



# EL LENGUAJE HTML



# ÍNDICE

## EL LENGUAJE HTML

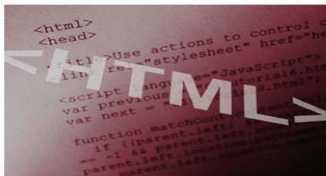
1. Introducción al lenguaje HTML .....	3
2. Las etiquetas de formato .....	9
3. Componentes gráficos HTML .....	18
4. Resumen .....	25



## 1. Introducción al lenguaje HTML

### Objetivos

1. Conocer la sintaxis básica del lenguaje HTML.
2. Conocer las características más importantes de las etiquetas de formato.



### El concepto HTML

Podríamos afirmar, sin lugar a dudas, que gran parte de la popularidad que ha adquirido Internet desde su nacimiento ha sido gracias a HTML.

HTML surgió ante la necesidad de transmitir datos a través de la red, con un formato adecuado que permitiera mostrar dichos datos en un dispositivo de visualización. Más concretamente, HTML consiste en un juego de etiquetas o marcas que permiten aplicar propiedades de formato a los datos.

Cada etiqueta contiene un elemento de apertura (<elemento>) y otro de cierre (</elemento>) entre los que se incluye el dato a formatear:

```
<center>
  Texto centrado
</center>
```

### Página Web y documento HTML

**Al conjunto de datos con sus respectivas etiquetas de formato se le llama**



**Documento HTML**

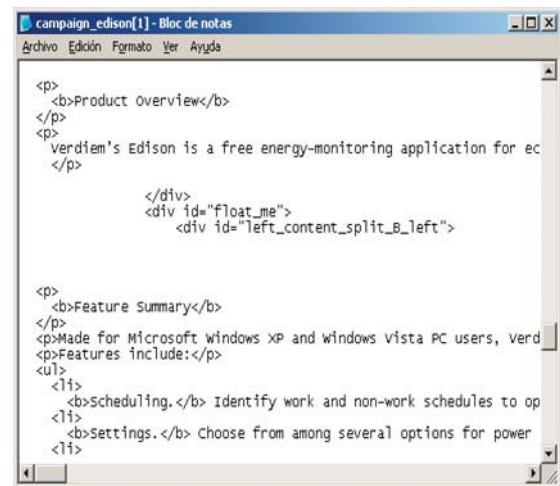
Cuando un documento HTML es recibido por el dispositivo de visualización, conocido como navegador, éste interpreta todas las etiquetas de marcado del documento y muestra en la pantalla del ordenador la información con el formato apropiado.

## EL LENGUAJE HTML

Lo que ve el usuario en su pantalla como resultado de interpretar el documento HTML se conoce como página Web. La siguiente imagen nos ayudará a distinguir ambos conceptos.



Página Web



Documento HTML

### Problemas de compatibilidad de documentos HTML

Como consecuencia del rápido crecimiento que ha experimentado la Web en los últimos 10 años, el lenguaje HTML ha sufrido una especie de sobreutilización. Esto ha provocado que:

- Los distintos fabricantes de navegadores incorporen etiquetas propias al lenguaje a fin de adaptarlo a los nuevos requerimientos de las aplicaciones.
- Lo que implica problemas de compatibilidad de documentos HTML entre plataformas, comportándose un mismo documento de forma diferente según el tipo de navegador.

Para solucionar estos problemas el W3C desarrolló a mediados del año 2000 un nuevo estándar de presentación conocido como XHTML. XHTML está basado en HTML pero utilizando las reglas de formación de XML, lo que permite a los navegadores compatibles con este estándar poder validar los documentos antes de ser procesados, rechazándose todos aquellos que no cumplan exactamente con las reglas definidas por el mismo.

### Diferencia entre documento HTML y su equivalente XHTML

En la siguiente imagen puedes ver la diferencia entre un documento HTML y su equivalente XHTML. Entre otras cosas, se aprecia que en HTML no se hace distinción entre mayúsculas y minúsculas, mientras que en XHTML sí.

