

UT2-Utilización de lenguajes de marcas en entornos web.

2º curso DAM distancia 22-23

Contenidos - 1ª parte - HTML

1.- Introducción a la utilización de lenguajes de marcas en entornos web.

1.1.- HTML: evolución y versiones.

1.2.- HTML y XHTML.

1.3.- Estructura de un documento HTML.

1.4.- Identificación de etiquetas y atributos HTML.

1.5.- Clasificación de los atributos comunes según su funcionalidad.

2.- Elementos HTML.

2.1.- Elementos de la estructura básica de un documento.

2.2.- Elementos de la sección de cabecera.

2.3.- Encabezados y párrafos.

2.3.1.- Saltos de línea y espacios en blanco.

2.3.2.- Comentarios.

2.4.- Semántica a nivel de texto.

2.5.- Elementos de listas.

2.6.- Elementos de tablas.

Contenidos - 1ª parte - HTML

2.7.- Elementos de formularios.

2.7.1.- Declaración de formulario

2.7.2.- Campos de formulario - input

2.7.3.- Campos de formulario - Área de texto

2.7.4.- Campos de formularios - Lista desplegable

2.7.5.- Campos de formulario - checkbox

2.7.6.- Botones de formulario

2.7.7.- Otros campos de formulario

2.7.8.- Campos de formulario con formato de fecha

2.7.9.- Campos de formulario - Rangos

2.7.10.- Organización de formularios

2.7.11.- Un ejemplo de formulario

2.8.- Multimedia.

2.9.- Secciones y etiquetas semánticas.

2.10.- Elemento iframe.

2.11.- Validación HTML.

1.- Introducción a la utilización de lenguajes de marcas en entornos web.

1.1.- HTML: 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, por lo tanto, todos ellos visualizan una página **HTML** de forma muy similar independientemente del sistema operativo sobre el que se ejecutan.

El origen de HTML fue un sistema de hipertexto para compartir documentos electrónicos en 1990. La primera propuesta oficial para convertir **HTML** en un estándar se realizó en 1993. Ninguna de las dos primeras propuestas de estándar que se hicieron (**HTML** y **HTML+**) consiguieron convertirse en estándares oficiales.

- **HTML 2.0** fue la primera versión oficial de HTML. El IETF publicó el estándar en septiembre de 1995.
- **HTML 3.2** se publicó el 14 de Enero de 1997, por el W3C. Incorporaba los applets de Java y texto alrededor de las imágenes.
- **HTML 4.0** se publicó el 24 de Abril de 1998. Entre las novedades que presentaba se encontraban las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- **HTML 4.01** se publicó el 24 de diciembre de 1999, como actualización de la versión anterior. En ese momento el W3C detuvo la actividad de estandarización de HTML hasta marzo de 2007, momento en que se retoma debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de **HTML 5.0**, que será la siguiente versión de este lenguaje.
- **HTML 5** es la versión más avanzada y la que se considera estándar actualmente, aunque ha ido evolucionando por diferentes especificaciones (HTML 5.1, HTML 5.2 y HTML 5.3). Es una versión viva en la que se sigue trabajando y su estándar actual (HTML 5.3) fue definido el 18 de octubre de 2018. Puedes encontrar las especificaciones de la última versión de HTML en el siguiente [enlace](#).

Uno de los cambios más importantes en HTML 5 es que todo lo relativo a la presentación del documento se pasa del HTML a las hojas de estilo CSS. Es decir, cuestiones como colores, fondos, tamaño o posicionamiento se especifican mediante CSS. El HTML se encarga de la información que se quiere mostrar, su estructura y su semántica. Esto hace que muchos elementos y atributos de HTML 4 hayan quedado obsoletos (****, **<center>**, **align**,...).

1.2.- HTML y XHTML.

El lenguaje **XHTML** (**EX**tensible **HyperText Markup Language**) es muy similar al lenguaje **HTML**. De hecho, no es más que una adaptación de **HTML** al lenguaje **XML**.

El estándar **XHTML 1.0** sólo añade pequeñas mejoras y modificaciones menores al estándar **HTML 4.01**, por lo que este último está prácticamente incluido en el primero. Pasar del **HTML 4.01 Strict** a **XHTML** no requiere casi ningún cambio.

El lenguaje **HTML** tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas, por ejemplo, pueden escribirse en mayúsculas, minúsculas e incluso combinando mayúsculas y minúsculas. El valor de los atributos de las etiquetas se pueden indicar con o sin comillas. Además, el orden en el que se abrían y cerraban las etiquetas no era importante.

La flexibilidad de **HTML** da lugar a páginas con un código desordenado, difícil de mantener y muy poco profesional.

XHTML soluciona estos problemas añadiendo ciertas normas en la forma de escribir las etiquetas y atributos. Son las normas que vimos en el capítulo 1 para que un fichero XML se considera bien formado. Así se consigue:

- Sencillez a la hora de editar y mantener el código.
- Al ser más regular, es más fácil escribir código que lo procese.
- Como es **XML** se pueden utilizar fácilmente herramientas creadas para procesar documentos **XML** genéricos (editores, **XSLT**, etc.).

1.2.- HTML y XHTML.

