

HTML y XHTML. Lenguajes para la Web

CONTENIDO

Evolución histórica
Estructura del documento
Elementos de HTML
Contenido de la cabecera
Contenido del cuerpo
XHTML
HTML 5
Validación
Ejercicios propuestos

2.1. Evolución histórica

HTML (HyperText Markup Language) aparece a principios de los años noventa cuando el investigador Tim Berners Lee, lo desarrolla para potenciar la colaboración entre físicos e investigadores de la energía nuclear.

El objetivo principal era enlazar los documentos mediante links (hipervínculos), de forma que al hacer clic sobre algún texto, se abriera otro documento relacionado con la información seleccionada.

En 1991 se publica el primer documento formal con el nombre de “HTML tags”. Esta primera versión tiene muchas limitaciones, por ello organismos internacionales colaboraron para desarrollar mejoras en el lenguaje.

En 1993 hubo un primer intento de realizar una especificación oficial de HTML, sin embargo no fue reconocida. Se trataba de la versión HTML 1.2.

En 1995 se publica el primer estándar oficial llamado HTML 2.0 por el organismo IETF (Internet Engineering Task Force).

En 1997 se publica la versión HTML 3.2 por la organización W3C (World Wide Web Consortium), que será a partir de ahora y hasta nuestros días, el organismo oficial encargado de publicar los estándares oficiales. Incluye tablas y applets de Java.

En 1998 aparece la versión HTML 4.0 que incluye muchas mejoras como soporte para lenguajes script, hojas de estilo css, facilidades de impresión, accesibilidad para discapacitados, etc. En este mismo año aparece también la primera especificación de XML. Al año siguiente se publica HTML 4.01 la última especificación oficial hasta el momento, se trata de una mera revisión que no incluye novedades significativas.

En el año 2000 el W3C publica XHTML 1.0, se trata de una reformulación de HTML 4.0 basada en XML. Cumpliendo unas directrices de compatibilidad puede funcionar sobre aplicaciones creadas para HTML, pero a la vez es conforme a XML, lo que le proporciona gran flexibilidad para añadir nuevos elementos y módulos.

XHTML 1.0 se ha convertido en un estándar aceptado y ampliamente utilizado por la comunidad de desarrolladores Web.

W3C siguió trabajando por este camino en un nuevo lenguaje XHTML 2.0, pero que ya no era compatible ni con HTML ni siquiera con XHTML 1.0.

En el año 2003 aparecieron los famosos XFORMS, los formularios de XHTML que mejoran considerablemente a los anteriores, gracias a que separan perfectamente el contenido de la presentación. Esta característica les permite funcionar sobre un portátil, teléfono móvil, agenda electrónica, etc.

Los XFORMS despertaron un renovado interés por desarrollar HTML en lugar de sustituirlo, ya que XML dependía de la implementación en los navegadores de nuevas tecnologías como RSS o Atom.

En el año 2004 las empresas Apple, Mozilla y Opera crearon un grupo de trabajo llamado WHATWG con el objetivo de continuar el desarrollo de HTML.

Una de sus premisas de trabajo era mantener total compatibilidad con versiones anteriores.

En el 2007 los grupos de trabajo W3C y WHATWG deciden unirse para el desarrollo de una especificación de HTML 5, lo que permite publicar a principios de 2008 el primer borrador de HTML 5.

En el 2009 el consorcio W3C decide poner fin al desarrollo del estándar XHTML 2.0 a favor de HTML 5.

En la actualidad siguen colaborando juntos y se prevé una especificación oficial en breve.

AÑO	EVENTO
1991	Nacimiento. "HTML tags"
1993	HTML 1.2
1995	HTML 2.0
1994	Se funda W3C
1997	HTML 3.2
1998	HTML 4.0 y XML
1999	HTML 4.01
2000	XHTML 1.0
2001	XHTML 1.1
2002	XHTML 2.0
2003	XFORMS
2004	Se funda WHATWG
2008	Borrador de HTML 5

Tabla 2.1: Evolución del lenguaje HTML

HTML 4.01 y XHTML 1.0 son las especificaciones más utilizadas.