Documento HTML	Documento XHTML
<pre>&lt;html &gt; &lt;head&gt;   &lt;TITLE&gt;Ejemplo&lt;/TITLE&gt; &lt;/head&gt; &lt;Body&gt;   &lt;b&gt;una linea&lt;/b&gt;&lt;br&gt;   &lt;b&gt;otra linea&lt;/b&gt;&lt;br&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1- strict.dtd"&gt;  &lt;html xmlns="http://www.w3.org/1999/xhtml"&gt; &lt;head&gt;   &lt;title&gt;Ejemplo&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;b&gt;una linea&lt;/b&gt;&lt;br&gt;   &lt;b&gt;otra linea&lt;/b&gt;&lt;br&gt; &lt;/body&gt; &lt;/html&gt;</pre>

### Sintaxis básica

#### Distinción entre mayúsculas y minúsculas

Antes de entrar a analizar las distintas etiquetas o marcas que forma el estándar XHTML vamos a revisar algunos puntos básicos sobre la sintaxis.

Como se ha indicado anteriormente, el estándar XHTML, al estar basado en XML, distingue entre mayúsculas y minúsculas. Teniendo en cuenta que todas las etiquetas están definidas en minúsculas, el siguiente bloque XHTML dentro de un documento impediría que éste fuera interpretado por el navegador:



Cuando un documento XHTML no cumple con las reglas del estándar se dice que está mal formado. Un documento XHTML mal formado no podrá ser leído por ningún navegador.

```
<p>
    texto incorrecto
<P>
```

### Tipos de etiquetas

Las etiquetas utilizadas por el estándar XHTML pueden ser de dos tipos:

#### Abiertas

Solamente necesitan aparecer una vez para realizar la acción asociada y tienen el formato `<etiqueta/>`. Un ejemplo es la etiqueta de salto de línea `<br/>`. Aquí tenemos una diferencia entre XHTML y HTML, pues en éste último las etiquetas abiertas no incluyen la barra inclinada.

#### Cerradas

Su acción se aplica sobre un bloque de texto que se incluye entre la etiqueta de apertura (`<center>`) y la de cierre (`</center>`). La gran mayoría de las etiquetas XHTML pertenecen a este grupo. En XHTML es obligatorio que todas las etiquetas estén correctamente anidadas, de lo contrario el documento se considerará mal formado.

Algunas etiquetas incluyen atributos, cuya misión es la de proporcionar información adicional sobre la etiqueta. Los atributos se incluyen dentro de la marca de apertura antes del carácter “>” y su valor debe obligatoriamente estar indicado entre comillas:

```
<p align = “center”>
    texto
</p>
```



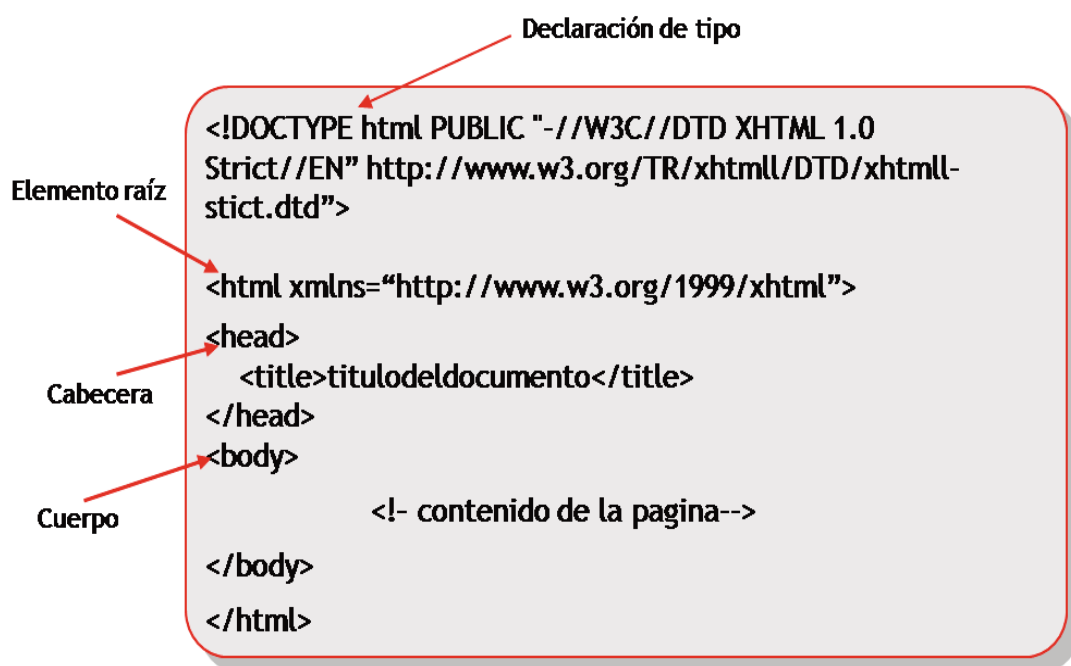
### Comentarios

Los comentarios incluyen información de texto que no debe ser interpretada por el navegador.

