

MANUAL



Php

# Aula Mentor



GOBIERNO  
DE ESPAÑA  
MINISTERIO  
DE EDUCACIÓN  
Y CIENCIA

SECRETARÍA GENERAL  
DE EDUCACIÓN  
DIRECCIÓN GENERAL  
DE EDUCACIÓN  
FORMACIÓN PROFESIONAL  
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL  
DE INFORMACIÓN Y  
COMUNICACIÓN EDUCATIVA

[www.mentor.mec.es](http://www.mentor.mec.es)



Aula  
**MENTOR**



## Teoría: El lenguaje HTML

### 1 - Introducción: el lenguaje HTML

El lenguaje **HTML** (HyperText Markup Language) ha sido diseñado para la publicación de contenidos en Internet. El lenguaje dispone de la sintaxis necesaria para formatear textos, incluir imágenes y animaciones, así como para enlazar con otros contenidos del mismo o de otro servidor.

Es un lenguaje interpretado por cada visualizador cliente (navegador), de tal forma que la apariencia de las páginas puede variar entre diferentes navegadores, versiones y plataformas. Por ejemplo, existen algunos que sólo muestran texto, como Lynx. Por esto, siempre debemos tener cuidado, cuando diseñamos nuestras páginas, para que otras personas puedan verlas correctamente.

En este apéndice se muestra la sintaxis compatible de mayor utilización referida al estándar HTML 4.0. Si el alumno o alumna desea obtener mayores conocimientos del tema o los necesita, les recomendamos matricularse en el curso de HTML del proyecto MENTOR. También hemos incluido referencias interesantes a páginas *web* sobre HTML en el apartado **Más Información** de este apéndice.

#### Estructura del Lenguaje HTML

El lenguaje HTML está estructurado mediante etiquetas (*tags*). Estas etiquetas normalmente están emparejadas y encierran un contenido. De esta manera, modifican la cualidad o formato del contenido mostrado en la pantalla del navegador. A continuación, se ofrece un esquema ilustrativo de su estructura:

```
<XXX> Inicio de la etiqueta XXX
  CONTENIDO
</XXX> Final de la etiqueta XXX
```

Las etiquetas pueden ser apiladas ordenadamente. En este caso el esquema quedaría así:

```
<XXX>
  <YYY>

  </YYY>
</XXX>
```

Nota: Algunas etiquetas sólo están integradas por la etiqueta inicial y, por lo tanto, no se cierran. Por ejemplo, la etiqueta **<HR>**, que dibuja una línea horizontal completa, sólo tiene esta etiqueta inicial.

Las letras de las etiquetas pueden ser mayúsculas o minúsculas indistintamente; para mayor claridad, en el curso se escriben siempre en mayúsculas. Asimismo, los propios navegadores,

además de soportar las etiquetas usuales del lenguaje HTML, disponen de etiquetas extras que añaden nuevas funcionalidades a las anteriores, si bien normalmente son incompatibles entre los diferentes navegadores. Por esto, aquí sólo se mostrarán las etiquetas estándares básicas.

Todo fichero HTML empieza con la etiqueta `<HTML>` para indicar el inicio del documento y acaba con la etiqueta `</HTML>` para indicar el final del mismo.

El documento en sí está dividido en 2 partes:

- El **encabezado**, encerrado por las etiquetas: `<HEAD>` `</HEAD>`.
- El **cuerpo**, encerrado por las etiquetas: `<BODY>` `</BODY>`.

En el **encabezado** se encuentra toda la información del documento que no se visualiza en el área principal del navegador, como el título de la página, contenido entre las etiquetas `<TITLE>` `</TITLE>`.

Dentro del cuerpo se halla todo el contenido que se muestra en el área principal del navegador, como texto, imágenes, etc.

Por lo tanto, la estructura básica de una página se escribe de la siguiente manera:

```
<HTML>

  <HEAD>
    <TITLE>Título de la página</TITLE>
  </HEAD>

  <BODY>
    Aquí aparecen todas las etiquetas y contenidos
    que se visualizan en el navegador.
  </BODY>

</HTML>
```

## 2 - Instrucciones básicas

A continuación mostramos una serie de instrucciones básicas y las más comunes. Informamos al lector de que el lenguaje HTML no interpreta los espacios en blanco o los saltos de línea del documento fuente, por lo cual existen caracteres especiales al efecto.

### 2.1. Formateo de caracteres y bloques

<code>&lt;B&gt;Texto&lt;/B&gt;</code>	<i><b>BOLD</b></i> : muestra el contenido "Texto" en negrita.
<code>&lt;I&gt;Texto&lt;/I&gt;</code>	<i>ITALIC</i> : muestra el contenido "Texto" en cursiva.

<pre>&lt;PRE&gt;Texto&lt;/PRE&gt;</pre>	<p>Preformateado: "Texto" aparecerá como si hubiese sido escrito por una máquina de escribir con una fuente de espacio fijo (tipo <i>Courier</i>) respetando espacios y saltos de línea.</p>
<pre>&lt;BLOCKQUOTE&gt; Texto &lt;/BLOCKQUOTE&gt;</pre>	<p>Cita Textual: destaca una cita dentro del texto general añadiendo márgenes izquierdo y derecho.</p>
<pre>&lt;P&gt;</pre>	<p>Separación entre párrafos: efectúa un salto de línea y deja una línea en blanco.</p>
<pre>&lt;BR&gt;</pre>	<p>Salto de línea: efectúa un salto de línea únicamente. Este elemento no se cierra (no existe &lt;/BR&gt;).</p>
<pre>&lt;HR&gt;</pre>	<p>Dibuja una línea horizontal. Este elemento no se cierra (no existe &lt;/HR&gt;).</p>
<pre>&lt;ADDRESS&gt; Nombre &lt;/ADDRESS&gt;</pre>	<p>Autor: indica el nombre del autor del documento formateando el texto en cursiva y alineado a la izquierda.</p>
<pre>&lt;H4&gt; Texto &lt;/H4&gt;</pre>	<p>Formato de texto mediante cabeceras: las etiquetas &lt;Hx&gt;, con x desde 1 hasta 6, muestran diferentes tamaños de letras y un salto de línea variable. La cabecera tipo 1 es la más grande, decreciendo el tamaño de la fuente al aumentar el número x.</p>

