

## Material Imprimible

### Curso HTML

## Módulo 4

### Tratamiento de texto

Las posibilidades que nos ofrece el HTML en cuestión de tratamiento de texto son realmente notables. No se limitan a lo visto hasta ahora, sino que van más lejos todavía. Varios ejemplos de ello son las listas, que sirven para enumerar y definir elementos, los textos preformateados y las cabeceras o títulos. Las listas son utilizadas para citar, enumerar y definir objetos. También son utilizadas corrientemente para desplazar el comienzo de línea hacia la derecha.

Podemos distinguir tres tipos de listas:

- Listas desordenadas
- Listas ordenadas
- Listas de definición

### Las listas desordenadas

Son delimitadas por las etiquetas `<ul>` y `</ul>` (unordered list). Cada uno de los elementos de la lista es citado por medio de una etiqueta `<li>` (sin cierre, aunque no hay inconveniente en colocarlo)

Podemos definir el tipo de viñeta empleada para cada elemento. Para ello debemos especificarlo por medio del atributo `type` incluido dentro de la etiqueta de apertura `<ul>`, si queremos que el estilo sea válido para toda la lista, o dentro de la etiqueta `<li>` si queremos hacerlo específico de un solo elemento

### Listas ordenadas

En este caso usaremos las etiquetas `<ol>` (ordered list) y su cierre. Cada elemento será igualmente precedido de su etiqueta `<li>`.

Del mismo modo que para las listas desordenadas, las listas ordenadas ofrecen la posibilidad de modificar el estilo. En concreto nos es posible especificar el tipo de numeración empleado eligiendo entre números (1, 2, 3...), letras (a, b, c...) y sus mayúsculas

(A, B, C,...) y números romanos en sus versiones mayúsculas (I, II, III,...) y minúsculas (i, ii, iii,...).

### Listas de definición

Cada elemento es presentado junto con su definición. La etiqueta principal es `<dl>` y `</dl>` (definition list). Las etiquetas del elemento y su definición son `<dt>` (definition term) y `<dd>` (definition definition) respectivamente.

**Blockquote** nos permite agregar citas o comentarios ocultos de bibliografía donde recurrimos para obtener la información y así evitar problemas con el derecho de autor. Esta información no estará disponible a simple vista a no ser que accedamos al código interno con la función 12 del teclado

```
<p>Los navegadores suelen aplicar sangría a los elementos blockquote.</p>
```

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

Durante casi 60 años, WWF ha estado protegiendo el futuro de la naturaleza. WWF, la organización de conservación líder en el mundo, trabaja en 100 países y cuenta con el apoyo de más de un millón de miembros en los Estados Unidos y cerca de cinco millones en todo el mundo.

La etiqueta **address** es una etiqueta preformateada que permite generar un estilo propio de dirección.

```
<address>
```

```
Written by John Doe.<br>
```

```
Visit us at:<br>
```

```
Example.com<br>
```

```
Box 564, Disneyland<br>
```

```
USA
```

```
</address>
```

### Enlaces en HTML

El atractivo original del HTML radica en la posible puesta en relación de los contenidos de los archivos introduciendo referencias bajo forma de enlaces que permitan un acceso

rápido a la información deseada. De poco serviría en la red tener páginas aisladas a las que la gente no puede acceder y desde las que la gente no puede saltar a otras.

Un enlace puede ser fácilmente detectado en una página. Basta con deslizar el puntero del ratón sobre las imágenes o el texto y ver cómo cambia de su forma original transformándose por regla general en una mano con un dedo señalador. Adicionalmente, estos enlaces suelen ir, en el caso de los textos, coloreados y subrayados para que el usuario no tenga dificultad en reconocerlos. Si no especificamos lo contrario, estos enlaces texto estarán subrayados y coloreados en azul. En el caso de las imágenes que sirvan de enlace, veremos que están delimitadas por un marco azul por defecto.

Para colocar un enlace, nos serviremos de las etiquetas `<a>` y `</a>`. Dentro de la etiqueta de apertura deberemos especificar asimismo el destino del enlace. Este destino será introducido bajo forma de atributo, el cual lleva por nombre href.