El contenido del comentario estará delimitado por los símbolos “<!--” y “-->”, pudiendo ocupar una o varias líneas:

```
<!--ejemplo de
comentario-->
```

Estructura de un documento XHTML. En la imagen podemos ver la estructura de un documento XHTML.



### Componentes básicos

En la sintaxis destacamos cuatro componentes básicos que, obligatoriamente, debe incluir todo documento XHTML:

#### declaración de tipo

Declara la versión del estándar XHTML utilizada por el documento. En este caso vemos que se trata de la versión estricta, existiendo otras como la transicional. El organismo internacional W3C es el encargado de definir el ámbito de aplicación de cada versión.

#### Cabecera

Indicado mediante la etiqueta `<html>`, es el elemento en el que están contenidos el resto de los elementos que componen el documento.

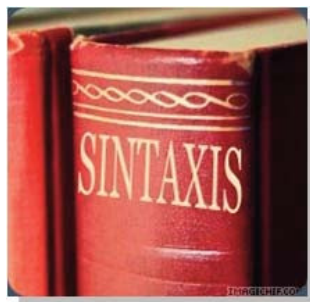
#### Elemento raíz

Contiene información de control sobre el documento, información para uso interno del navegador y que en ningún caso será mostrada en la página.

#### Cuerpo

Todos los datos que se quieren mostrar en la página con sus respectivas etiquetas de formato deben ser incluidos dentro del cuerpo. La etiqueta `<body>`, que es la que delimita el cuerpo de la página dispone de una serie de atributos opcionales que permiten establecer ciertas características generales de la página, como el color o imagen de fondo.

Declara la versión del estándar XHTML utilizada por el documento. En este caso vemos que se trata de la versión estricta, existiendo otras como la transicional. El organismo internacional W3C es el encargado de definir el ámbito de aplicación de cada versión.





## 2. Las etiquetas de formato

### Introducción

Aunque, no es la intención de este capítulo realizar un exhaustivo estudio de todas las etiquetas que forman parte del estándar XHTML, consideramos que todo programador de aplicaciones Web debe tener unos conocimientos básicos sobre este estándar. Es por ello que dedicaremos este capítulo a revisar las características más importantes de las principales etiquetas de formato XHTML.



Sí analizaremos con más profundidad en el siguiente capítulo las etiquetas para la construcción de interfaces gráficas de usuario, puesto que el conocimiento de las mismas resulta de gran importancia para un programador.

Pasemos pues a revisar los distintos tipos de etiquetas de formato que se incluyen en XHTML, todas ellas podrán ser incluidas en cualquier parte del cuerpo (`<body>` `</body>`).

### Estilo de texto

Como su propio nombre indica, estas etiquetas se emplean para aplicar un estilo a determinadas partes del texto. Entre las más importantes están:

#### **`<b>`**

Pone en negrita el texto delimitado por la etiqueta.

#### **`<u>`**

Subraya el texto delimitado por la etiqueta.

#### **`<i>`**

Aplica el estilo cursiva al texto delimitado por la etiqueta.

#### **`<h1>..<h6>`**

Se trata de un conjunto de seis etiquetas que van de la `<h1>` a la `<h6>` y que aplican un formato especial al texto que delimitan. Se les conoce como etiquetas de encabezado, puesto que el formato que aplican es adecuado para definir distintos niveles de título o encabezado dentro de un documento.

### Organización del texto

Mediante estas etiquetas podemos distribuir y organizar el contenido de la página. Entre las más importantes destacamos:

`<br/>`

`<p>`

`<center>`

`<div>`

`<hr/>`

#### `<br/>`

Los saltos de línea “naturales” que introducimos en el texto que forma parte de un documento XHTML no son reconocidos por el navegador. Por tanto, para forzar al navegador a que introduzca un salto de línea en una parte de la página debemos indicárselo mediante la etiqueta abierta `<br/>`.

#### `<p>`

Delimita un párrafo dentro del documento, insertando un salto de línea al final del mismo. Dispone de un atributo, llamado `align`, que define la alineación horizontal del párrafo respecto a la página. Los posibles valores de este atributo son: “left”, “right”, “center” o “justified”.