## 2.1. Listas:

Es interesante mostrar contenidos en forma de lista. Existen 3 tipos de lista, a saber

<pre>&lt;UL&gt; &lt;LI&gt; Palabra 1 &lt;LI&gt; Palabra 2 &lt;LI&gt; Palabra 3 &lt;LI&gt; Etcétera. &lt;/UL&gt;</pre>	<p><b>Lista no numerada</b> (<i>unordered list</i>): presenta una lista sin que sus elementos estén precedidos de un número secuencial.</p>
<pre>&lt;OL&gt; &lt;LI&gt; Palabra 1 &lt;LI&gt; Palabra 2 &lt;LI&gt; Palabra 3 &lt;LI&gt; Etcétera. &lt;/OL&gt;</pre>	<p><b>Lista ordenada</b> (<i>ordered list</i>): presenta una lista cuyos elementos están precedidos de un número secuencial.</p>
<pre>&lt;DL&gt; &lt;DT&gt; Término 1 &lt;DD&gt; Def. término 1</pre>	<p><b>Lista de definición</b> (<i>definition list</i>): se utiliza para glosarios o definiciones de términos. A diferencia de las 2 anteriores, aquí cada</p>

<pre>&lt;DT&gt; Término 2 &lt;DD&gt; Def. término 2 &lt;/DL&gt;</pre>	<p>bloque de texto está formado por 2 etiquetas. <code>&lt;DT&gt;</code> (<i>Definition Term</i>), que indica la palabra que deseamos definir, y <code>&lt;DD&gt;</code> (<i>Definition Definition</i>), que es la definición propiamente dicha.</p>
---	--

Se pueden anidar unas listas dentro de otras.

Nota: Si deseamos incluir un comentario dentro de la página HTML, utilizaremos los dos símbolos: `<!--` (para abrir el comentario) y `-->` (para cerrarlo).

### 3 - Enlaces y anclas

El uso de enlaces es una de las características principales del lenguaje HTML. Éstos han provocado la rápida difusión del lenguaje, ya que permiten conectar contenidos del mismo o de diferentes servidores de manera muy sencilla e intuitiva. Usualmente, los enlaces tienen la siguiente estructura:

```
<A HREF="xxx"> yyy </A>
```

donde 'xxx' es el destino del enlace e 'yyy' representa el indicador de enlace (un texto y/o una imagen).

Distinguimos 3 usos de los enlaces:

<pre>&lt;A NAME="marca"&gt; &lt;/A&gt; &lt;A HREF="#marca"&gt; &lt;/A&gt;</pre>	<p><b>Enlaces dentro de una misma página</b> (también llamados "anclas") utilizados para poder saltar dentro de una misma página. La primera línea es la marca en la página (también llamada ancla) e indica el sitio exacto adonde se desea saltar; la segunda línea es el enlace, que nos llevará al ancla.</p>
<pre>&lt;A HREF="pagina.htm"&gt; YYY &lt;/A&gt; &lt;A HREF="pagina.htm#marca"&gt;   YYY &lt;/A&gt; &lt;A   HREF="http://dominio/pag.htm" &gt;   YYY &lt;/A&gt;</pre>	<p><b>Enlaces a otras páginas</b> utilizados para poder "enlazar" páginas del mismo o de diferentes servidores. El primer ejemplo es un enlace a otra página del mismo servidor. En el segundo, enlazamos una página del mismo servidor mostrando una determinada posición de la página (ancla). En el último ejemplo enlazamos a una página de otro servidor.</p>

<pre>&lt;A   HREF="mailto:email@dominio"&gt;   YYY &lt;/A&gt;</pre>	<p><b>Enlaces a una dirección e-mail</b> usados para facilitar el envío de correo electrónico.</p>
---	--

## 4 - Incluir imágenes

La forma de incluir imágenes en nuestros documentos HTML es muy parecida a la utilizada para incluir enlaces. En lugar de indicar la página a la que queremos saltar, establecemos el nombre del fichero que contiene la imagen que deseamos mostrar. La estructura de la etiqueta es la siguiente:

```
<IMG SRC="imagen.gif" ALT="descripción"
  ALIGN=alineamiento WIDTH=ancho HEIGHT=alto>
```

A continuación, explicamos la función de los atributos de la etiqueta IMG:

<p><b>SRC</b></p>	<p>Indica el camino y el nombre de fichero donde se encuentra la imagen que se quiere incluir. Ésta puede residir en el mismo o en otro servidor. Además, este atributo es el único que siempre debe aparecer en la etiqueta.</p>
<p><b>ALT</b></p>	<p>Se utiliza para mostrar un texto emergente cada vez que se coloca sobre la imagen el puntero del ratón.</p>
<p><b>ALIGN</b></p>	<p>Se emplea para alinear la imagen en la página HTML. Los valores posibles son: <b>TOP</b>, <b>MIDDLE</b>, <b>BOTTOM</b>, <b>RIGHT</b> y <b>LEFT</b>.</p>
<p><b>WIDTH</b> <b>HEIGHT</b></p>	<p>Se usan para modificar el tamaño de la imagen en la página HTML. El tamaño se expresa en pixels (por ejemplo, <b>WIDTH = 200</b>) o con un porcentaje del tamaño de la imagen (por ejemplo, <b>WIDTH = 60%</b>). Si no se incluyen estos atributos, la imagen aparece con las mismas dimensiones con las que ha sido creada.</p>

Nota: Esta etiqueta no se cierra. Es decir, no existe `</IMG>`.