### **Enlaces internos**

Son los enlaces que apuntan a un lugar diferente dentro de la misma página. Este tipo de enlaces son esencialmente utilizados en páginas donde el acceso a los contenidos puede verse dificultado debido al gran tamaño de la misma. Mediante estos enlaces podemos ofrecer al visitante la posibilidad de acceder rápidamente al principio o final de la página o bien a diferentes párrafos o secciones.

### **Enlaces locales**

Un sitio web está constituido de páginas interconexas. Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios. Estos directorios suelen contener diferentes secciones de la página, imágenes, sonidos...Es por ello que en muchos casos no nos valdrá con especificar el nombre del archivo, sino que tendremos que especificar además el directorio en el que nuestro archivo.html está alojado.

### **Enlaces remotos**

Son los enlaces que se dirigen hacia páginas que se encuentran fuera de nuestro sitio web, es decir, cualquier otro documento que no forma parte de nuestro sitio.

Este tipo de enlaces es muy común y no representa ninguna dificultad. Simplemente colocamos en el atributo HREF de nuestra etiqueta `<A>` la URL o dirección de la página con la que queremos enlazar.

### **Enlaces a direcciones de correo**

Los enlaces a direcciones de correo son aquellos que al pincharlos nos abre un nuevo mensaje de correo electrónico dirigido a una dirección de mail determinada. Estos enlaces son muy habituales en las páginas web y resultan la manera más rápida de ofrecer al visitante una vía para el contacto con el propietario de la página.

Para colocar un enlace dirigido hacia una dirección de correo colocamos mailto: en el atributo href del enlace, seguido de la dirección de correo a la que se debe dirigir el enlace.

### **Enlaces con archivos**

Este no es un tipo de enlace propiamente dicho, pero lo señalamos aquí porque son un tipo de enlaces muy habitual y que presenta alguna complicación para el usuario novato. El mecanismo es el mismo que hemos conocido en los enlaces locales y los enlaces remotos, con la única particularidad de que en vez de estar dirigidos hacia una página web está dirigido hacia un archivo de otro tipo.

Si pinchamos un enlace de este tipo nuestro navegador descarga el fichero, haciendo la pregunta típica de "¿Qué queremos hacer con el archivo. Abrirlo o guardarlo en disco"

La etiqueta <footer> define un pie de página para un documento o sección.

Un elemento <footer> debe contener información sobre su elemento contenedor.

Un elemento <footer> generalmente contiene:

- información de autoría
- información registrada
- Información del contacto
- mapa del sitio
- volver al principio enlaces
- documentos relacionados

Puede tener varios elementos <footer> en un documento.

### **Formularios HTML**

¿De qué forma podemos intercambiar información con nuestro visitante? Desde luego, este nuevo aspecto resulta primordial para gran cantidad de acciones que se pueden llevar a cabo mediante el Web: Comprar un artículo, rellenar una encuesta, enviar un comentario al autor...

Hemos visto anteriormente que podíamos, mediante los enlaces, contactar directamente con un correo electrónico. Sin embargo, esta opción puede resultar en algunos casos poco versátil si lo que deseamos es que el navegante nos envíe una información bien precisa. Es por ello que el HTML propone otra solución mucho más amplia: **Los formularios.**

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Son muy utilizados para realizar búsquedas o bien para introducir datos personales por ejemplo en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Usando HTML podemos únicamente enviar el formulario a un correo electrónico.

Si queremos procesar el formulario mediante un programa la cosa puede resultar un poco más compleja ya que tendremos que emplear otros lenguajes más sofisticados. En este caso, la solución más sencilla es utilizar los programas prediseñados que nos proponen un gran número de servidores de alojamiento y que nos permiten almacenar y procesar los datos en forma de archivos u otros formatos. Si sus páginas están alojadas en un servidor que no les propone este tipo de ventajas, siempre pueden recurrir a servidores de terceros que ofrecen este u otro tipo de servicios gratuitos para webs. Por supuesto, existe otra alternativa que es la de aprender lenguajes como ASP o PHP que nos permitirán, entre otras cosas, el tratamiento de formularios.

Los formularios son definidos por medio de las etiquetas `<form>` y `</form>`. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario.