#### `<center>`

Mediante esta etiqueta centramos horizontalmente el contenido delimitado por la misma. Estos contenidos puede ser cualquier elemento HTML válido (texto, imágenes, párrafos, etc.) dentro del `<body>`.

#### `<div>`

Se trata de una etiqueta con la que delimitamos cualquier bloque de código XHTML válido. El principal objetivo de esta etiqueta es definir distintos bloques en la página a los que se aplicarán estilos diferentes mediante hojas de estilos.

#### `<hr/>`

Se conoce como etiqueta separador y su misión es incluir líneas de separación dentro de una página. Utilizando los atributos adecuados, puede modificarse el ancho, grosor y aspecto de la línea. Dichos atributos son los siguientes:

- `size`.- Define el grosor de la línea en número de píxeles.
- `width`.- Define el ancho de la línea en número de píxeles.
- `align`.- Establece la alineación horizontal de la línea respecto a la página.  
Sus valores posibles son “left”, “right” y “center”.
- `color`.- Define el color de la línea.

Al cerrar este último cuadro de texto, aparecen las imágenes de abajo.



## EL LENGUAJE HTML

La siguiente imagen nos muestra un ejemplo de utilización de las etiquetas anteriores.

Texto XHTML

```
<body>
<h1>Titulo del párrafo</h1>
<p>
    Texto del párrafo principal<br>
    Esto es <b>negrita</b> y esto es <i>cursiva</i>
</p>
<hr size="4"/>
<h2>Subtítulo del apartado</h2>
<p align="center">párrafo centrado</p>
</body>
```

Resultado en navegador



Además de texto, gran parte de las páginas Web existentes incluyen imágenes para hacer más atractiva la presentación de los contenidos.

La etiqueta abierta `<img/>` permite insertar una imagen en un documento y configurar los distintos aspectos de la misma a través de sus atributos.

Entre los principales atributos de esta etiqueta tenemos:

### src

Dirección del archivo que contiene la imagen. Puede ser una dirección relativa o absoluta.

### alt

Texto alternativo que aparecerá en lugar de la imagen si ésta no se localiza.

### height

Alto de la imagen en píxeles.

### width

Ancho de la imagen en píxeles.

### Hipervínculos

Los hipervínculos ofrecen la posibilidad a los usuarios de la página de “navegar” a otras partes de la misma, o incluso a otras páginas ubicadas en cualquier servidor.

La mayor parte de las páginas Web existentes cuentan con este tipo de elementos. Los podemos identificar fácilmente porque, al situar el ratón sobre el texto o imagen que hace de enlace el puntero cambia de forma, transformándose en una mano. En el caso de los textos de enlace, además se muestran con un color diferente al resto del documento.

### Hipervínculos



### Cómo conseguir que un texto o imagen se convierta en hipervínculo

Para conseguir que un texto o imagen se convierta en un hipervínculo, debemos delimitar el elemento correspondiente mediante la etiqueta <a>.

### href

Contiene la referencia al punto de enlace o destino. Puede ser la URL relativa o absoluta de la página destino o, en el caso de un enlace local, el nombre de la marca asociada al destino.

```
<!--Enlace a la página destino.html-->
<a href="destino.html" >Ir destino</a>
<!--Enlace a la zona del documento "fin"-->
<a href="#fin" >Ir a fin</a>
```



### name

Se utiliza para definir una marca dentro del documento a fin de que pueda quedar enlazado con otra parte del mismo. Así pues, el elemento `<a>` debería utilizarse también en la zona del documento donde se vaya a establecer una marca, en este caso empleando el atributo `name`.

```
<a name="fin">Zona final</a>
```

En el caso de que queramos definir una imagen como hipervínculo, el elemento `<img>` correspondiente tendrá que quedar delimitado por la etiqueta `<a>`.

### Tablas

Las tablas son quizá los elementos más utilizados en una página Web, pues además de su propia función de presentación de información en forma tabular, se emplean frecuentemente en la organización de los contenidos de las páginas.