## 5 - Otras etiquetas de aspecto

En este apartado incluimos una serie de etiquetas que cambian la presentación de la página HTML.

### 5.1 Fondos

Podemos cambiar el fondo de nuestra página de dos maneras diferentes: con un color uniforme o mediante una imagen tápiz. Esto se consigue introduciendo un nuevo atributo en la etiqueta **BODY**, que ya hemos visto anteriormente, de esta manera:

<pre>&lt;BODY   BGCOLOR="#XXYYZZ"   TEXT="#XXYYZZ"   LINK="#XXYYZZ"   VLINK="#XXYYZZ"   ALINK="#XXYYZZ"&gt;</pre>	<p>Mediante el modificador <b>BGCOLOR</b> de la etiqueta <b>BODY</b> elegimos el color de fondo de la página. #XXYYZZ se refiere al color en formato hexadecimal. Es posible que, al cambiar el color por defecto del fondo de la página, sea necesario modificar los colores de texto y de los enlaces para una correcta visualización de los contenidos. Los modificadores son los siguientes:</p> <p><b>TEXT</b> - color del texto.</p> <p><b>LINK</b> - color de los enlaces.</p> <p><b>VLINK</b> - color de los enlaces visitados.</p> <p><b>ALINK</b> - color de los enlaces activos (el que adquieren en el momento de ser pulsados).</p>
<pre>&lt;BODY   BACKGROUND="Archivo"&gt;</pre>	<p>El modificador <b>BACKGROUND</b> de la etiqueta <b>BODY</b> se usa para incluir una imagen de fondo en la página. "Archivo" hace referencia a una imagen en formato gif o jpg que haya en el propio o en otro servidor. Además, se pueden añadir todos los modificadores vistos en el modo anterior para establecer los colores del texto y de los enlaces.</p>

## 5.2 Características de la fuente de un texto

Para establecer las características de la fuente de un texto, usamos la siguiente etiqueta:

```
<FONT FACE="fuente" SIZE="tamaño" COLOR="#XXYYZZ">Texto</FONT>
```

donde los atributos son:

<p><b>FACE</b></p>	<p>Mediante el modificador <b>FACE</b> de la etiqueta <b>FONT</b> elegimos el tipo o tipos de fuente para el "Texto" contenido entre la etiqueta de inicio y de cierre.</p>
<p><b>SIZE</b></p>	<p>Usando este modificador cambiamos el tamaño de la fuente. Se puede asignar un tamaño de fuente absoluto de 1 a 7 (1 es la más pequeña)</p> <p>El tamaño normal o base de la página es el 3. Además, se puede cambiar relativamente respecto a este tamaño normal escribiendo, por ejemplo, <b>&lt;FONT SIZE=+1&gt;</b> para aumentar un punto el tamaño respecto al tamaño base de la página.</p> <p>Incluso se puede cambiar el tipo de fuente normal de todo el</p>

	documento incluyendo al principio de la página HTML (justo a continuación de la etiqueta <code>BODY</code> ) la etiqueta: <code>&lt;BASEFONT SIZE=5&gt;</code> . Así cambiaríamos la fuente por defecto de toda la página a tamaño 5.
<code>COLOR</code>	Mediante el modificador <code>COLOR</code> seleccionamos el color de la fuente. <code>#XXYYZZ</code> se refiere al color en formato hexadecimal.

### 5.3 Alineación de texto e imágenes

Resulta muy sencillo alinear cualquier texto o imagen incluidos en la página HTML utilizando las etiquetas `CENTER`, `LEFT`, `RIGHT`. El siguiente ejemplo alinea el contenido en el centro de la página:

```
<CENTER>Texto o Imagen</CENTER>
```

Además, para alinear imágenes puede usarse el modificador `ALIGN` de la etiqueta `IMG`, como hemos indicado en el apartado anterior.

## 6 - Creación de tablas

Es muy útil usar tablas para la distribución y presentación de contenidos dentro de una página HTML. El esquema básico de una tabla es el siguiente:

```
<TABLE> Inicio de la tabla
  <TR> Primera fila
    <TD> Contenido </TD> Primera columna de la primera fila
    <TD> Contenido </TD> Segunda columna de la primera fila
  </TR> Final de la primera fila

  <TR> Segunda fila
    <TD> Contenido </TD> Primera columna de la segunda fila
    <TD> Contenido </TD> Segunda columna de la segunda fila
  </TR> Final de la segunda fila
</TABLE> Final de la tabla
```

Las etiquetas necesarias para definir una tabla son las siguientes:

<pre>&lt;TABLE BORDER=borde   WIDTH=largo   HEIGHT=ancho   BGCOLOR=#XXYYZZ   CELLPACING=sep_celdas   CELLPADDING=sep_cont&gt;</pre>	<p>Indican el comienzo y final de una tabla.</p> <p>Los atributos que se utilizan para modificar la apariencia de la tabla son:</p> <ul style="list-style-type: none"> <li><code>BORDER</code> establece la anchura del borde de la tabla (<code>BORDER=0</code> indica que la tabla no tiene bordes)</li> </ul>
---	--