HTML 5 se va incorporando progresivamente.

La decisión de elegir uno u otro lenguaje depende de varios factores, entre ellos el tamaño y el número de usuarios del sitio Web.

Solo como orientación:

- Para grandes empresas y sitios comerciales con mucho público, sería aconsejable usar XHTML y CSS avanzado en varias hojas de estilos.

- Para sitios Web pequeños y páginas personales se puede utilizar HTML por su simplicidad y una hoja de estilos sencilla.

2.2. Estructura del documento

Los documentos HTML son archivos de texto plano formados por etiquetas de apertura y cierre (por ejemplo: `<html>... </html>`) que permiten dar formato al contenido del documento.

Una página web es el resultado que ofrece un navegador al interpretar dicho documento.

La estructura básica de un documento HTML es:

`<! Declaración de Tipo de Documento >`

`<html>`

`<head >`

`<title>` Título de la página `</title>`

Elementos opcionales de la cabecera

`</head>`

`<body>`

Contenido del cuerpo

`</body>`

`</html>`

Tanto la declaración de tipo de documento, como las etiquetas `html`, `head` y `body` son opcionales, pero deben incluirse para que el documento sea correcto y legible.

La etiqueta `<html>` es la raíz del documento por lo que `head` (cabecera) y `body` (cuerpo) deben estar totalmente contenidas en ella.

Además `head` y `body` no deben solaparse entre sí, por lo tanto, el documento debe tener un gran bloque principal (`html`) y dos subbloques al mismo nivel.

Estos dos bloques a su vez, pueden tener otros subbloques que tampoco deben solaparse.

Es buena costumbre separar los bloques importantes por líneas de comentarios.

La declaración de tipo de documento se coloca al principio, fuera de todo bloque.

En unidades posteriores estudiaremos las DTD (Document Type Definition - Definición de Tipo de Documento), por ahora es suficiente saber que se trata de un documento externo con extensión `.dtd` y que existen 3 posibilidades: estricta, de transición y con marcos.

Al indicar la DTD al principio del documento le decimos al navegador cómo interpretar adecuadamente el código.

HTML 4.01 recomienda separar por completo el contenido del formato.

DTD estricta:
<p>Si cumplimos escrupulosamente con esta recomendación, almacenando toda la información sobre el formato en una hoja de estilos CSS, entonces podemos utilizar la declaración estricta:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN" "http://www.w3.org/TR/html4/strict.dtd"></pre>
DTD de transición:
<p>Si queremos combinar contenido e información de formato en nuestro documento HTML, entonces debemos utilizar la declaración de transición:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></pre>
DTD con marcos:
<p>Si queremos incluir marcos en nuestro documento, entonces tenemos que usar una DTD que es muy similar a la anterior, pero que sustituye el elemento body por frameset, en este caso la declaración sería:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></pre>

En HTML 5 la declaración es más simple: `<!DOCTYPE html>`

Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Mi primera página</title>
  </head>
  <body>
    <h1>Hola Mundo</h1>
  </body>
</html>
```

Actividad 2.1:

Copia este código en un archivo de texto, guárdalo con el nombre mipagina.html y luego observa el resultado en el navegador.

2.3. Elementos de HTML

Los elementos son los componentes básicos del lenguaje HTML. Cada elemento puede tener atributos y contenido. La mayoría de elementos tienen una etiqueta de inicio, una etiqueta de fin y un contenido encerrado entre ambas. Si no tienen contenido se les llama vacíos.

Los atributos de un elemento (casi siempre opcionales) se colocan en la etiqueta de apertura.

Sintaxis.

- La sintaxis básica para un elemento no vacío tiene la forma:

```
<nombre_elemento atributo1="valor1" atributo2="valor2"...>
    contenido del elemento
</nombre_elemento>
```

- La sintaxis para un elemento vacío (V):

```
<nombre_elemento atributo1="valor1" atributo2="valor2"... >
```

2.3.1. <html>

Definición: delimita el contenido del documento.

Aparición: apertura y cierre opcionales.

Atributos: **dir**, **lang**, **version**.

- **dir**: indica el sentido de lectura de texto cuando no corresponde al habitual (árabe).
Valores: ltr (de izquierda a derecha) y rtl (de derecha a izquierda).
- **lang**: indica el idioma por defecto del documento.
Valores: de (alemán), en (inglés), es (español), fr (francés), etc.
- **version**: indica la versión del lenguaje HTML, está desaprobado porque ya se proporciona en la declaración **<!doctype>**.

Contenido de html: **<head>** y **<body>**.

Ejemplo:

```
<html lang="es" dir="ltr">
    <head>...</head> <body>...</body>
</html>
```

Actividad 2.2:

Indica que mipagina.html está en español y con dirección rtl, observa el resultado, después deja la dirección definitiva en ltr.

2.3.2. <head>

Definición: delimita la cabecera de un documento.

Aparición: apertura y cierre opcionales.

Atributos: **lang**, **dir**, **profile**.

- **profile**: indica donde están los perfiles para interpretar los elementos meta.

Valores: URI (Uniform Resource Identifier – Identificador Uniforme de Recurso), dirección web completa del recurso.

Contenido de la cabecera: **<title>**, **<base>**, **<meta>**, **<link>**, **<object>**, **<script>** y **<style>**.

Todos los elementos son opcionales excepto **<title>** que es obligatorio.

El contenido de la cabecera es una información general de todo el documento, por eso se escribe al principio y es lo primero que analiza el navegador.

2.3.3. <body>

Definición: delimita el cuerpo de un documento.

Aparición: apertura y cierre opcionales.

Atributos: **lang**, **dir**, **id**, **class**, **title**, **style**, **onload**, **onunload** y los eventos intrínsecos.

- **id**: sirve para asignar un nombre único a un elemento.

Se utiliza para identificar elementos en hojas de estilo y en scripts.

- **class**: sirve para asignar un nombre de clase a un elemento.

Es posible usar el mismo nombre de clase para varios elementos.

Se usa para mejorar el rendimiento de las hojas de estilo.

- **title**: sirve para agregar un comentario asociado a un elemento.

Los navegadores muestran este comentario en una ventana emergente cuando el ratón se sitúa sobre el elemento.

- **style**: sirve para aplicar información de estilo a un elemento.

- **onload**: este evento ocurre cuando el navegador finaliza la carga de un documento.

- **onunload**: este evento ocurre cuando el navegador elimina un documento de una ventana o marco.

Atributos desaprobadados: **background**, **text**, **link**, **alink**, **vlink**, **bgcolor**.

Estos atributos no deben usarse, ya que la forma correcta es a través de hojas de estilo o con estilos en línea como en el siguiente ejemplo.

Ejemplo:

```
<body style="background-color: blue"> ... </body>
```

Actividad 2.3:

Cambia el color de fondo con el atributo `style` y con el atributo desaprobado `bcolor`.

Comprueba que se obtiene idéntico resultado.

2.3.4. Comentarios `<!-- -->`

Definición: sirve para insertar una o varias líneas con comentarios que no serán interpretadas por el navegador. No se pueden anidar.

Ejemplo:

```
<!--mipagina.html-->
<html>
  <head> ... </head>
  <!-- Aquí empieza el cuerpo del documento -->
  <body> ... </body>
</html>
```

Actividad 2.4:

Agrega comentarios a `mipagina.html` para separar las partes del documento.

2.3.5. Conjuntos de atributos

Los atributos de los elementos `html` se organizan en grupos, de modo que se usarán los siguientes conjuntos cada vez que se describa un elemento, para evitar poner la lista completa.

`%i18n = lang, dir`

`%coreattrs = id, class, style, title`

`%events = onload, onunload, onblur, onchange, onfocus, onreset, onselect, onsubmit, onabort, onkeydown, onkeyup, onkeypress, onclick, ondblclick, onmousedown, onmouseover, onmouseout, onmousemove, onmouseup`

`%attrs = %i18n + %coreattrs + %events`

Listado de eventos intrínsecos (`%events`) y su descripción:

Eventos de `<body>` y `<frameset>`

Atributo	Descripción
<code>onload</code>	Ocurre cuando termina la carga completa de un documento
<code>onunload</code>	Ocurre cuando el navegador elimina un documento

Eventos de **formulario**

Atributo	Descripción
onblur	Ocurre cuando un campo pierde el foco
onchange	Ocurre cuando un campo pierde el foco y su valor se modifica después
onfocus	Ocurre cuando un campo recibe el foco
onreset	Ocurre cuando se reinicia un formulario
onselect	Ocurre cuando se selecciona un texto de un campo de texto
onsubmit	Ocurre cuando se envía un formulario

Eventos de **imagen**

Atributo	Descripción
onabort	Ocurre cuando la carga de una imagen se interrumpe

Eventos con el teclado

Atributo	Descripción
onkeydown	Ocurre cuando se pulsa una tecla encima de un elemento
onkeypress	Ocurre cuando se pulsa y se suelta una tecla encima de un elemento
onkeyup	Ocurre cuando se suelta una tecla encima de un elemento

Eventos con el ratón

Atributo	Descripción
onclick	Ocurre cuando se hace click sobre un elemento
ondblclick	Ocurre cuando se hace doble click sobre un elemento
onmousedown	Ocurre cuando se pulsa el botón del ratón encima de un elemento
onmousemove	Ocurre cuando el ratón se mueve estando encima de un elemento
onmouseout	Ocurre cuando el ratón sale de un elemento
onmouseover	Ocurre cuando el ratón se sitúa sobre un elemento
onmouseup	Ocurre cuando el botón del ratón se suelta estando encima

(Válidos para todos los elementos excepto <base>, <bdo>,
, <frame>, <frameset>, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> y <title>.)

Para utilizar estos eventos intrínsecos hay que conocer un lenguaje script, que sirva para ejecutar una determinada acción cuando se produzca el evento.

Ejemplo:

```
<body onload="alert('Carga completa');">
```

(Observación: asegúrate de utilizar el carácter comilla simple del teclado)

2.4. Contenido de la cabecera

Elementos:

`<title>`, `<base>`, `<meta>`, `<link>`, `<object>`, `<script>` y `<style>`.

2.4.1. `<title>`

Definición: indica el nombre del sitio Web en la barra superior o pestaña del navegador.

Aparición: etiquetas de apertura y cierre obligatorias.

Atributos: `lang`, `dir`.

Ejemplo:

```
<head>
  <title> Mi primera página </title>
</head>
```

2.4.2. `<base>`

Definición: indica la dirección raíz del sitio Web, la cual permite resolver las direcciones relativas.

Aparición: sin etiqueta de cierre (V).

Atributos: `href`.

- `href`: se usa siempre para indicar la dirección raíz del documento.

Ejemplo:

```
<base href="http://www.juanmacr.es/">
```

En caso de tener una dirección relativa como `"/ASIR.html"`, se completaría dando lugar a la dirección absoluta: `"http://www.juanmacr.es/ASIR.html"`.

Actividad 2.5:

- Elige un tema para tu sitio Web diferente del resto de compañeros.
 - Date de alta en un hosting gratuito.
- (Si prefieres un hosting de pago tendrás que elegir también tu dominio)
- Crea un nuevo espacio.
 - Sube el contenido de `mipagina.html` y renómbrala como `index.html`
 - En `title` indica el nombre de tu empresa.
 - Comprueba que tu sitio está publicado en Internet.
-