Las etiquetas utilizadas para la generación de tablas son las siguientes:

`<table>`

`<td>`

`<tr>`

`<th>`

### `<table>`

Se trata de la etiqueta principal, pues en su interior se define el contenido de la tabla. La organización de la misma vendrá determinada por otras etiquetas que deben estar incluidas dentro de `<table>`. Antes de analizar estas otras etiquetas, veamos algunos de los atributos más importantes que podemos utilizar con la etiqueta `<table>` y que permiten “jugar” con el formato de la misma:

- **width**. Determina el ancho de la tabla. Su valor puede ser absoluto en píxeles o relativo en porcentaje.
- **bgcolor**. Color de fondo de la tabla.
- **background**. Imagen de fondo de la tabla.
- **border**. Ancho de los bordes y líneas de separación de la tabla. Si no se especifica su valor es 0; la tabla se mostrará sin bordes y sin líneas de separación entre celdas.
- **bordercolor**. Color de las líneas de separación de la tabla.



### <tr>

Se utiliza para delimitar una fila dentro de una tabla, así pues, se deberá incluir una etiqueta <tr> por cada fila a generar.

### <td>

Cada celda dentro de una fila debe estar delimitada por la etiqueta <td>, dentro de la cual se incluirá el bloque de texto o imagen que estará contenida en ella. Una etiqueta <td> puede utilizar además los siguientes atributos:

- **colspan**. Indica el número de columnas de la tabla que debe ocupar la celda.
- **rowspan**. Indica el número de filas que debe ocupar la celda.
- **align**. Alineación horizontal del contenido de la celda. Sus posibles valores son los mismos que se indicaron para el atributo align de <p>.

### <th>

Tiene la misma finalidad que <td>, aplicando además un formato especial (negrita y centrado) al contenido de la celda. Se emplea para aquellas celdas que van a servir de título de una fila o columna de la tabla.

## Listas

Las listas facilitan la organización de la información, permitiendo su presentación como una serie de elementos que pueden venir precedidos por un número o una viñeta.

Las etiquetas utilizadas para la generación de listas son:



### <ol>

Define una lista numerada en donde cada elemento de la misma aparecerá precedido por un número. Dichos números se generan de forma automática y correlativa. Mediante el atributo type podemos definir el tipo de numeración utilizada. Entre sus posibles valores tenemos:

1. Numeración estándar.
- I. Numeración romana.
- A. Numeración con letras mayúsculas.
- a. Numeración con letras minúsculas.

### <ul>

Define una lista no numerada, donde cada elemento de la misma viene precedido por un símbolo o viñeta. Dicho símbolo puede definirse mediante el atributo type, cuyos posibles valores son: “disc”, “circle” y “square”.

### <li>

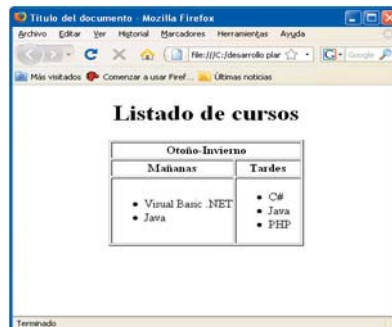
Delimita los elementos de una lista, tanto para listas numeradas como no numeradas.