<p>Filas</p> <p><code>&lt;/TABLE&gt;</code></p>	<ul style="list-style-type: none"> <li>• <b>WIDTH</b>, <b>HEIGHT</b> fijan el largo y ancho de la tabla respectivamente. Se expresan en pixels o mediante un porcentaje de la dimensión de la pantalla.</li> <li>• <b>BGCOLOR</b> establece el color de fondo de la tabla. Se expresa en formato hexadecimal.</li> <li>• <b>CELLSPACING</b> fija la separación entre las celdas de la tabla. Se expresa en pixels. Por defecto, dicha separación es de 2 pixels.</li> <li>• <b>CELLPADDING</b> determina la separación entre el borde y el contenido dentro de cada celda. Se expresa en pixels. Por defecto, dicha separación es de 1 pixel.</li> </ul>
<p><code>&lt;TR&gt;Celdas&lt;/TR&gt;</code></p>	<p>Señalan el inicio y el final de una fila (<i>table row</i>). Hay que repetirlas tantas veces como filas deseemos obtener.</p>
<p>Contenido</p> <p><code>&lt;TD ALIGN="alineam_horiz" VALIGN="alineam_vert" COLSPAN=extender_horiz ROWSPAN=extender_vert BGCOLOR=#XXYYZZ&gt;</code></p> <p><code>&lt;/TD&gt;</code></p>	<p>Establecen el inicio y el final de una celda que contiene texto, imágenes, enlaces, otra tabla, etc... Hay que repetirlas tantas veces como columnas deseemos obtener.</p> <p>Los atributos que se usan son:</p> <ul style="list-style-type: none"> <li>• <b>ALIGN</b> establece el alineamiento horizontal del contenido en la celda. Los valores posibles son: <b>CENTER</b>, <b>RIGHT</b> y <b>LEFT</b>.</li> <li>• <b>VALIGN</b> establece el alineamiento vertical del contenido en la celda. Los valores posibles son: <b>TOP</b>, <b>BOTTOM</b> y <b>MIDDLE</b>.</li> <li>• <b>COLSPAN</b> indica la extensión de la celda en nº de columnas. Así, una celda puede ocupar varias columnas.</li> <li>• <b>ROWSPAN</b> indica la extensión de la celda en nº de filas. Así, una celda puede ocupar varias filas.</li> <li>• <b>BGCOLOR</b> establece el color de fondo de la celda. Se expresa en formato hexadecimal. Si hemos incluido este atributo en <code>&lt;TABLE&gt;</code>, cambiará el color por defecto de la celda.</li> </ul>
<p><code>&lt;TH&gt; Contenido &lt;/TH&gt;</code></p>	<p>Establece el inicio y el final de una celda de tipo cabecera (<i>header</i>). Se distingue de <code>&lt;TD&gt;</code> en que</p>

muestra el texto en negrita y centrado. Se pueden aplicar los mismos atributos que en la etiqueta `<TD>`.

Se pueden anidar unas tablas dentro de otras.

## 7 - Formularios

Los formularios permiten al usuario introducir información y enviarla al servidor. Existen diferentes tecnologías para implementar esta funcionalidad: *CGI*, *Servlets*, Páginas dinámicas (en *PHP* y *ASP*), *PERL*, etcétera.

La estructura típica de un formulario es la siguiente:

```
<FORM ACTION="acción_URL" METHOD="método" ENCTYPE="tipo_codificado">
```

Cuerpo del formulario con los diferentes elementos para la introducción de datos y los botones de envío o de borrado.

```
</FORM>
```

Los atributos de la etiqueta `FORM` tienen las funciones siguientes:

<code>ACTION</code>	Indica la acción que se debe realizar. Normalmente suele ser una <i>URL</i> que apunta a un <i>CGI</i> , <i>Servlet</i> o página dinámica.
<code>METHOD</code>	Establece el método utilizado para el envío de la información. Los valores posibles son "GET" (por defecto) o "POST". Con el método es "GET" el contenido introducido por el usuario se añade a la <i>URL</i> como si fuera parte de ésta; en cambio, con "POST" la información va en el cuerpo de datos separadamente.
<code>ENCTYPE</code>	Denota el tipo de codificación utilizada para enviar los datos. Por ejemplo, "TEXT/PLAIN" consigue que las respuestas sean recibidas como un fichero de texto, perfectamente legible y sin codificar.

### Elementos del cuerpo de un formulario.

Podemos dividirlos en cinco tipos:

1. Introducción por medio de texto (cajas de texto)
2. Introducción por medio de menús
3. Introducción por medio de botones

4. Botones de envío y de borrado.

5. Elemento invisible.

### 1. Introducción por medio de texto

En este procedimiento el usuario debe completar la información que desea enviar al servidor. Tiene la estructura siguiente:

```
<INPUT TYPE="xxx" NAME="yyy" VALUE="zzz" SIZE="aaa" MAXLENGTH="bbb">
```

donde los atributos de la etiqueta `INPUT` se usan para establecer lo siguiente:

<code>TYPE</code>	Indica el tipo de elemento que utilizamos para la introducción de información. Los valores posibles son: <ul style="list-style-type: none"> <li>● <code>text</code> para establecer el elemento como un 'edit' de texto.</li> <li>● <code>password</code> es igual que el anterior, pero al escribir sólo se muestran asteriscos. Por esto se usa mucho para introducir claves.</li> </ul>
<code>NAME</code>	Establece el nombre del elemento para poder relacionarlo con su contenido.
<code>VALUE</code>	Fija el contenido inicial del elemento.
<code>SIZE</code>	Establece la longitud del elemento en la pantalla.
<code>MAXLENGTH</code>	Establece el número máximo de caracteres que se pueden introducir.

