS

****No contenedores****

Elemento	Descripción
-----	-----
`base`	URL base que se usará para las URL relativas contenidas en el documento
`link`	Añadir información externa afín al documento (CSS), ayuda a navegación, RSS, información de contacto
`meta`	Términos por los que debe ser encontrada ` <meta >`<="" content="Free Web tutorials" name="description" td=""/>

3.3 Formato al texto de un párrafo

Elemento	Descripción
-----	-----
`p`	Párrafo
`h1`	Encabezado de nivel i (Entre 1 y 6). El tamaño de la letra es mayor a menor i. No deben usarse estas etiquetas para formatear texto. Solo para títulos de párrafos.
`b`	Negrita
`i`	Cursiva
`u`	Subrayado
`sup`	Superíndice
`sub`	Subíndice
`strong`	Elemento resaltado. Normalmente en negrita aunque podrían ponerlo en cursiva y en naranja.

3.4. Listas

Elemento	Descripción
-----	-----
`ul`	Lista desordenada (viñetas)
`ol`	Lista ordenada (numeritos de orden)
`li`	Cada elemento de una lista
`dl`	Lista de definición (Tiene sangría)
`dt`	Cada término de una lista de definición
`dd`	Cada definición de una lista de definición

```
```html
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">
```

```
<html>
```

```
 <head>
```

```
 <title>Listas</title>
```

```
 </head>
```

```
 <body>
```

```
 <h3>Ejemplo de lista desordenada: Modulos de 1Âº de ASIR</h3>
```

```
 Fundamentos de Hardware
 Gestión de Bases de Datos

```

```
 <h3>Ejemplo de lista ordenada: Modulos de 1Âº de ASIR</h3>
```

```
 Fundamentos de Hardware
 Gestión de Bases de Datos

```

```
 <h3>Ejemplo de lista de definición: Modulos de 1Âº de ASIR</h3>
```

```
 <dl> <dt>Fundamentos de Hardware</dt>
 <dd>Componentes físicos de un ordenador</dd>
 <dt>Gestión de Bases de Datos</dt>
```

```

 <dd>Diseño y uso de bases de datos relacionales</dd>
 </dl>
</body>
</html>

```

## 3.5. Tablas

Elemento	Descripción
<code>table</code>	Contenido de una tabla
<code>tr</code>	Cada línea de la tabla
<code>td</code>	Cada celda de la tabla
<code>colgroup</code>	Para agrupar columnas
<code>tbody</code>	Para agrupar líneas de la tabla
<code>thead</code>	Línea cabecera de la tabla
<code>th</code>	Cada celda de la cabecera
<code>tfoot</code>	Fila pie de la tabla

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
 <head>
 <title>Tablas</title>
 </head>

 <body>
 <h3>Ejemplo de tabla: Notas del môdulo de LMSGI de 1º de ASIR</h3>
 <table border = "1">
 <thead> <tr> <th>Eval</th> <th>LMSGI</th> </tr> </thead>
 <tfoot align="center"> <tr> <td>Media</td> <td>8</td> </tr> </tfoot>
 <tbody align="center">
 <tr> <td>Primera</td> <td>6</td> </tr>
 <tr> <td>Segunda</td> <td>7</td> </tr>
 <tr> <td>Tercera</td> <td>8</td> </tr>
 </tbody>
 </table>
 </body>
</html>

```

## 3.6. Formularios

Elemento	Descripción
<code>form</code>	Contenido de un formulario (Solicitar información al usuario)
<code>input</code>	Caja de texto para texto corto. Según el atributo <code>type</code> puede introducirse un texto sin más, un campo en el que al escribir lo que se visualicen sean asteriscos, un boton de radio.. <div> <code>text</code>  <code>password</code>  <code>email</code>  <code>checkbox</code>  <code>radio</code>  <code>submit</code>  <code>reset</code>  <code>date</code>  <code>submit</code> (habria que decirle el method o lo mandará todo por GET)....         </div>
<code>textarea</code>	Caja de texto para texto largo
<code>select</code>	Menú desplegable para elegir de una lista de opciones

Elemento	Descripción
<code>option</code>	Cada una de las opciones del desplegable
<code>button</code>	Un botón. Así puede introducirse en el botón cualquier otro elemento HTML como imágenes (en el creado con <code>input</code> no se puede eso)
<code>fieldset</code>	Agrupar elementos de un formulario
<code>legend</code>	Título al fieldset
<code>label</code>	Etiqueta del campo del formulario

### 3.7. Frames (En desuso por completo)

Elemento	Descripción
<code>frameset</code>	Partición de la ventana del navegador en marcos (files o columnas), Para partirlo en filas y columnas deben anidarse frames
<code>frame</code>	Define marco que contiene información