Ahora veamos un ejemplo de una página HTML donde se hace uso conjuntamente de tablas y listas:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>Titulo del documento</title>
</head>
<body>
  <center>
    <h1>Listado de cursos</h1>
    <table border="1">
      <tr>
        <th colspan="2">Otoño-Invierno</th>
      </tr>
      <tr>
        <th>Mañanas</th>
        <th>Tardes</th>
      </tr>
      <tr>
        <td>
          <ul>
            <li>Visual Basic .NET</li>
            <li>Java</li>
          </ul>
        </td>
        <td>
          <ul>
            <li>C#</li>
            <li>Java</li>
            <li>PHP</li>
          </ul>
        </td>
      </tr>
    </table>
  </center>
</body>
</html>
```

La página resultante al cargar el documento anterior en el navegador se muestra a continuación:



### Marcos

Los marcos permiten dividir la ventana del navegador en zonas. Cada zona o frame puede contener una página diferente.

La utilización de marcos resulta muy útil cuando se quiere tener en un sitio Web una parte del mismo siempre visible, como puede ser el caso de una barra de navegación. Sin embargo, debido a los problemas que presentan en ciertos tipos de navegadores, los marcos están ya en desuso, consiguiéndose un efecto similar pero más estándar y fiable mediante la utilización de hojas de estilo.

Para organizar la ventana del navegador en marcos, debemos crear una página inicial donde se establezca dicha división a través de dos etiquetas especiales.

### frameset

Define una división en filas y columnas. Mediante la utilización de `<frameset>` anidados se crea la división en áreas de la ventana. Esta etiqueta admite los siguientes atributos:

- **cols.** Dimensiones de cada una de las columnas, separadas por una coma (","),. Los valores pueden ser absolutos o porcentuales.
- **rows.** Dimensiones de cada una de las filas, separadas por una coma (","),. Al igual que cols, se pueden especificar valores relativos o absolutos.

### frame

El contenido de cada área definida mediante `<frameset>` se especifica a través de la etiqueta `<frame>`. Esta etiqueta cuenta con los siguientes atributos:

- **src.** Dirección de la página que será cargada en el área del frame.
- **name.** Nombre asignado al frame de cara a cargar el contenido de una página especificada por un enlace.



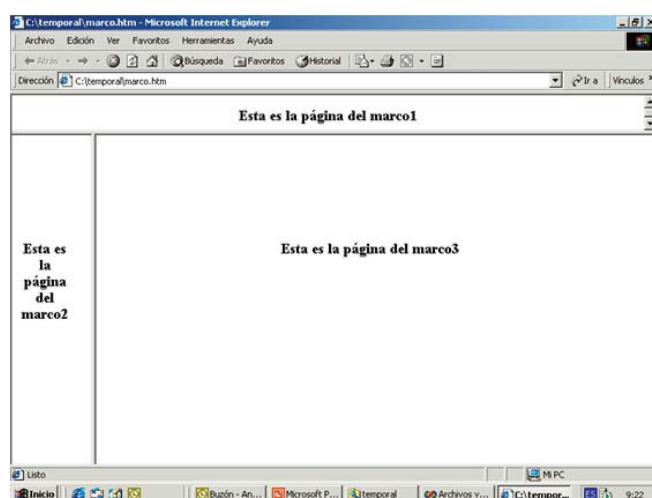


### Ejemplo

La siguiente página de ejemplo crearía una estructura de tres marcos. El primero ocuparía el ancho completo de la página y tendría un alto equivalente al 20% de la misma, mientras que los otros dos tendrían el mismo alto, siendo el ancho de 200 píxeles para el primero de ellos:

```
<html>
<head>
<frameset rows="20%,*">
  <frame src="pagina1.htm"/>
  <frameset cols="200,*">
    <frame src="pagina2.htm"/>
    <frame src="pagina3.htm"/>
  </frameset>
</frameset>
</head>
</html>
```

Aquí se aprecia el aspecto que tendría la ventana del navegador al cargar la página anterior:



Observa dos detalles significativos del código HTML correspondiente a esta página. Por un lado, vemos que la división en zonas de la ventana se define en el <head> de la página de inicio y, por otro, podemos comprobar cómo esta página de inicio no tiene cuerpo.

## 3. Componentes gráficos HTML

### División de etiquetas

El estándar XHTML proporciona también un conjunto de etiquetas para la generación de elementos gráficos que permiten capturar datos de usuario en una página Web y enviarlos a la capa intermedia para su procesamiento.



Estas etiquetas son de gran importancia para el programador Web, puesto que para el correcto procesamiento y tratamiento de los datos es necesario conocer los detalles sobre la forma en que son capturados desde la capa cliente.

Este conjunto de etiquetas podemos dividirlas en dos grandes grupos:

- Formularios
- Controles

### Formularios HTML

Para que la información proporcionada por el usuario a través de los controles gráficos pueda ser enviada a la aplicación que los va a procesar, es necesario que estos controles estén definidos dentro de lo que se conoce como un formulario HTML.

El formulario HTML no ofrece ningún aspecto físico determinado en la página donde se encuentra, tan sólo tiene como misión la de englobar los controles encargados de recoger la información.

Se define mediante la etiqueta `<form></form>`. Pero lo más importante son sus atributos, en especial los siguientes:

#### action

Indica la URL de la aplicación o proceso que recogerá y procesará los datos. Puede ser una página ASP/ASP.NET, un servlet, un script de PHP, etc.

#### method

Indica el método de envío de los datos. Su valor puede ser "GET" o "POST". Si el valor es "GET", los datos serán enviados en la cabecera de la petición como parte de la URL, mientras que si es "POST" la información viajará en el cuerpo de la petición. La forma de envío de los datos puede influir en cómo serán recogidos por el proceso del servidor.

Para que un formulario HTML pueda hacer el envío de los datos, es necesario que cuente con un botón de tipo submit. Al ser pulsado, este botón provoca lo que se conoce como una petición HTTP, incluyendo en ella los datos que se van a enviar a la aplicación del servidor, siguiendo los criterios definidos anteriormente para el método de envío especificado en method.

### Controles HTML

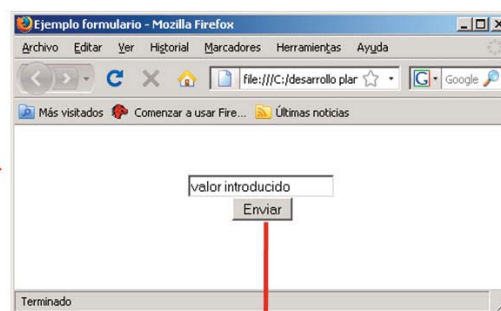
Como hemos indicado, el estándar XHTML incluye etiquetas para la generación de distintos tipos de controles HTML.

Independientemente del tipo de control de que se trate, la etiqueta que lo genere deberá incluir un atributo llamado name que representará el nombre del parámetro enviado en la petición para ese control. En cuanto al valor del parámetro, éste dependerá del tipo de control utilizado.

### Ejemplo