Nota: `INPUT` no tiene etiqueta de cierre.

Cuando es necesario introducir un texto que pueda alcanzar una gran longitud u ocupar varias líneas, como un comentario, es conveniente utilizar un elemento de texto de múltiples líneas.

Esto se consigue con la etiqueta:

```
<TEXTAREA NAME="yyy" ROWS="número" COLS="número">
```

contenido inicial

```
</TEXTAREA>
```

donde los atributos de la etiqueta se usan para lo siguiente:

<code>NAME</code>	Establece el nombre del elemento para poder relacionarlo con
-------------------	--

	su contenido.
ROWS	Representa el número de filas.
COLS	Representa el número de columnas.

## 2. Introducción por medio de menús

Si deseamos que el usuario escoja entre varias opciones en lugar de escribir el texto, haremos uso de los elementos de tipo menú. La estructura más usual es la siguiente:

```
<SELECT NAME="yyy" MULTIPLE SIZE="aaa">

  <OPTION VALUE="bbb" selected>
    literal_opción 1
  </OPTION>

  <OPTION VALUE="ccc">
    literal_opción 2
  </OPTION>

  <OPTION VALUE="ddd" selected>
    literal_opción 3
  </OPTION>

</SELECT>
```

La etiqueta `SELECT` se usa para señalar el inicio y el final del elemento menú. Entre la etiqueta de inicio y de final se encuentran las diferentes opciones que seleccionará el usuario. Sus atributos son éstos:

NAME	Establece el nombre del elemento para poder relacionarlo con su contenido.
MULTIPLE	Permite que se puedan seleccionar múltiples opciones.
SIZE	Fija el número de opciones visibles en la pantalla sin necesidad de desplazarse. Sólo funciona conjuntamente con <code>MULTIPLE</code> . Si es mayor que 1, aparecerá una lista; en caso contrario, se verá una lista desplegable.

La etiqueta `OPTION` se usa para establecer una opción del elemento menú. El literal que visualizará el usuario se encierra entre la etiqueta de inicio y de final. Sus atributos son éstos:

VALUE	Establece el valor que se asigna a la variable.
SELECTED	Si se incluye en alguna de la opciones, esta opción será seleccionada por defecto.

### 3. Introducción por medio de botones

Si deseamos que el usuario confirme una opción determinada con Sí o No, utilizamos el elemento `checkbox`. Tiene el esquema siguiente:

```
<INPUT TYPE="checkbox" NAME="yyy" CHECKED>
```

Sus atributos son éstos:

NAME	Establece el nombre del elemento para poder relacionarlo con el contenido.
CHECKED	Especifica el elemento que aparece seleccionado por defecto.

Además, si queremos que el usuario pueda decidir entre varias posibilidades, utilizaremos el elemento `radiobutton`:

```
<INPUT TYPE="radio" NAME="yyy" VALUE="zzz" CHECKED>
```

Sus atributos son éstos:

NAME	Establece el nombre del elemento para poder relacionarlo con el contenido. Para asociar varios botones de radio a una única variable les pondremos a todos el mismo <code>NAME</code> .
VALUE	Especifica el valor que se asignará a la variable.
CHECKED	Fija la única opción seleccionada por defecto.

### 4. Botones de envío y de borrado

Son elementos esenciales que sirven para que el usuario pulse en ellos y envíe la información al servidor o borre los datos introducidos en la página.

```
<INPUT TYPE="tipo" VALUE="zzz">
```

Sus atributos son éstos:

TYPE	Establece el tipo de botón. Para el envío de datos escribiremos <code>SUBMIT</code>
------	---

	y para borrar los datos de la página "RESET".
VALUE	Especifica el texto que queremos que aparezca en el botón.

## 5. Elemento invisible

A veces, es necesario guardar alguna variable oculta en el formulario para almacenarla en la página activa. Para esto, se utiliza el siguiente elemento invisible:

```
<INPUT TYPE=HIDDEN NAME="yyy" VALUE="zzz">
```

Sus atributos son éstos:

NAME	Establece el nombre del elemento para poder relacionarlo con su contenido.
VALUE	Especifica el valor que se asignará a la variable.

## 8 - Marcos

Un marco (*frame*, en inglés) es una ventana independiente dentro de la ventana general del navegador. Cada marco tiene sus propiedades diferenciadas: bordes y barras de desplazamiento propias. Usando marcos cada página se dividirá, en la práctica, en varias páginas independientes.

Para crear marcos necesitamos un documento HTML específico, que llamamos documento de definición de marcos. En él especificamos el tamaño y la posición de cada marco y el documento HTML que contendrá. La estructura básica de la página principal es la siguiente:

```
<HTML>

<HEAD>
  <TITLE>Mi primera página con marcos</TITLE>
</HEAD>

<FRAMESET COLS="20%,80%">

  <FRAME NAME="indice" SRC="indice.html">
  <FRAME NAME="principal" SRC="inicio.html">
  <NOFRAMES>
    <P>¡Tu navegador no permite marcos!</P>
  </NOFRAMES>

</FRAMESET>

</HTML>
```

Como se puede observar, en este caso no existe la etiqueta `BODY`.

A continuación, explicamos las etiquetas anteriores con detalle:

```
<FRAMESET ROWS="xxx, yyy" COLS="vvv, zzz"> Marcos </FRAMESET>
```

Se usa para definir la distribución de los marcos en la página principal mediante columnas (`COLS`) y filas (`ROWS`). Estas etiquetas son anidables, por lo que podemos incluir unos marcos dentro de otros. Los formatos admitidos para estos atributos son los siguientes:

- Porcentual: como en las tablas, podemos fijar el tamaño de un marco indicando el porcentaje que ocupará del espacio total disponible.
- Absoluto: el marco correspondiente tendrá el tamaño especificado en pixels.
- Sobre el espacio sobrante: escribiendo un asterisco (\*) indicamos que el marco debe ocupar todo el espacio sobrante. Podemos poner este símbolo en varios marcos, que se repartirán el espacio equitativamente. Pero si queremos que uno tenga más espacio, debemos ponerle un número delante del asterisco. Así, un marco con un espacio de 3\* será tres veces más grande que su compañero con sólo 1\*.

En el ejemplo de arriba se crean dos marcos en la ventana principal: una columna, que ocupa el 20% de la página, y otra, que ocupa el 80%.

```
<FRAME>
```

Esta etiqueta define las características de un marco. Los atributos admitidos son los siguientes:

<code>NAME</code>	Asigna el nombre al marco para que podamos referirnos a él.
<code>SRC</code>	Indica la <i>URL</i> del documento HTML que ocupará el marco.
<code>SCROLLING</code>	Establece la aparición de barras de desplazamiento en el marco. Su valor por defecto es <code>AUTO</code> (el navegador decide si se ven o no) Las otras opciones son <code>YES</code> y <code>NO</code> .
<code>NORESIZE</code>	Si lo escribimos, el usuario no podrá cambiar el tamaño del marco.
<code>FRAMEBORDER</code>	Establece el borde del marco. Para eliminar el borde su valor debe ser 0.
<code>MARGINWIDTH</code>	Fija los márgenes horizontales dentro de un marco. Su formato se expresa en pixels.
<code>MARGINHEIGHT</code>	Fija los márgenes verticales dentro de un marco. Su formato se expresa en pixels.

`<NOFRAMES>` Mensaje `</NOFRAMES>`

Estas etiquetas se utilizan para mostrar un mensaje o una página alternativa cuando el navegador del usuario no soporta marcos.

## 9 - Caracteres especiales

Debido a que el lenguaje HTML no admite directamente determinados caracteres por compatibilidad internacional, hay que sustituirlos utilizando la siguiente tabla:

Código	Resultado
á, Á, é, É, etcétera...	á, Á, é, É, í, Í, ó, Ó, ú y Ú
&ntilde	ñ
¿	¿
¡	¡
o	o
a	a
TM	TM
©	©
®	®
	(espacio en blanco, que no puede ser usado para saltar de línea)

## 10 - Más información

A continuación, indicamos algunas direcciones de Internet donde se puede encontrar más información sobre el lenguaje HTML.

<http://www.w3.org/>

Página oficial del World Wide Web Consortium (W3C) donde pueden encontrarse especificaciones, guías, software y aplicaciones relacionadas con las tecnologías de Internet. Incluye los manuales de referencia de HTML. La página está en inglés.

<http://html.programacion.net/>

Página en castellano con cursos y tutoriales sobre HTML que pueden ayudar mucho a los alumnos y alumnas a aprender este lenguaje.

<http://www.webestilo.com/html>

En este sitio se puede aprender el lenguaje HTML de una manera amena y eficiente. Está en castellano.

# Curso de PHP 5 – Apéndice – El lenguaje HTML

MENTOR – CNICE MEC 2006

# Curso PHP 5

## Teoría: Introducción al lenguaje

### 1 - Objetivos

#### Objetivos

- **Adquirir unos conceptos generales y básicos** sobre el lenguaje de programación PHP.
- **Aprender a escribir con PHP programas sencillos** en los que el alumno distinga e interrelacione los elementos que integran el código PHP y los que pertenecen al lenguaje HTML.
- **Conocer inicialmente los materiales del curso** y saber utilizar las herramientas necesarias (Editor dev-PHP, Servidor Apache, dirección de Internet) para realizar los ejercicios y actividades propuestos en las **Actividades**.

### 2 - Características del lenguaje PHP

#### Qué es PHP

PHP es un **lenguaje de programación** que se ejecuta en servidores *web* y permite crear páginas HTML de forma dinámica. Las siglas PHP provienen de *Personal Home Page*, que podemos traducir como **Procesador personal de páginas web** o de **Hipertexto**.

#### Su historia

El lenguaje PHP fue creado a finales de 1994 por **Rasmus Lerdorf**, que inicialmente utilizó las páginas *web* elaboradas con este lenguaje para comprobar quién consultaba su currículum. Por entonces no puso a disposición de los usuarios esta sencilla herramienta.

A principios de 1995 publicó la primera versión disponible para el público, que fue conocida como "Herramientas para paginas *web* personales" (*Personal Home Page Tools*). Esta primera versión estaba integrada por un analizador sintáctico, muy sencillo, que sólo comprobaba algunas macros, así como por una serie de utilidades comunes de las páginas *web* de aquella época, un libro de visitas, un contador y otras rutinas elementales.

A mediados de 1995 el analizador sintáctico de la primera versión fue reelaborado dando origen a la versión 2, que se denominó PHP/FI debido a que en ella se combinaron el analizador sintáctico inicial (PHP) con otro programa, escrito también por el mismo autor, que procesaba datos de formularios (FI). A esta combinación de **herramientas para páginas**

[web personales](#) e [intérprete de formularios](#) le añadió [soporte para mSQL](#). A partir de entonces PHP/FI creció rápidamente y bastantes usuarios empezaron a contribuir en la elaboración del código.

Según datos estadísticos aproximados, se calcula que a finales de 1996 el lenguaje PHP/FI era utilizado al menos en 15.000 páginas *web* de todo el mundo y a mediados de 1997 este número había aumentado a unas 50.000.

A mediados de 1997 el lenguaje PHP, a partir de las aportaciones hechas por muchos usuarios, sufrió un cambio importante al dejar de ser un proyecto personal de Rasmus y convertirse en el proyecto de un grupo mucho más organizado. El analizador sintáctico se reelaboró por completo por Zeev Suraski y Andi Gutmans. [Se pusieron así las bases de la versión 3 de PHP](#) aprovechando la mayoría del código de PHP/FI e incorporando nuevo código, que convirtió la versión 3 en un paquete sólido y potente como herramienta para crear páginas *web* dinámicamente.

A finales de 1999 se distribuyeron tanto PHP/FI como PHP3 en muchos productos comerciales, como el servidor web "C2's StrongHold" y Redhat Linux, hasta el punto de que puede estimarse que ya entonces el lenguaje PHP era usado por más de 1.000.000 de servidores en todo mundo.

[En mayo de 2000 se puso a disposición del público la versión 4 de PHP](#). Esta versión utiliza el potente motor de *scripts* **Zend**, que aporta al intérprete PHP mayor rapidez y le proporciona altas prestaciones. Además, soporta otros servidores *web*, no sólo Apache, que permiten funcionar a PHP como módulo nativo.

[En julio de 2004 se puso a disposición del público la versión 5 de PHP](#), que es la que vamos a utilizar en este curso. Esta versión han mejorado significativamente el rendimiento y las capacidades de PHP utilizando el nuevo motor de *scripts* **Zend 2.0**, que aporta cambios en la Programación orientada a objetos y aumenta las posibilidades del lenguaje en muchos otros aspectos ([ver nota de abajo](#)).

A continuación y a lo largo de todo el curso veremos teórica y prácticamente las ventajas y características de esta última versión de PHP, que es la que vamos a utilizar en el presente curso.

## Sus principales cualidades

1. PHP es un lenguaje **sencillo**. Es decir, si ya se sabe programar en lenguaje C, Perl, Java o en el *Shell* de Unix, [su sintaxis resulta cómoda y bastante intuitiva](#), como se verá en los programas que elaboremos. Ya hemos indicado en la presentación del curso que para hacer este curso es necesario saber programar en algún lenguaje, ya que en el desarrollo del mismo se suponen conocimientos suficientes de programación, pues no se van a explicar más que los contenidos específicos de PHP. En este contexto hay que entender el término "**sencillo**", que hemos usado al referirnos al lenguaje PHP. Su código se lee muy bien, no hay cabeceras ni secciones previas y, por tanto, los programas elaborados con este lenguaje se mantienen fácilmente. [Cualquier alumno que sepa algún lenguaje de programación puede aprender lo fundamental de PHP en poco tiempo](#). Si, además, quiere llegar a realizar operaciones más complejas, como consultar bases de datos, comunicar procesos mediante *sockets*, utilizar FTP o correo electrónico, habrá de tener los conocimientos necesarios sobre estos temas para poder hacerlo con PHP.

Por otra parte, **en el uso de aplicaciones elaboradas con PHP el usuario utiliza un interfaz ya familiar como el navegador web**, el mismo que viene empleando para acceder a cualquier otra página, sin que necesite aprender ninguna combinación especial de teclas, ni instalar otro software adicional diferente de su navegador habitual, ni disponer de un sistema operativo concreto, Unix o Windows.

2. PHP es un lenguaje **rápido**, es decir, a pesar de ser interpretado, la consulta de un usuario a través de Internet al servidor donde está **el programa PHP se ejecuta con bastante velocidad**, de forma que el resultado de una solicitud aparece de forma casi instantánea en la pantalla de su ordenador, convertido en HTML, a no ser que las condiciones del tráfico o de configuración de la red lo impidan.

3. PHP es un lenguaje **multiplataforma**, es decir, su código fuente es interpretado por un servidor, esté instalado éste en un ordenador que **tenga el sistema operativo Unix, Mac o que tenga Windows**. El mismo código es compatible tanto para uno como para otro sistema. Esta característica le da a PHP unas posibilidades muy amplias y le permite ofrecer prestaciones muy variadas.

No obstante, **debemos advertir que PHP nació y se ha desarrollado casi siempre para Linux**, por lo cual su utilización en servidores reales en Internet tiene más sentido y alcanza su eficiencia óptima en este sistema operativo.

**La versión del Curso de PHP archivada en el CD está pensada para funcionar en la plataforma Windows**, ya que la mayoría de las Aulas Mentor y de los alumnos que se matriculan en las mismas dispone exclusivamente de este sistema operativo. No obstante, el código fuente PHP de los ejercicios del curso y el escrito por el alumno en sus actividades es totalmente compatible en todas las plataformas. De ahí su inmediata portabilidad.

Por otra parte, tal como hemos diseñado el curso, **el servidor Apache está instalado en el mismo ordenador local donde trabaja el alumno**. Por ello, solamente el usuario que esté en cada momento realizando las actividades del curso hará peticiones de páginas web al servidor Apache. En consecuencia, éste nunca se verá saturado por la concurrencia de muchos usuarios. Además, para hacer el curso, **el alumno no necesitará estar conectado a Internet** para tener que acceder a un servidor remoto, con el consiguiente ahorro económico y de tiempo.

4. PHP es un lenguaje **libre de licencia, gratuito**. No hay que pagar para poder utilizarlo, **nos deja libres para distribuir sin cargo alguno las aplicaciones que hagamos con él** e incluso podemos mejorar y ampliar sin limitación alguna su código y el de las herramientas que utiliza. Gracias a la colaboración y a la generosa donación de muchos usuarios programadores, PHP mejora y se amplía cada día. Además, gracias a esta colaboración desinteresada está ampliamente probado como herramienta.

5. PHP es un lenguaje que **dispone de numerosas librerías** que facilitan en gran medida la ardua tarea de confeccionar aplicaciones. Debido a esto, es un lenguaje muy apto para ser utilizado tanto por programadores noveles como por desarrolladores profesionales, pues hay librerías adecuadas a desarrolladores de todos los niveles. Además, existen en la red múltiples recursos que hacen más fácil el desarrollo de cualquier aplicación.

6. PHP es un lenguaje que **permite programar utilizando objetos**. Ésta es una importante potencialidad de PHP, dado que la Programación Orientada a Objetos es una tendencia presente prácticamente en todos los lenguajes, por lo cual **PHP puede usar directamente**

estos objetos e incorporarlos dentro de su código. Cualquier programador acostumbrado a utilizar o crear objetos en sus programas podrá servirse de estos componentes en PHP con la misma facilidad que si lo hiciera con otro lenguaje.

7. PHP es un lenguaje que **puede acceder a muchas bases de datos de casi todos los formatos**: Adabas D, dbm, dBase, filePro, Hyperware, Informix, InterBase, LDAP, Microsoft SQL Server, mSQL, MySQL, Oracle, PostgreSQL, Solid, Sybase, etcétera. Esta potencialidad hace de PHP un poderoso lenguaje de Internet, ideal para el acceso, tratamiento y consulta de datos archivados en estos gestores de información.

8. PHP es un lenguaje que **ofrece soporte para poder realizar otras múltiples operaciones**, como acceder a servidores IMAP, envío de correo con SMTP, acceso a servidores de FTP, acceso a SNMP para gestión de redes y equipos, generación dinámica de gráficos y documentos PDF, análisis de documentos XML, corrector ortográfico, generación de datos en WDDX, etcétera.

9. PHP es un lenguaje que **lleva incorporado en su propio diseño el acceso en red desde cualquier puesto de trabajo**. Su código permite controlar perfectamente el acceso, tanto positiva como negativamente, de los usuarios al servidor forzando, si es preciso, su identificación a las opciones que lo requieran.

10. PHP es un lenguaje que **permite proteger de forma eficaz el código fuente de las aplicaciones** informáticas creadas con él, siempre que se tomen las medidas de seguridad adecuadas en el servidor donde se hayan instalado. Como veremos, el código de los programas está sólo en el servidor, que recibe la solicitud de usuario, interpreta el código y devuelve la respuesta en forma de página HTML en el ordenador del cliente.

**Resumiendo, podemos decir que PHP es un lenguaje similar a las ASP (Active Server Pages) de Microsoft, pero más potente, rápido, gratuito, multiplataforma y abierto a mejoras y ampliaciones permanentemente. Debido a esto, PHP se está imponiendo a otros lenguajes de programación como alternativa para el desarrollo de aplicaciones informáticas en Internet.**

## Mejoras añadidas a la versión 5 de PHP

Como ya hemos comentado, la versión 5 de PHP ha supuesto una mejora muy importante respecto a las anteriores versiones. Conserva las características mencionadas en el apartado anterior y, además, añade éstas nuevas:

1. La novedad más importante en PHP5 es la **mejora de la programación orientada a objetos** mediante un nuevo Modelo de Objetos. El manejo de objetos en PHP 5 se ha re-escrito completamente, permitiendo una mejora en rendimiento y muchas características nuevas. En versiones previas de PHP, los objetos eran manejados como tipos primitivos (por ejemplo enteros y cadenas). En el nuevo enfoque, los objetos son referenciados por gestor, y no por valor (puede pensarse en el gestor como un identificador de objeto).

2. **PHP5 incluye un depurador de errores** mejorado de errores que ayuda a seguir los problemas que se van planteando al escribir programas, interpretarlos y compilarlos con un lenguaje tan poco tipificado como PHP. De esta manera se pueden ver sugerencias en tiempo de ejecución sobre la inter-operabilidad del código escrito y compatibilidad hacia adelante, que te ayudarán a mejorar tus scripts poco a poco.

3. **El nuevo intérprete de PHP5 ha incluido nuevas funciones de usuario.** Así, el programador dispone de nuevas funciones de tratamiento de cadenas, matrices, fecha y hora, bases de datos y muchas otras. Algunas de las cuales se verán en la teoría y ejercicios prácticos.

### 3 - Funcionamiento de un programa codificado con PHP

Es importante tener una concepción general de cómo funciona PHP antes de iniciar el curso. **Veamos esquemáticamente los pasos que se dan desde que se inicia un programa en PHP hasta que se ven los resultados de su ejecución en la pantalla del cliente.**

El funcionamiento es bastante sencillo:

- Se escribe el código fuente de la página *web* en lenguaje PHP.
- Se guarda este código en el servidor *web*.
- El navegador de un usuario solicita al servidor la página codificada y guardada en los dos pasos anteriores.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del código PHP al navegador del usuario, que ve en su pantalla la página en formato HTML.

**Observa en el esquema siguiente el proceso explicado en las líneas anteriores:**