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">

<html>
 <head> <title>Frames</title> </head>

 <frameset rows="20%,30%,*">
 <frame src="LMSGI02_Ejemplo07_1.html" /> <frame src="LMSGI02_Ejemplo03.html" />
 <frame src="LMSGI02_Ejemplo05.html" />
 </frameset>
</html>

```

### 3.8. Capas o divs

- Las capas son páginas que se pueden incrustar dentro de otras: Bloques con contenido HTML que pueden situarse en la página de forma dinámica.
- Los atributos de una capa pueden y deben definirse dentro de una hoja de estilo (posición, visibilidad..). El contenido debe estar en la parte principal de la página.
- Las capas tienen sentido cuando se les pueden aplicar estilos CSS
- Las capas pueden ser movidas y modificadas desde un lenguaje de script (es lo más importante) Aunque antes era difícil que las implementaciones de los distintos navegadores -incompatibles entre sí- permitiesen un código que funcionase de la misma forma en todos ellos.

Elemento	Descripción
<code>div</code>	Sección dentro del documento XHTML
<code>id</code>	Identificador único
<code>style</code>	Estilo al contenido
<code>class</code>	Estilo al contenido

### 3.9. Otros elementos

Elemento	Descripción
<code>a</code>	Enlace a página web
<code>img</code>	Insertar imagen en página web
<code>abbr</code>	Forma abreviada
<code>acronym</code>	Acrónimo
<code>blockquote</code>	Texto con sangría
<code>q</code>	Cita. Se añaden marcas de citación

Elemento	Descripción
<code>br</code>	Línea en blanco.

```
Mi Google

El <abbr title="señor">Señor</abbr>
El <acronym title="Partido Socialista Obrero Español">PSOE</acronym>
```

## 4. XHTML frente a HTML

- Lenguaje XHTML es similar al HTML. De hecho, es adaptación de HTML a XML.
- Estándar XHTML 1.0 añade pequeñas mejoras y modificaciones a HTML 4.01 por lo que pasar de HTML 4.01 a Strict no requiere ningún cambio casi.
- XHTML no tiene sintaxis tan permisiva. añade normas a la forma de escribir etiquetas y atributos.

### 4.1. XHTML: Diferencias sintácticas y estructurales con HTML

El esquema básico del documento para cumplir la especificación debe cumplir:

- Debe haber declaración DOCTYPE en el prólogo del documento y su identificador público debe hacer referencia a alguna de las tres DTD definidas por el W3C:

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

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

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

- El elemento raíz debe ser `<html>`
- El elemento raíz debe indicar el espacio de nombres XHTML usando `<html xmlns="http://www.w3.org/1999/xhtml">`
- Las etiquetas se tienen que cerrar en el orden inverso al que se abren, nunca pueden solaparse
- Nombres de etiquetas y atributos siempre en minúsculas
- Valor de atributos siempre entre comillas y con algun valor
- Todas las etiquetas deben cerrarse siempre (permitido `<br/>`)
- 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 blancos sobrantes dentro del valor del atributo.
- El código Javascript debe encerrarse entre etiquetas especiales (`<![CDATA[ ]]>`) para evitar que el navegador interprete de forma errónea caracteres como & y <.
- Las páginas XHTML no deben usar el atributo `name` sino el atributo `id`
- En XHTML debe separarse el formato del contenido. Los párrafos deben separarse consistentemente y las cabeceras h1-h6 solo deben usarse para destacar los apartados. Debe darse formato usando CSS.

Introducción a XHTML: <https://uniwebsidad.com/libros/xhtml1?from=librosweb>

### 4.2. Ventajas e inconvenientes de XHTML sobre HTML

Ventajas:

- Compatibilidad parcial con navegadores antiguos. Información visualizada aunque sin formato
- Un mismo documento puede adoptar diseños distintos en diferentes apartados
- Sencillo de editar y mantener

- Compatible con los estándares que desarrollaba (en su día) W3C como recomendale para futuros navegadores (agentes de usuario)
- Documentos XHTML 1.0 tienen (en su día tenían) mejor rendimiento en las herramientas web
- Separación de contenidos y presentación los hace más adaptables
- Se pueden usar herramientas para procesar XML genéricos (editores, XSLT,..)

Inconvenientes:

- Navegadores antiguos no son totalmente compatible con los estándares.
- Muchas herramientas de diseño web no generan código XHTML correcto.

## 5. Herramientas de diseño web

Hay herramientas de edición de código, de software de diseño web y recursos.

**Adobe Color CC:** Mejora el color de las fotografías. Analiza el tono de las ilustraciones y forma soluciones armónicas. Gama de tonalidades para usar. Ajustes de color, tono, saturación, curvas, máscaras de luminosidad. Gratuita y fácil de usar. Disponible en Adobe InDesign y Adobe AfterEffects. Accesible también desde Photoshop.

**Google Fonts:** Fuentes originales y adecuadas. Catálogo gratuito con más de 800 opciones. Sección Featured con colecciones temáticas creada por la misma herramienta.

**Whatfontis:** Nos dice qué fuente era. Subiendo una imagen o una URL a su sitio. Permite editar dicha imagen y obtener mejores resultados. Tiene comunidad de usuarios.

**Mockflow:** Construir esquemas de diseño de forma amplia. Arrastrar y soltar para colocar los elementos con iconos preconfigurados.

**Tinypng:** Optimizar imágenes comprimiendo contenidos gráficos para evitar la lentitud. Las optimiza de forma automática.

**Adobe Edge Reflow:** Creadores de diseño web responsive. Crea prototipos web funcionales a través de herramientas. Adaptable a cualquier resolución. Cumple HTML5 y CSS3.

**Notepad++:** Identifica expresiones, optimiza rutinas, uso sencillo. También SublimeText existe.

**Pixabay:** Banco de imágenes que puede usarse de forma gratuita.

**Ceros:** Para crear piezas de contenido diseñando infografías, ebooks, banners, micrositos, revistas..

**Canva:** Creación de imágenes, cabeceras para blogs y web. Creación de publicaciones en redes. Sencilla, gratuita y online.