```
<html x:mlns=http://www.w3.org/1999/xhtml>
<head>
  <title>Ejemplo formulario</title>
</head>
<body>
  <center>
    <form action="pagina.asp" method="post">
      <input type="text" name="datoejemplo"/><br>
      <input type="submit" value="Enviar"/>
    </form>
  </center>
</body>
</html>
```

Genera



Envía  
dataejemplo=valor introducido

### Controles de texto

En HTML existen tres controles de texto:

#### 1. Control de texto estándar

Se trata de un control con aspecto de caja de texto que permite al usuario introducir una línea de texto en su interior. Se genera con la etiqueta `<input>`, indicando en su atributo `type` el valor `"text"`.

#### 2. Control de contraseña

Su aspecto es igual al anterior, si bien por cada carácter introducido por el usuario se muestra un símbolo (habitualmente `"*"`) que impide ver el texto escrito en el control. Se genera también con la etiqueta `<input>`, indicando en el atributo `type` el valor `"password"`.

#### 3. Caja de texto multilínea

Es similar a un control de texto estándar, con la particularidad de que permite la introducción de un texto de varias líneas. Se genera con la etiqueta `<textarea></textarea>`.

Por ejemplo, el siguiente campo enviará el parámetro `código=300` utilizando un control `hidden`:

```
<input type = "hidden" name = "codigo" value = "300"/>
```

Con cualquiera de los tres controles descritos anteriormente, el valor que se enviará como parámetro será el texto introducido por el usuario en el control. Si el control se deja en blanco, se incluirá una cadena vacía como valor del parámetro.

Otro tipo de control, es el de campo oculto que representa un texto que se va a enviar en la petición y que no se quiere hacer visible al usuario. Se genera con la etiqueta `<input>`, indicando en su atributo `type` el valor `"hidden"`. El texto que corresponde con el valor del parámetro se incluirá en el atributo `value` del control.

### Botones de radio

Este tipo de control permite al usuario elegir una opción entre un grupo de ellas excluyentes entre sí. Al pulsar sobre uno de los botones del grupo éste pasa a activarse, desactivándose el que estuviera marcado en ese momento.

Los botones de radio se generan con la etiqueta `<input>`, indicando en su atributo `type` el valor “radio”. Para conseguir que todos los botones del grupo sean excluyentes entre sí, es necesario que todos ellos cuenten con el mismo valor del atributo `name`. La siguiente figura muestra un grupo de botones de radio para la selección de una ciudad.