Diferencias sintácticas y estructurales con HTML

Para que el código HTML se pueda considerar XML bien formado, y por tanto, XHTML, tiene que cumplir:

- Toda la página debe estar contenida en un (único) elemento raíz, **<html>**.
- Los nombres de las etiquetas y atributos siempre se escriben en minúsculas.
- El valor de los atributos, incluso los numéricos, siempre se encierra entre comillas.
- En los atributos en los que el nombre coincide con su valor, no puede darse el valor por entendido, es decir, no se pueden comprimir. Este tipo de atributos no son muy habituales.
- Todas las etiquetas deben cerrarse siempre. **XHTML permite** que en lugar de abrir y cerrar de forma consecutiva la etiqueta (

) se puede utilizar la sintaxis **
** para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

En general, los diseñadores web suelen trabajar con HTML. El XHTML es más apreciado por los desarrolladores, que aprecian la regularidad adicional. De cualquier manera, los tres primeros puntos de la anterior lista se consideran una buena práctica y se suelen cumplir siempre.

Por otro lado, hay que tener en cuenta que los navegadores no procesan HTML y XHTML exactamente igual. En concreto, en caso de errores para HTML el navegador intentará mostrar el mayor contenido posible. Pero si el documento se sirve como XHTML (con un tipo MIME XML), cualquier error de sintaxis causa que no se muestre el documento.

1.3.- Estructura de un documento HTML.

La estructura de una página HTML ha de ser coherente con la que hemos visto en el tema anterior para cualquier documento XML. Por ello tendrá un prólogo y un ejemplar.

Prólogo

Todo documento HTML ha de tener una declaración del tipo de documento donde se le indica al navegador el tipo de documento que se va a iniciar y la versión de HTML utilizada para la codificación del mismo y, además, le permite interpretarlo correctamente.

Para la versión **HTML 4.0**, hay **tres prólogos distintos** que definen tres tipos de documentos HTML:

- **HTML 4.0 Strict.** Es la DTD utilizada por defecto con HTML 4.0. En estos documentos no se permite el uso de los elementos declarados deprecated en otras versiones o Recomendaciones HTML. La declaración del tipo de documento correspondiente es:

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


1.3.- Estructura de un documento HTML.

- **HTML 4.0 Transitional.** Permite el uso de todos los elementos que permite el HTML 4.0 Strict, además de los elementos deprecated. La declaración del tipo de documento correspondiente es:

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

- **HTML 4.0 Frameset.** Es una variante de HTML 4.0 Transitional para documentos que usan frames. En estos documentos el elemento body hay que reemplazarlo por un elemento frameset. La declaración del tipo de documento correspondiente es:

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

En el caso de **HTML5** solo existe una declaración de tipo de documento:

```
<!DOCTYPE html>
```

1.3.- Estructura de un documento HTML.

Ejemplar

En un documento HTML está delimitado por las etiquetas **<html>** y **</html>**. El ejemplar puede, a su vez dividirse en dos partes:

- La **cabecera**, delimitada por las etiquetas **<head>** y **</head>**. Contiene la información sobre el título de la página, el autor, palabras clave, etc. Dentro de esta sección es obligatorio definir el título del documento, para ello se usan las etiquetas **<title>** **</title>**. Esta información no se presentará en la ventana del navegador, salvo el título, que aparecerá en la barra de título de la parte superior.
- El **cuerpo**, contiene la información que se va a presentar en la pantalla. Está limitado por las etiquetas **<body>** y **</body>**, salvo en los documentos de tipo **HTML 4.0 Frameset** donde éstas se sustituyen por **<frameset>** y **</frameset>**.

1.4.- Identificación de etiquetas y atributos HTML.

Un documento HTML está formado por etiquetas y atributos.

Al igual que en **XML** las etiquetas pueden ser de apertura, <etiqueta>, o de cierre, </etiqueta>. Una de las diferencias con **XML** es que la cantidad de etiquetas de **HTML** está limitada a aquellas que están definidas por el lenguaje.


Aunque **HTML** define una gran cantidad de etiquetas, estas no son suficientes para crear páginas complejas ya que la definición completa de ciertos elementos, como las imágenes y los enlaces, requiere información adicional. Como no es posible crear una etiqueta por cada elemento diferente, se añade la información adicional a las etiquetas mediante los atributos, dando lugar a los elementos.

Para cada uno de los atributos hay definido un conjunto de valores que se le puede asignar. Si el valor de un atributo no es válido, el navegador lo ignora.

Cada una de las etiquetas HTML define los atributos que puede utilizar, aunque algunos de ellos son comunes a muchas etiquetas.

1.5.- Clasificación de los atributos comunes según su funcionalidad.

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

Atributo	Descripción
name = "texto"	Permite asignar el nombre "texto" a un objeto HTML
title = "texto"	Asigna un título a un elemento HTML, mejorando así la accesibilidad. Dicho título es mostrado por los navegadores cuando el usuario pasa el ratón por encima del elemento. Es especialmente útil con los elementos: a , link , img , object , abbr y acronym
id = "texto"	Permite identificar al elemento HTML sobre el que se aplica de forma única mediante el identificador "texto". Sólo es útil cuando se trabaja con <u>CSS</u> y con  <u>Javascript</u> . No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.
style = "texto"	Permite aplicar al elemento HTML el estilo "texto" directamente.
class = "texto"	Permite aplicar al elemento HTML el estilo "texto" definido en las CSS. No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.

1.5.- Clasificación de los atributos comunes según su funcionalidad.

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																
dir	Indica la dirección del texto por lo que sólo puede tomar dos valores: ltr (left to right) de izquierda a derecha. Es el valor por defecto. rtl (right to left) de derecha a izquierda.																
lang = "código"	Especifica el idioma del elemento mediante un código predefinido. Los posibles valores de este atributo se encuentran en el documento RFC 1766 , algunos de los valores posibles son: <table><tr><th>Código</th><th>Idioma</th><th>Código</th><th>Idioma</th></tr><tr><td>en</td><td>Inglés (Gran Bretaña)</td><td>es</td><td>Español</td></tr><tr><td>en-US</td><td>Inglés americano</td><td>fr</td><td>Francés</td></tr><tr><td>ja</td><td>Japones</td><td>fr-CA</td><td>Francés de Canadá</td></tr></table>	Código	Idioma	Código	Idioma	en	Inglés (Gran Bretaña)	es	Español	en-US	Inglés americano	fr	Francés	ja	Japones	fr-CA	Francés de Canadá
Código	Idioma	Código	Idioma														
en	Inglés (Gran Bretaña)	es	Español														
en-US	Inglés americano	fr	Francés														
ja	Japones	fr-CA	Francés de Canadá														
xml:lang = "codigo"	Especifica el idioma del elemento mediante un código definido según la recomendación RFC 1766 .																

1.5.- Clasificación de los atributos comunes según su funcionalidad.

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

- **Atributos de eventos y atributos para los elementos que pueden obtener el foco:**
Sólo se utilizan en las páginas web dinámicas creadas con JavaScript. Como no es nuestro objetivo no lo vamos a contemplar.

2.- Elementos HTML.

2.- Elementos HTML.

Introducción

Un elemento HTML está formado por:

- Una **etiqueta de apertura**.
- Cero o más **atributos**.
- **Opcionalmente, un texto**, encerrado por la etiqueta. No todas las etiquetas pueden encerrar texto.
- Una **etiqueta de cierre**. Para algunos elementos no hay etiqueta de cierre o es opcional (ejemplo, input dentro de formularios).

Según el modo en que los elementos ocupan el espacio disponible en la página, los elementos pueden ser de dos tipos: ([referencia w3schools - HTML Block and Inline Elements](#))

- **Elementos en línea**. Sólo ocupan el espacio necesario para mostrar sus contenidos. Su contenido puede ser texto u otros elementos en línea.
- **Elementos de bloque**. 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 allí. Su contenido puede ser texto, elementos en línea u otros elementos de bloque.

2.- Elementos HTML. Introducción

El siguiente ejemplo muestra la diferencia entre ambos comportamientos:

```
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>Ejemplo de la diferencia entre los elementos en línea
6        y los elementos de bloque
7    </title>
8  </head>
9  <body>
10    <h1>Los encabezados son elementos de bloque.</h1>
11    <p>Y los párrafos también.</p>
12    <a href="#">Los enlaces son elementos de línea</a>
13    <p>Incluso si esta definido dentro de un párrafo, <strong>un texto resaltado</strong> sigue siendo un elemento en línea.
14    </p>
15  </body>
16 </html>
```

Los encabezados son elementos de bloque.

Y los párrafos también.

[Los enlaces son elementos de línea](#)

Incluso si esta definido dentro de un párrafo, **un texto resaltado** sigue siendo un elemento en línea.

2.1.- Elementos de la estructura básica de un documento.

Los elementos que definen la estructura básica de un documento son:

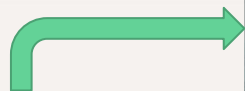
<html> - Define el inicio (y final con "</html>") de un documento HTML. Contiene a los elementos:

<head> - Define el inicio (y final con "</head>") de la cabecera del documento.

<body> - Define el inicio (y final con "</body>") del cuerpo del documento. Permite definir formatos que se aplican a los elementos de la página de manera global, como por ejemplo el color de fondo del texto o la anchura de los márgenes.

Un ejemplo de un documento HTML básico que utiliza estos elementos es:

```
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>Ejemplo de la estructura de un documento HTML</title>
6    </head>
7    <body>
8      Aquí es donde se coloca la información que se quiere visualizar en el navegador.
9    </body>
10 </html>
```



2.2.- Elementos de la sección de cabecera.

Se clasifican en dos tipos:

- **Elementos contenedores:**

Elemento	Descripción
<u>title</u>	Título del documento. Elemento obligatorio.
<u>script</u>	Script incrustado. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.
<u>style</u>	Estilo aplicado al documento utilizando CSS. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.

- **Elementos no contenedores:**

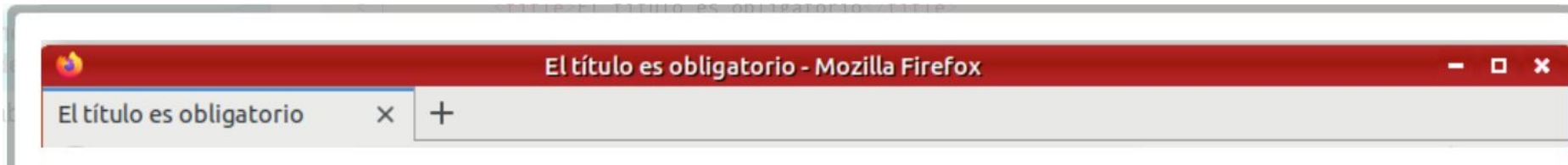
Elemento	Descripción
<u>base</u>	URI base del documento
<u>isindex</u>	Prompt de entrada de datos. (Eliminado de los estándares web aunque todavía algún navegador lo soporta)
<u>link</u>	Enlaces a documentos externos de librerías
<u>meta</u>	Metadatos sobre la página, como la codificación de caracteres, descripción o autores.

2.2.- Elementos de la sección de cabecera.

Ejemplo:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>El título es obligatorio</title>
6      </head>
7      <body>
8          ...
9      </body>
10 </html>
```

El título aparece en la barra del navegador y como nombre de la pestaña. El ejemplo anterior se vería así:



Para que los acentos se vean bien hay que guardar el ejemplo con la codificación indicado en el documento, UTF-8.

2.3.- Encabezados y párrafos.

Para agrupar el texto en párrafos se usa el elemento **<p>**. Es un elemento de bloque.

```
<p>Texto del párrafo </p>
```

Para los encabezados, en HTML se definen 6 elementos:

<h1>,<h2>,<h3>, <h4>, <h5>, <h6>

Cuanto menor es el número, mayor es la importancia del encabezado. El texto marcado debe servir como encabezado a la sección en la que aparece. Se pueden utilizar para organizar jerárquicamente el contenido de la página.

2.3.- Encabezados y párrafos.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Párrafos y encabezados</title>
6   </head>
7   <body>
8     <h1>Equipos</h1>
9     <p>Primer párrafo</p>
10    <p>Segundo párrafo</p>
11    <h2>Recién ascendidos</h2>
12    <p>...</p>
13    <h1>Jugadores</h1>
14    <p>...</p>
15  </body>
16 </html>
```



Equipos

Primer párrafo

Segundo párrafo

Recién ascendidos

...

Jugadores

...

2.3.2.- Comentarios.

Se pueden introducir comentario en los ficheros HTML así:

```
1 | <!-- comentario -->
```

También pueden ser multilínea:

```
1 | <!-- comentario  
2 |   de varias  
3 |   líneas -->
```

Los comentarios no son procesados por los navegadores, sirven para documentar el código.

2.4.- Semántica a nivel de texto.

Algunos elementos útiles son:

- [<a>](#), para definir hipervínculos.
- [](#), para representar que el texto marcado es importante.
- [](#), para indicar énfasis.
- [
](#), introduce un salto de línea.
- [<small>](#), para comentarios accesorios, como lo que suelen aparecer en letra pequeña.

Elemento <a>

Este elemento hace que el texto encerrado entre las etiquetas sea un hipervínculo.

El atributo más importante es **href**, que indica la URL del vínculo, A veces se usa **href="#"** para referirse a la misma página en la que aparece el vínculo.

El atributo **target** permite elegir donde se abrirá el vínculo. Los valores más usados son:

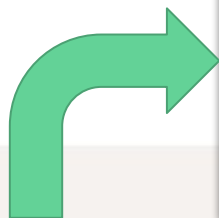
- **target = "_blank"**, para que se abra en una ventana/pestaña nueva.
- **target = "_self"**, para que se abra en la misma. Es el valor por defecto.

2.4.- Semántica a nivel de texto.

Elemento <a>

Ejemplo:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Semántica a nivel de texto</title>
6   </head>
7   <body>
8     Texto marcado como <strong>importante</strong>.
9     <br>
10    Texto con <em>énfasis</em>
11    <br>
12    Texto marcado <small> con el elemento small </small>
13    <br>
14    Pulsa <a href="https://www.w3.org/">aquí</a> para ir a la página del W3C.
15  </body>
16 </html>
```



Texto marcado como **importante**.
Texto con *énfasis*
Texto marcado con el elemento small
Pulsa aquí para ir a la página del W3C.

2.5.- Elementos de listas.

Hay tres tipos de listas: ordenadas, desordenadas y listas de definición.

Elemento	Descripción
<u>ul</u>	Delimita los elementos que forman una lista desordenada
<u>ol</u>	Delimita los elementos que forman una lista ordenada. Tiene varios atributos.
<u>dl</u>	Delimita los elementos que forman una lista de definición
<u>li</u>	Cada uno de los elementos de una lista ordenada o desordenada.
<u>dt</u>	Cada uno de los términos que se definen de una lista de definición.
<u>dd</u>	Cada una de las definiciones de una lista de definición.

2.5.- Elementos de listas.

Un ejemplo de un documento HTML que muestra la forma de utilizar estos elementos es:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Listas</title>
6   </head>
7   <body>
8     <h1>Ejemplo de lista desordenada: Módulos de 1º de ASIR</h1>
9     <ul>
10      <li>Fundamentos de Hardware</li>
11      <li>Gestión de Bases de Datos</li>
12    </ul>
13    <h1>Ejemplos de listas ordenadas: Módulos de 1º de ASIR</h1>
14    Comenzando en 1
15    <ol>
16      <li>Fundamentos de Hardware</li>
17      <li>Gestión de Bases de Datos</li>
18    </ol>
19    Comenzando en 4
20    <ol start = "4">
21      <li>Fundamentos de Hardware</li>
22      <li>Gestión de Bases de Datos</li>
23    </ol>
24    <h1>Ejemplo de lista de definición: Módulos de 1º de ASIR</h1>
25    <dl>
26      <dt>Fundamentos de Hardware</dt>
27      <dd>Componentes físicos de un ordenador</dd>
28      <dt>Gestión de Bases de Datos</dt>
29      <dd>Diseño y uso de bases de datos relacionales</dd>
30    </dl>
31  </body>
32 </html>
```

Ejemplo de lista desordenada: Módulos de 1º de ASIR

- Fundamentos de Hardware
- Gestión de Bases de Datos

Ejemplos de listas ordenadas: Módulos de 1º de ASIR

Comenzando en 1

1. Fundamentos de Hardware
2. Gestión de Bases de Datos

Comenzando en 4

4. Fundamentos de Hardware
5. Gestión de Bases de Datos

Ejemplo de lista de definición: Módulos de 1º de ASIR

Fundamentos de Hardware
Componentes físicos de un ordenador
Gestión de Bases de Datos
Diseño y uso de bases de datos relacionales



2.6.- Elementos de tablas.

Las tablas son una forma habitual de presentar información de manera compacta y fácil de entender para el que la lee. Aunque hay muchas posibilidades, en general las tablas muestran una serie de datos (columnas) comunes para varios elementos (filas). Como ejemplos, podemos pensar en un horario, o en una tabla comparativa de varios productos.

Los elementos para definir una tabla son los siguientes (no es necesario usar todos):

Elemento	Descripción
<u>table</u>	Delimita el contenido de una tabla.
<u>tr</u>	Delimita cada una de las líneas de la tabla.
<u>td</u>	Delimita el contenido de cada celda de la tabla.
<u>colgroup</u>	Permite agrupar columnas.
<u>tbody</u>	Permite agrupar líneas de la tabla.
<u>thead</u>	Define la línea cabecera de la tabla.
<u>th</u>	Delimita cada una de las celdas de la cabecera
<u>tfoot</u>	Define la fila pie de la tabla.
<u>caption</u>	Para añadir una leyenda a la tabla


2.6.- Elementos de tablas.

En HTML las tablas están formadas por filas (elemento <tr>,) y estas a su vez por celdas. Las celdas pueden ser de datos (elemento <td>) o de cabecera (elemento <th>).

- El elemento <table> contiene al resto de elementos.
- Por cada fila de la tabla habrá un elemento <tr>. Este elemento contiene una serie de elementos <td> o <th> (uno por columna).
- Normalmente en la primera fila las celdas son elementos <th> y en el resto <td>.

Ejemplo de tabla básica:

En el navegador se verá así:



Nombre	Apellido	Edad
Juan	Puertas	54
Eva	Montes	44

En [este tutorial](#) puedes ver ejemplos de tablas avanzadas.

NOTA IMPORTANTE: Aunque era habitual hace años, no es correcto usar tablas para la maquetación de la página.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>Tablas</title>
6      </head>
7      <body>
8          <table>
9              <caption>Tabla de socios</caption>
10             <tr>
11                 <th>Nombre</th>
12                 <th>Apellido</th>
13                 <th>Edad</th>
14             </tr>
15             <tr>
16                 <td>Juan</td>
17                 <td>Puertas</td>
18                 <td>54</td>
19             </tr>
20             <tr>
21                 <td>Eva</td>
22                 <td>Montes</td>
23                 <td>44</td>
24             </tr>
25         </table>
26     </body>
27 </html>
```

2.7.- Elementos de formularios.

Los formularios permiten recoger información que el usuario introduce en el navegador.

Es importante validar estos datos, para detectar los errores en local (en el propio equipo), en caso de haberlos. De este modo se evita sobrecargar la red con datos erróneos y sobrecargar al servidor con tareas innecesarias.

Normalmente se combinan los formularios con código **JavaScript**, que ayuda a realizar esas validaciones.

2.7.- Elementos de formularios. Ejemplo (clica en la imagen)

Bienvenido al registro de usuarios de nuestro foro

Introduzca su nombre de usuario y contraseña

Usuario

Contraseña

Repita Contraseña

Describa a continuación sus intereses:

Por último seleccione el sistema operativo de su ordenador:

Linux ▼

Estoy conforme con la política de privacidad de la página ☒

Registrarme

2.7.1.- Declaración de formulario

La apertura y cierre de un formulario se hace mediante el elemento form.

La etiqueta permite especificar una serie de atributos para ajustar sus características. Los principales son:

- **name**— nombre del formulario
- **action**— Acción que se ejecuta cuando se pulsa el botón de enviar formulario
- **enctype**— Formato en el que se enviarán los valores del formulario, depende del contenido se enviarán de una u otra manera.
- **method**— Método de envío de la transmisión de datos, se puede elegir entre dos métodos GET y POST

```
<form name="miFormulario" action="/paginaQueEjecutaLaAccion.php" method="get">
```

2.7.1.- Declaración de formulario

La diferencia entre los métodos de envío GET y POST es la siguiente:

GET	POST
<ul style="list-style-type: none">✓ Permite pasar valores en ASCII con un límite de 100 caracteres✓ Los valores de las variables que se transmiten se pueden ver en la URL y van concatenadas por el símbolo de AND (&) por ejemplo: <code>http://www.pagina.com/index.php?variable1=valor1&variable2=valor2&variable3=valor3</code>	<ul style="list-style-type: none">✓ Permite pasar valores de variables y otros elementos tales como archivos✓ Carece de restricciones de longitud como el método GET✓ Las variables y sus valores no son visibles en la URL

2.7.2.- Campos de formulario - input

Dentro de un formulario puede haber varios tipos de controles: campos de texto normales, campos para contraseñas, fechas o botones de radio, entre otros.

El elemento <input> (que no tiene etiqueta de cierre) se puede usar para varios tipos de control, según el valor que tome el atributo **type**. Por ejemplo, para **type="text"**, que es el valor por defecto, se obtiene un campo de texto. Con **type="radio"**, un botón de radio.

Además, de **type**, los atributos más importantes de la etiqueta son:

- **name** – nombre del campo
- **size** – número de caracteres visibles en el campo. Por defecto 20.
- **maxlength** – número máximo de caracteres que el usuario podrá introducir en el campo
- **value** – valor por defecto del campo de texto
- **placeholder** – valor sugerido, se presenta en color gris y desaparece al hacer foco con el cursor en el campo
- **readonly** – El valor del campo no puede ser modificado por el usuario
- **autofocus** – Sitúa el cursor del ratón en el campo una vez cargada la página
- **required** – No se podrá enviar el formulario hasta que el campo esté cumplimentado

2.7.2.- Campos de formulario - input

Ejemplo:

```
<input type="text" name="usuario" size="30" maxlength="20" placeholder="Escriba aquí el nombre de usuario" required>
```

Se vería en el navegador así:

Usuario

2.7.3.- Campos de formulario - Área de texto

Permite recoger información abierta del usuario pero permitiendo un mayor número de caracteres. Se especifica mediante la etiqueta <textarea>

Para especificar un valor previo en el campo debe escribirse entre las etiquetas <textarea> de apertura y cierre

Atributos principales

- **name** - nombre del campo
- **rows** – número de filas. Reemplazable por la propiedad CSS height
- **cols** – número de columnas. Reemplazable por la propiedad CSS width

También puede utilizar algunos de los atributos especificados para input:

- **placeholder**
- **readonly**
- **autofocus**
- **maxlength**
- **required**

2.7.3.- Campos de formulario - Área de texto

Por ejemplo, la siguiente línea de código **HTML**:

```
Describe a continuación sus intereses: <br> <textarea name="area"></textarea>
```

Quedaría en el navegador del siguiente modo:

Describe a continuación sus intereses:



2.7.4.- Campos de formularios - Lista desplegable

Permite que el usuario seleccione un valor entre diferentes opciones.

Etiquetas

- <select> para definir la lista
- <option> para cada una de las opciones

Atributos - <select>

- **name**– Nombre del elemento
- **size**– Número de elementos de la lista desplegable que se mostrarán
- **multiple**– Permite seleccionar varias opciones manteniendo pulsada la tecla **Ctrl** al seleccionarla

Atributos - <option>

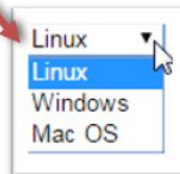
- **selected**– campo que aparece seleccionado por defecto
- **value**– valor que se transmite cuando se envía el formulario

2.7.4.- Campos de formularios - Lista desplegable

A continuación puedes encontrar un ejemplo de código HTML:

```
1 Por último seleccione el sistema operativo de su ordenador:  
2 <select name="Sistema operativo favorito">  
3   <option value="Linux">Linux </option>  
4   <option value="Windows">Windows </option>  
5   <option value="MacOS">Mac OS </option>  
6 </select>
```

```
24 <select name="Sistema operativo favorito">  
25   <option value="Linux">Linux </option>  
26   <option value="Windows">Windows </option>  
27   <option value="MacOS">Mac OS </option>  
28 </select>
```



2.7.5.- Campos de formulario - checkbox

Permite que el usuario indique si está conforme con una o varias opciones.

Etiqueta: `<input type="checkbox">`

Atributos

- **name**– obligatorio
- **checked**– el atributo aparece marcado
- **value**– valor que se transmitirá al enviar el formulario con ese campo seleccionado

Veamos un ejemplo de código HTML:

```
1 | Estoy conforme con la política de privacidad de la página
2 |
3 | <input type="checkbox" name="conforme" checked>
```

Estoy conforme con la política de privacidad de la página ☒

2.7.6.- Botones de formulario

2.7.6.1.- Botón de envío

Permite enviar el formulario a un destinatario, por ejemplo a una página determinada en un servidor web para que procese los datos o una dirección de correo electrónico. Todo dependiendo del valor del atributo **action** de la etiqueta de apertura de formulario.

Etiqueta: `<input type="submit">`

Atributos

- **value**— permite indicar el texto del botón

Display a Submit Button

First name:

Last name:

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

2.7.6.- Botones de formulario

2.7.6.2.- Botón de anulación

Borra el contenido de los campos cumplimentados en el formulario

Etiqueta: <input type="reset">

Atributos

- **value**— permite indicar el texto del botón

Display a Reset Button

Click on the reset button to reset the form.

Enter your email:

Enter a PIN:

2.7.7.- Otros campos de formulario

Dependiendo de los valores que queramos recoger podemos utilizar definiciones de campos más apropiadas como las que se ven en la siguiente tabla.

Campo	Etiqueta	Notas
Oculto	<code><input type="hidden"></code>	
De contraseña	<code><input type="password"></code>	
Para envío de ficheros	<code><input type="file"></code>	Requiere formulario definido con <code>post</code> y <code>enctype="multipart/form-data"</code>
Para recoger correo electrónico	<code><input type="email"></code>	Valida formato automáticamente
Para recoger una URL	<code><input type="url"></code>	Valida formato automáticamente
Para recoger números enteros en un rango	<code><input type="number"></code>	Atributos: <code>max</code> – valor máximo <code>min</code> – valor mínimo <code>step</code> – incremento del contador

2.7.8.- Campos de formulario con formato de fecha

Dependiendo de la fecha que queramos recoger y sus características podemos utilizar diferentes variaciones en las etiquetas.

<input type="datetime-local">

Permite seleccionar día, mes, año y hora

Birthday (date and time):

<input type="date">

Permite seleccionar día, mes y año

Birthday:

<input type="month">

Permite seleccionar mes y año

Birthday (month and year):

2.7.8.- Campos de formulario con formato de fecha

Dependiendo de la fecha que queramos recoger y sus características podemos utilizar diferentes variaciones en las etiquetas.

<input type="week">

Select a week:

Permite seleccionar una semana

<input type="time">

Select a time:

Permite seleccionar una hora

Atributos

- **min**– Menor hora seleccionable
- **max**– Mayor hora seleccionable
- **step**– Incremento del contador del campo en segundos

2.7.9.- Campos de formulario - Rangos

Permite seleccionar un valor dentro de un rango.

Etiqueta: <input type="range">

Atributos

- **min**– valor mínimo del rango
- **max**– valor máximo del rango
- **step**– valor del incremento del contador
- **value**– valor inicial del contador

Display a Range Field

Volume (between 0 and 50):

A horizontal range slider with a dark blue track and a grey handle. The handle is positioned approximately one-third of the way from the left.

Enviar consulta

2.7.10.- Organización de formularios

Etiqueta <label>

Permite asociar a cada campo del formulario una etiqueta con su nombre. El texto mostrado entre las etiquetas **<label>** se muestra y constituye además una ayuda de usabilidad a personas invidentes.

```
1 | <label for="conforme">Acepto el acuerdo de licencia</label>  
2 | <input type="checkbox" name="licencia" id="conforme" value="ok">
```

En el ejemplo la etiqueta **label** se asocia al campo tipo **input** mediante el atributo **for** que contiene el valor del identificador (**id**) de la etiqueta **input**. En el navegador se vería:

Acepto el acuerdo de licencia ☐

2.7.10.- Organización de formularios

Etiqueta <fieldset>

Agrupar los campos de formulario que estén entre la etiqueta de apertura y cierre **<fieldset>** y los rodea con un borde.

```
<h1>The fieldset element</h1>

<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Etiqueta <legend>

Escrita inmediatamente a continuación de la etiqueta de apertura **<fieldset>** agrega un texto relacionado con los campos agrupados

The fieldset element

Personalia:

First name:

Last name:

Email:

Birthday: 

2.7.11.- Un ejemplo de formulario

En este [enlace](https://fpdistancia.educa.madrid.org/pluginfile.php/95215/mod_scorm/content/1/LM!) puedes ver un posible código que genera un formulario como el que aparece en la imagen inferior:

Formulario

← → ↻ 🏠 https://fpdistancia.educa.madrid.org/pluginfile.php/95215/mod_scorm/content/1/LM!

Ejemplo de formulario: Matriculación en el ASIR

Nombre: Apellidos: Sexo: ☐ M ☐ H

Fecha de nacimiento: día mes año

Escoge los módulos en los que te matriculas:

<input type="checkbox"/> Fundamentos de Hardware	<input type="checkbox"/> Servicios de Red e Internet
<input type="checkbox"/> Gestión de Bases de Datos	<input type="checkbox"/> Administración de sistemas Operativos
<input type="checkbox"/> Implantación de Sistemas Operativos	<input type="checkbox"/> Implantación de Aplicaciones Web
<input type="checkbox"/> Planificación y Administración de Redes	<input type="checkbox"/> Administración de Sistemas Gestores de Bases de Datos
<input type="checkbox"/> Lenguajes de Marcas y Sistemas de Gestión de Información	<input type="checkbox"/> Seguridad y Alta Disponibilidad
<input type="checkbox"/> Formación y Orientación Laboral	<input type="checkbox"/> Empresa e Iniciativa Emprendedora
<input type="checkbox"/> Proyecto de Administración de Sistemas Informáticos en Red	<input type="checkbox"/> Formación en Centros de Trabajo

Estudios previos:

Pulsa el botón de "enviar" para formalizar la matricula o el boton de "restablecer" para limpiar el formulario.

2.8.- Multimedia.

Otros elementos habituales son:

Elemento	Descripción
<u>img</u>	Permite insertar una imagen en una página web.
<u>audio</u>	Para insertar audio
<u>video</u>	Para insertar vídeo
<u>object</u>	Para incrustar contenido multimedia (audio, vídeo, ficheros PDF, applets...)

Elemento

Este elemento se usa para insertar una imagen. No tiene etiqueta de cierre. Los atributos más importante son:

- **src.** Especifica la ruta de la imagen. Puede ser local, o una URL cualquiera. Los formatos soportados dependen de cada navegador, pero en general se pueden utilizar JPG, PNG y GIF, entre otros.
- **alt.** Texto alternativo. Se usa si no se puede cargar la imagen o para los lectores de pantalla. Es obligatorio.
- **height.** Altura. Si no se especifica, se escoge el tamaño original. Si se especifica, se escala. Es de los pocos casos en HTML5 en que un atributo tiene información de representación.
- **width.** Anchura de la imagen. Como la anterior.

2.8.- Multimedia.

Elemento

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset='UTF-8'>
5          <title>Imágenes</title>
6      </head>
7      <body>
8          <p> Imagen en internet </p>
9          <img src='https://i.blogs.es/aa1b9a/luna-100mpx/450_1000.jpg' alt='la luna'>
10         <p>Imagen no encontrada</p>
11         <img src='http://rutaincorrecta.jpg' alt='la luna'>
12         <p>Imagen local</p>
13         <img src='index.jpeg' alt='logo'>
14     </body>
15 </html>
```

2.8.- Multimedia.

Elemento

Se vería así: _____

Para la segunda etiqueta ****, como no se encuentra la imagen, el navegador muestra el texto alternativo.

Para poder probar este ejemplo, descarga [esta imagen](#) y "guárdala como" index.jpeg en el mismo directorio que la el fichero html.

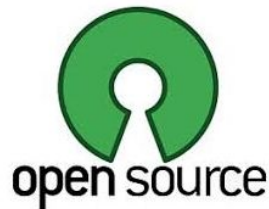
Imagen en internet



Imagen no encontrada

la luna

Imagen local



2.9.- Secciones y etiquetas semánticas.

La siguiente imagen muestra una disposición habitual para estos elementos. Para obtenerla hay que usar CSS.

HTML5 - New Semantic Elements

- `<header>`
- `<nav>`
- `<section>`
- `<article>`
- `<footer>`
- `<aside>`



2.9.- Secciones y etiquetas semánticas.

El elemento **<div>** se usa para agrupar otros elementos en secciones, tanto para organizar el contenido como para posicionarlo mediante hojas de estilo.

En **HTML5** aparecieron varias **etiquetas semánticas** para estructurar el contenido de la página, y por tanto solo se debería usar **<div>** cuando no haya una etiqueta más apropiada

Estas etiquetas son:

- **<header>** Contiene contenido introductorio para la sección de la página en que aparece. Es habitual que contenga los elementos de encabezado, h1,...,h6.
- **<aside>** Se utiliza para contenido parcialmente relacionado con el contenido principal. No tiene por qué mostrarse en un lateral.
- **<footer>** Contiene información sobre la sección correspondiente, como el autor. No tiene que estar necesariamente en la parte de inferior.
- **<section>** Una sección genérica dentro del documento.
- **<article>** Representa un elemento que se puede distribuir de manera independiente o reutilizable.
- **<nav>** Contiene vínculos, internos o externos. Suele usarse para la barra de navegación.

2.10.- Elemento iframe.

El elemento **iframe** permite integrar una página web dentro de otra. En ciertos aspectos, sustituye a la especificación de marcos (frameset) de HTML 4, ya obsoleta.

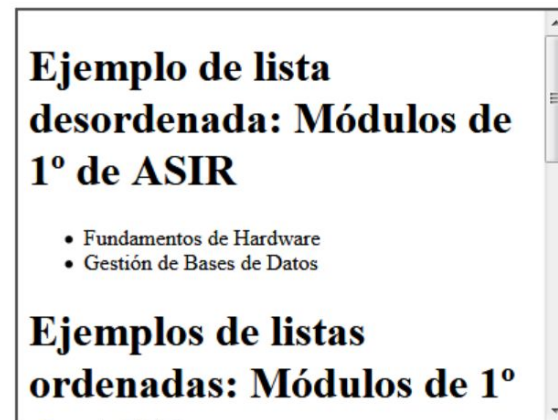
Atributos más importantes de **iframe**:

- **height**: Altura, por defecto en píxeles. Si no cabe todo el contenido, se usa una barra de desplazamiento. También se puede fijar con CSS.
- **width**: Anchura, como el anterior.
- **src**: La ruta al contenido inicial del iframe.
- **name**: Nombre, para referirse al iframe desde un vínculo.

A continuación, vamos a ver un ejemplo de uso habitual.

[Ejemplo de listas](#) [Ejemplo de tablas](#)

Aquí está el iframe



2.10.- Elemento iframe.

El contenido del **iframe** cambia al seguir los vínculos. Para conseguirlo, en los vínculos hay que poner como valor del atributo **target** el valor del atributo name del iframe en que queramos que se cargue.

Es decir, para que se carguen en un **iframe** con **name = "contenido"** se usará:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>iframe</title>
6   </head>
7   <body>
8     <a href = "listas.html" target = "contenido">Ejemplo de listas</a>
9     <a href = "tablas.html" target = "contenido">Ejemplo de tablas</a>
10    <p>Aquí está el iframe</p>
11    <iframe height = "300" width = "400" name = "contenido" src = "listas.html">
12  </body>
13 </html>
```

[Ejemplo de listas](#) [Ejemplo de tablas](#)

Aquí está el iframe



Inicialmente, el **iframe** carga "listas.html". Luego cambia entre "listas.html" y "tablas.html" al seguir los vínculos. Estos ficheros son los que se han utilizado en los ejemplos anteriores. Para probar el ejemplo, tienes que tener los tres ficheros en el mismo directorio y abrir "iframe.html".

2.11.- Validación HTML.

Es posible validar si nuestro código **HTML** cumple con la especificación de **HTML5**, para ello existen diferentes páginas en Internet que nos facilitan este servicio:

- **W3C** - <https://validator.w3.org/>
- **Validator.nu** - <https://html5.validator.nu/>