Dentro de esta etiqueta `<form>` debemos especificar algunos atributos:

#### **action**

Define el tipo de acción a llevar a cabo con el formulario. Como ya hemos dicho, existen dos posibilidades:

- El formulario es enviado a una dirección de correo electrónico
- El formulario es enviado a un programa o script que procesa su contenido

En el primer caso, el contenido del formulario es enviado a la dirección de correo electrónico especificada por medio de una sintaxis de este tipo:

```
<form action="mailto:direccion@correo.com" ...>
```

Si lo que queremos es que el formulario sea procesado por un programa, hemos de especificar la dirección del archivo que contiene dicho programa. La etiqueta quedaría en este caso de la siguiente forma:

```
<form action="dirección del archivo" ...>
```

La forma en la que se expresa la localización del archivo que contiene el programa es la misma que la vista para los enlaces.

### **method**

Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son post y get. A efectos prácticos y, salvo que se os diga lo contrario, daremos siempre el valor post.

### **enctype**

Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser "text/plain". Así conseguimos que se envíe el contenido del formulario como texto plano dentro del email.

Si queremos que el formulario se procese automáticamente por un programa, generalmente no utilizaremos este atributo, de modo que tome su valor por defecto, es decir, no incluiremos enctype dentro de la etiqueta <form>

### **Texto corto**

Las cajas de texto son colocadas por medio de la etiqueta <input>. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: type y name.

- **size** Define el tamaño de la caja en número de caracteres. Si al escribir el usuario llega al final de la caja, el texto ira desfilando a medida que se escribe haciendo desaparecer la parte de texto que queda a la izquierda.
- **maxlength** Indica el tamaño máximo del texto que puede ser tomado por el formulario. Es importante no confundirlo con el atributo size. Mientras el primero define el tamaño aparente de la caja de texto, maxlength indica el tamaño máximo real del texto que se puede escribir. Podemos tener una caja de texto con un tamaño aparente (size) que es menor que el tamaño máximo (maxlength). Lo que ocurrirá en este caso es que, al escribir, el texto ira desfilando dentro de la caja

hasta que lleguemos a su tamaño máximo definido por `maxlength`, momento en el cual nos será imposible continuar escribiendo.

- **value** En algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo `value`.

### Texto oculto

Podemos esconder el texto escrito por medio asteriscos de manera a aportar una cierta confidencialidad. Este tipo de campos son análogos a los de texto con una sola diferencia: reemplazamos el atributo `type="text"` por `type="password"`:

### Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<textarea>` y su cierre correspondiente.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc.

Dentro de la etiqueta `textarea` deberemos indicar, como para el caso visto anteriormente, el atributo `name` para asociar el contenido a un nombre que será asemejado a una variable en los programas de proceso. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

**rows** Define el número de líneas del campo de texto.

**cols** Define el número de columnas del campo de texto.

### Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta con su respectivo cierre: `<select>`

Como para los casos ya vistos, dentro de esta etiqueta definiremos su nombre por medio del atributo `name`. Cada opción será incluida en una línea precedida de la etiqueta `<option>`.

**size**

Indica el número de valores mostrados de la lista. El resto pueden ser vistos por medio de la barra lateral de desplazamiento.

**multiple**

Permite la selección de más varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas ctrl o shift. Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

**Botones de radio**

Existe otra alternativa para plantear una elección, en este caso, obligamos al internauta a elegir únicamente una de las opciones que se le proponen. La etiqueta empleada en este caso es <input> en la cual tendremos el atributo type ha de tomar el valor radio.

**Envío, borrado y demás en formularios HTML**

Los formularios han de dar plaza no solamente a la información a tomar del usuario sino también a otra serie de funciones. Concretamente, han de permitirnos su envío mediante un botón. También puede resultar práctico poder proponer un botón de borrado o bien acompañarlo de datos ocultos que puedan ayudarnos en su procesamiento.

**botón de envío**

Para dar por finalizado el proceso de relleno del formulario y hacerlo llegar a su gestor, el navegante ha de validarlo por medio de un botón previsto a tal efecto.

**botón de borrado**

Este botón nos permitirá borrar el formulario por completo en el caso de que el usuario desee rehacerlo desde el principio.

Fuentes: <https://desarrolloweb.com/>