```
<input name="ciudad" type="radio" value="50"/>Valencia <br/>  
<input name="ciudad" type="radio" value="30"/>Logroño <br/>  
<input name="ciudad" type="radio" value="11"/>Cádiz <br/>
```



- ☐ **Valencia**
- ☐ **Logroño**
- ☐ **Cádiz**

Obsérvese como todos los botones de radio cuentan con un atributo `value`. El valor del atributo `value` del botón seleccionado será enviado como parámetro al servidor cuando se produzca el submit del formulario.

Si se seleccionase el botón (Logroño), al realizar el submit del formulario, al servidor se enviaría para este control el parámetro: `ciudad=30`.

Si queremos que un botón aparezca seleccionado de forma predeterminada al cargarse la página en el navegador, deberíamos indicar el atributo `checked` (sin asignarle ningún valor) en el control correspondiente.

Por ejemplo, para preseleccionar Valencia:

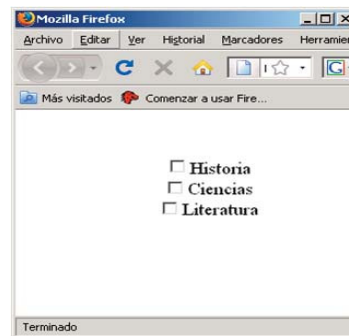
```
<input name = "ciudad" type = "radio"  
value = "50" checked/>
```

### Casillas de verificación

Las casillas de verificación o checkbox son controles que, al igual que los de tipo radio, permiten al usuario elegir opciones dentro de una página aunque, a diferencia de éstos, las opciones presentadas por un grupo de checkbox no son excluyentes entre sí, pudiendo seleccionar más de una.

Para crear un control de tipo checkbox se utiliza la etiqueta abierta `<input>`, indicando en su atributo `type` el valor `"checkbox"`. Dado que las opciones no son excluyentes entre sí, no es necesario que el atributo `name` de cada control del grupo tenga el mismo valor, sin embargo, muchas veces se le asigna a todos los controles del grupo el mismo nombre para facilitar la captura de datos en el servidor.

```
<input type="checkbox" name="tema"
value="historia"/>Historia<br/>
<input type="checkbox" name="tema"
value="ciencias"/>Ciencias<br/>
<input type="checkbox" name="tema"
value="literatura"/>Literatura<br/>
```



En este listado de ejemplo se utilizan tres controles checkbox que tienen como objetivo permitir al usuario seleccionar sus temas de preferencia.

### El atributo checked

Al igual que sucede con los botones de radio, si queremos que un checkbox aparezca seleccionado de forma predeterminada se deberá incluir en el mismo el atributo `checked`. Si un checkbox no se encuentra seleccionado al producirse el submit del formulario, no será enviado ningún parámetro al servidor para ese control.

En el ejemplo anterior, en el que hay varios checkbox con el mismo nombre, si se selecciona más de uno se enviará un parámetro con tantos valores como checkbox estén seleccionados, separándose cada valor del siguiente por una coma (","), Si seleccionásemos los tres temas se enviaría el siguiente parámetro:

```
tema = historia, ciencias, literatura
```



### Listas de selección

El control, listas de selección, presenta una lista de opciones al usuario, entre las que podrá elegir una o varias de ellas.

Para generar este control se utiliza la etiqueta `<select></select>`, donde su atributo `size` (alto del control en número de filas) determinará si se trata de una lista abierta (`size` mayor de 1) o desplegable (`size=1`).



En el interior de `<select>` se incluirá una etiqueta `<option></option>` por cada opción que se quiera mostrar en la lista. Cada elemento `<option>` incluirá en el atributo `value` un valor de texto asociado a la opción y que será enviado al proceso del servidor cuando se produzca el submit del formulario.

### Ejemplo

El siguiente ejemplo crea una lista desplegable con posibles configuraciones de memoria RAM a elegir. Cada configuración de memoria tiene asociado un precio que corresponderá al valor del parámetro enviado al servidor para ese control.

```
<select name = "memoria">
  <option value = "40">1 GB</option>
  <option value = "80">2 GB</option>
  <option value = "150">4 GB</option>
  <option value = "275">8 GB</option>
</select>
```

En la siguiente imagen se muestra el aspecto que tendría dicha lista al ser cargada en el navegador.



### Botones de pulsación

Existen tres tipos de botones de pulsación que se pueden utilizar en un formulario HTML. Los tres se generan mediante la etiqueta `<input>`, aunque cada tipo requiere un valor diferente del atributo `type`.

#### Submit

Este tipo de botón tiene asociado una funcionalidad definida, de modo que al ser pulsado se lanzará una petición del programa del servidor indicado en el atributo `action` del formulario, al que se le enviarán una serie de parámetros con los valores de cada control o grupo de controles del formulario.

#### Reset

También tiene una funcionalidad predefinida, consistente en limpiar todos los campos de texto del formulario.

#### Button

Este botón no tiene ninguna funcionalidad predefinida. Normalmente, se le suele asociar un script de cliente para realizar algún tipo de comprobación de datos cuando se produce la pulsación del control.





### 4. Resumen

Has llegado al final de esta lección de formación que denominamos “El lenguaje HTML”.

En esta lección hemos estudiado los siguientes contenidos:

