

UNIDAD 2

USO DE LOS LENGUAJES DE MARCAS EN ENTORNOS WEB

HTML

HTML (Hyper Text Markup Language) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

HTML VERSIONES

HTML 2.0. En 1995 se publica el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML, es decir, el HTML 1.0 no existió como estándar. HTML 2.0 no soportaba tablas. Se simplificaba al máximo la estructura del documento para agilizar su edición, donde la declaración explícita de los elementos body, html y head es opcional.

HTML 3.2. La versión HTML 3.2 se publicó en 1997 y es la primera recomendación de HTML publicada por el W3C (Consortio internacional). Esta revisión incorporó los últimos avances de las páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.

HTML 4.0. En diciembre de 1997 se aprobó el estándar HTML 4.0, creado para estandarizar los marcos (frames), las hojas de estilo y los scripts. En septiembre de 2001 se aprobó el estándar HTML 4.01. HTML 5.0. La versión definitiva de la quinta revisión del estándar se publicó en octubre del 2014.

HTML5

HTML5 es la quinta evolución del código HTML por lo que su base de trabajo es la misma.

Estructura de una página

La estructura básica de una página EN HTML5 es:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Titulo de la web</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="estilos.css" />
</head>
<body>
  <header> <p>Mi sitio web creado en html5</p> </header>
  <nav> Navegación</nav>
  <section>
    <article>
      <p> Contenido (ademas de imagenes, citas, videos etc.) </p>
    </article>
  </section>
  <aside>
    <p>contenido</p>
  </aside>
  <footer> Creado por mi 11/ 2018 </footer>
</body>
</html>
```

Ahora veamos cómo funcionan estas etiquetas. La primera etiqueta con la que nos vamos a encontrar para indicar al navegador que se trata de una página HTML 5 es:

`<!DOCTYPE html>` Esta etiqueta debe ser la primera sin espacios ni contenido que le preceda de esta forma el navegador identifica que es HTML 5 y lo que pueda leer lo leerá y lo que no lo ignorará. La siguiente etiqueta es `<html>` esta etiqueta irá cerrada al final del documento y envolverá todo el código como ocurría en las otras versiones HTML. La etiqueta `<html>` puede ir acompañada por el atributo `lang` que identifica el idioma en el que se va a realizar la página web. De esta forma nos podemos encontrar `<html lang="es">` que identificaría el idioma español. Siguiendo con versiones anteriores el siguiente paso es dividir la página en dos secciones cabecera y cuerpo. Para ello utilizaremos las etiquetas `<head>` cabeza y `<body>` cuerpo. En la sección del `<head>` nos encontraremos la parte más descriptiva de la página y que en muchas ocasiones no puede ser visualizada por el usuario como las palabras clave de identificación o indexación de la página por los buscadores o la vinculación con la página css pero también nos encontramos otros elementos como el título, etc. En el `<head>` nos vamos a encontrar etiquetas básicas como `<meta>` que significa descripción y en función del atributo que le acompañe describirá un apartado u otro y no será visualizado por el navegador servirá de ayuda a otros elementos como por ejemplo los motores de búsqueda.

Por ejemplo : `<meta charset="utf-8">` - describe el formato de codificación de caracteres .
`<meta name="description" content="HTML 5">` - descripción de la página `<title> Curso HTML 5 / CSS 3 </title>` - título de la página Las etiquetas `<meta>` no tienen cierre la etiqueta `<title>` sí. En el `<body>` nos encontramos el contenido real de la página que puede visualizar el usuario como el texto, videos, etc. Este contenido se puede dividir en dos partes. Por un lado la estructura de la información y por otra las características como colores, posición, tamaños, etc. Siguiendo el esquema dentro del body debemos utilizar etiquetas de estructura de contenido:



La primera etiqueta debe indicar que el contenido que estamos diseñando es la cabecera. Se trata de la etiqueta `<header> </header>` (cabecera) Debajo nos encontramos la barra de navegación. Para este objeto utilizaremos la etiqueta `<nav></nav>`. En la línea inmediatamente inferior nos encontramos con dos elementos. Por un lado el contenido que vendrá especificado por la etiqueta `<section> <article> </article>... </section>`. Y por otro lado la sección lateral con la etiqueta `<aside> </aside>`. Para finalizar no encontramos el pie cuyas etiquetas son: `<footer> </footer>` Las etiquetas de los diferentes elementos que componen el body se van a ir situando entre las etiquetas de apertura y cierre siguiendo la estructura de la página de arriba abajo y de izquierda a derecha. De ahí que `<section>` se

encuentre por encima de <aside> porque el contenido está a la izquierda y la sección lateral a la derecha en la misma línea. Otra etiqueta interesante es el comentario. Según se van maquetando los elementos de la página en muchas ocasiones necesitamos comentar que tenemos o que hay en esa sección. Las etiquetas de comentario son: <!-- comentario -->

ORGANIZACIÓN DEL TEXTO

A) Párrafos Las primeras etiquetas que organizan el texto por diferentes tamaños son las cabeceras o encabezados. Las cabeceras van desde <h1> el tamaño mayor a <h6> el tamaño más pequeño. Cada vez que cerramos una cabecera se produce un salto doble. Cuando estamos tratando con más de un encabezado de manera continua existe una etiqueta que los agrupa como un grupo de encabezados es <hgroup>. Otra etiqueta que nos sirve para crear párrafos es <p>. Esta etiqueta es la propia de los párrafos y no varía el tamaño de la letra simplemente realiza un salto doble. La etiqueta
 se utiliza para forzar la separación entre líneas del párrafo y genera un salto simple. Esta etiqueta es única y no tiene cierre.

B) Secciones Como indicamos en la unidad anterior las secciones lo que marcan es una zona de contenido que se puede mover o maquetar en diferentes zonas de la página sin que pierda su contexto. En este sentido utilizábamos la etiqueta <section>.

En toda sección además nos podemos encontrar diversos artículos. Los artículos se podrían definir exactamente igual que las secciones zonas de contenido que tiene su propio contexto y que se pueden mover. Pero en este caso ese movimiento se realiza dentro de las secciones e identifica cada parte de contenido dentro de la sección.

La etiqueta <mark> destaca el texto marcándolo como importante y lo resalta con amarillo en el navegador.

La etiqueta <address> indica en cursiva aquellos elementos que son direcciones, nombres, etc. es decir que indican a quien pertenece el contenido o el autor, etc. Por último nos encontramos la etiqueta <time> con su atributo datetime que nos indica que ahí hay una fecha o una hora.

Otras etiquetas

<address> Indica en cursiva aquellos elementos que son direcciones,

 Negrita

<blockquote> Sangría- bloque de texto

<cite> Cita

 Contenedor de texto eliminado (aparecerá tachado)

<ins> Contenedor de texto insertado en un párrafo <ins> (aparecerá subrayado)

<dfn> Representación de definición de un término

<div> Contenedor

 Texto enfatizado

<hr> Línea horizontal de separación

<i> Cursiva

 Imagen

<pre> Texto preformateado

<q> Texto entrecomillado

<script> Contenedor para script

<small> Disminuir el tamaño del texto

 Sección en línea en un documento

 Negrita o texto reforzado

<sub> Texto subíndice

<sup> Texto superíndice

<var> Variable

OTROS ELEMENTOS:

El elemento `<meter>` representa una medida dentro de un rango conocido.

`<p>Uso del disco (GB): <meter max="80" value="45"></meter></p>`

Los elementos `<details>` y `<summary>` nos brindan la posibilidad de mostrar o esconder contenido.

`<details>`

`<summary>Cliqueame para mostrar o esconder</summary>`

`<p>Aquí va el contenido que se muestra o esconde dependiendo de nuestro clic. </p>`

`</details>`

La etiqueta `<progress>` muestra el avance de un proceso.

`<p>Progresión de la tarea:<progress id='barra_progreso' max="100" value="40"></progress></p>`

ESTILOS

A) ESTILOS EN LÍNEA

Los estilos se aplican con el atributo `style` dentro de la propia etiqueta a la que estamos aplicando el estilo. Ejemplo. Para aplicar estilo a `<h1>Título</h1>` Vamos a indicar que "Título" sea con tipografía Tahoma a 24 pt y azul oscuro. Si nos fijamos "Título" está incluido entre dos etiquetas `<h1>` por lo tanto se debe aplicar el atributo `style` a la etiqueta `<h1>` del siguiente modo:.

`<h1 style="font-family:Tahoma,Geneva,Sans-serif;font-size:24pt;color: #006;">Título</h1>`

Estilos más comunes :

| | |
|------------------------|--|
| background-color | color de fondo |
| color | color texto |
| background-image..... | imagen de fondo |
| Font-size | tamaño de la fuente |
| font-family..... | familia fuente: Arial Verdana Courier Times |
| font-weight | ancho letra bold bolder |
| font-style..... | estilo de letra: italic |
| text-align | alineacion del texto: left, center, justify, right |

Todos los estilos tienen su valor separados por dos puntos y entre estilos tenemos punto y coma. Otra variación es cuando un estilo puede tener más de un valor. Por ejemplo en el tipo de fuente podemos indicar más de una fuente por si el usuario no tiene la inicial para la

que está diseñada la web. También podemos ver que el atributo Font-family tiene 3 valores o tipografías: Tahoma, Geneva y San-serif, separadas por comas porque pertenecen a un mismo atributo. Otras características a comentar es las unidades de medida de las tipografías que pueden ser en pt, px, em, etc. También debemos hablar de los valores de los colores que pueden darse en 3 formatos: – Formato texto: el valor es el nombre de un color en inglés como red, navy, etc. – Formato hexadecimal: nace de la relación de 3 colores rojo, verde y azul.

B) HOJA DE ESTILO INTERNA EN EL ARCHIVO HTML

Necesitamos añadir un elemento <style> en el archivo HTML. La hoja de estilo estará en el interior de ese elemento como se muestra a continuación:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Mi primera página con estilo</title>
  <style>
    body { color: purple; background-color: #d8da3d }
  </style>
</head>

<body>
...
```

IMÁGENES ()

Para insertar imágenes en un sitio usaremos la etiqueta . Esta etiqueta debe ir acompañada del atributo src que es el que contiene el URL de la ubicación del archivo de la imagen. El atributo alt se utiliza para dar una breve descripción de la imagen, la cual podremos observar en caso que el navegador del usuario, por alguna razón, no muestre la misma. La etiqueta no tiene cierre. El atributo height define la altura de la imagen y el atributo width define el ancho de la imagen.

Código:

```

```

SONIDO (<audio>)

La etiqueta <audio> permite insertar sonido.

Código:

```
<body>
<audio controls> <source src="ruta/nombre.ogv">
  <source src="ruta/nombre.mp3">
  source src="ruta/nombre.wav">
</audio>
</body>
```

VIDEO(<video>)

Código:

```
<body>
  <video width=".." height=".." controls>
    <source src="ruta.ogv">
  </video>
</body>
```

ENLACES

Probablemente la característica que más ha influido, junto con las imágenes, el espectacular desarrollo de la Web hayan sido los enlaces (links). Un enlace aparece generalmente como un texto azul subrayado y cuando situamos el cursor sobre él se transforma en una mano con el dedo índice extendido. Si pulsamos sobre el enlace saltamos a otra parte del documento, a otro documento situado en cualquier lugar, o incluso se abre el programa de correo para enviar un mensaje a la dirección indicada.

En general los enlaces tienen la siguiente estructura

```
<A HREF="URL"> Texto del enlace</A>
```

El texto del enlace es lo que se visualizará en el navegador. La URL es la dirección donde apunta el enlace y puede ser de estos tipos:

a) Enlace a otro lugar del mismo documento

En este caso la URL se sustituye por un marcador en la misma página. El marcador puede ser texto colocado en el lugar al que queremos saltar. No importa lo que sea ya que no se verá. Veamos un ejemplo para saltar al inicio de esta página. Escribiríamos en el lugar desde el que queremos saltar:

```
<A HREF="#inicio"> Ir al Inicio</A>
```

Y en el sitio exacto adonde queremos saltar, debemos poner la siguiente etiqueta:

```
<A NAME="inicio"> </A>
```

Una vez escrito esto y guardada la página veríamos al abrirla con el Navegador

[Ir al Inicio](#)

Al pulsar sobre ella nos llevaría al principio de la página ya que fue allí donde coloqué el marcador

b) Enlace a otra página local

En este caso la URL se dará de forma relativa (igual que para poner un fondo a la página). Por ejemplo si tenemos dos páginas en el mismo directorio [pagina1.htm](#) y [pagina2.htm](#) y queremos poner un enlace desde la [pagina1.htm](#) a la [pagina2.htm](#) debemos escribir

```
<a href="pagina2.htm">Ir a pagina2</a>
```

Con esto es suficiente, no hay que crear ningún marcador a no ser que quieras ir a un sitio concreto de la [pagina2.htm](#). En ese caso tendrías que poner en el sitio adonde quieres llegar de la [pagina2.htm](#)

```
<a name="sitio"></a>
```

y en la [pagina1.htm](#)

```
<a href="pagina2.htm#sitio">Ir a sitio de la página 2</a>
```

Las palabras **forma relativa**, al principio de esta sección, constituyen un enlace de este último tipo

c) Enlace a una dirección de Internet

Ahora la URL debe darse de forma absoluta

Enlace

te llevaría a la página de entrada del [“Enlace”](#)

LISTAS.

Listas no ordenadas:

Las listas no ordenadas van dentro de la etiqueta y de su cierre . Cada punto que queramos añadir a la lista, lo haremos dentro de la etiqueta y su cierre.

Si no le indicamos nada a la etiqueta , ésta se generará de forma automática. Pero si queremos definir nosotros mismo el símbolo del punto, podemos gracias al atributo “style”.

Vamos con un pequeño ejemplo en código:

```
<ul>
  <li style="list-style-type:circle">Esto es un tipo de punto.</li>
  <li style="list-style-type:square">Este es otro.</li>
  <li style="list-style-type:disc">Y este es otro diferente.</li>
</ul>
```

Cuyo resultado visual será el que veremos a continuación:

o Esto es un tipo de punto.

□ Este es otro.

• Y este es otro diferente.

Listas ordenadas:

Las listas ordenadas van enmarcadas dentro de las etiquetas . Cada punto de la lista se escribe con la misma etiqueta que en las no numeradas: . Pero al ser listas ordenadas los símbolos serán números y éstos se irán generando automáticamente por orden, conforme escribamos nuevos puntos.

En las listas ordenadas podemos hacer que el primer punto comience con el número que nosotros queramos. Lo conseguiremos gracias al atributo “start”. Los siguientes puntos que escribamos se generarán automáticamente por orden, partiendo de ese número.

Por ejemplo, si queremos que nuestra lista empiece por el número 20, sólo deberemos escribir lo siguiente:

```
<ol>
  <li start="20">Este será el número 20. </li>
  <li>Este será el 21. </li>
  <li> Este será el 22. Y así sucesivamente. </li>
</ol>
```

Y el resultado será el siguiente:

20. Este será el número 20.

21. Este será el 21.

22. Este será el 22.

Y así sucesivamente

Listas de definiciones: <dl>, <dt> y <dd>.

Si lo que vamos a hacer es un listado de definiciones, podemos usar las etiquetas <dl>, <dt> y <dd>. Vamos a explicarlas por partes:

La etiqueta <dl> viene de los términos ingleses “Definiton list” y nos indica que dentro de ella, entre ella y su cierre, va a ir una definición.

La etiqueta <dt> viene de los términos “Definition term” y dentro de ella irá el término que vamos a definir. Para entendernos mejor, dentro de <dt> irá el título de la definición.

La etiqueta <dd> viene de los términos “Definition description” y nos dice que dentro de ésta irá la definición.

Si escribimos varios listados de definición, éstas se separarán automáticamente entre ellas para facilitar su diferenciación.

Aquí podemos ver un ejemplo de código de dos listado de definición:

```
<dl> <dt>Aquí va el término que definiremos</dt> <dd>Y aquí dentro irá la definición  
propiamente dicha.</dd> </dl> <dl> <dt>Aquí va la segunda definición</dt> <dd>Y aquí  
dentro irá la segunda definición, separada automáticamente de la anterior.</dd> </dl>
```

Cuyo resultado será el siguiente:

Aquí va el término que definiremos

Y aquí dentro irá la definición propiamente dicha.

Aquí va la segunda definición

Y aquí dentro irá la segunda definición, separada automáticamente de la anterior.

TABLAS

Una tabla no es otra cosa más que un medio de organizar datos en filas y columnas. Este concepto ha estado presente en nuestra sociedad por un largo período de tiempo y ha sido adoptado por HTML en sus etapas iniciales, como una forma de transmitir información que, de otro modo, no sería comprendida tan fácilmente.

Como muchas otras estructuras de HTML, las tablas son construidas utilizando elementos.

En particular, una tabla básica puede ser declarada usando tres elementos, a saber, [table](#) (el contenedor principal), [tr](#) (representando a las filas contenedoras de las celdas) y [td](#) (representando a las celdas). Dejémoslo más claro con un ejemplo:

```
<table>  
  <tr>  
    <td>Celda 1</td>  
    <td>Celda 2</td>  
    <td>Celda 3</td>  
  </tr>  
  <tr>  
    <td>Celda 4</td>  
    <td>Celda 5</td>  
    <td>Celda 6</td>  
  </tr>  
</table>
```

Celda 1 Celda 2 Celda 3

Celda 4 Celda 5 Celda 6

Para poner bordes: style="border: 1px solid black"

Una celda de encabezado es un tipo especial de celda utilizada para organizar y categorizar otras celdas en la tabla. Dicho esto, es difícil imaginar una tabla donde una

celda de encabezado no tenga utilidad. Casi cualquier tabla puede beneficiarse de un grupo de celdas de encabezado bien ubicado.

En el siguiente ejemplo, construiremos una tabla para mostrar información acerca del clima en los próximos días. Aquí, las celdas de encabezado, representadas por el [elemento](#) `th`, son ubicadas en la primera fila de la tabla, encima de las celdas comunes.

```
<table>
  <tr>
    <th>Hoy</th>
    <th>Mañana</th>
    <th>Miércoles</th>
  </tr>
  <tr>
    <td>Soleado</td>
    <td>Mayormente soleado</td>
    <td>Parcialmente nublado</td>
  </tr>
  <tr>
    <td>19°C</td>
    <td>17°C</td>
    <td>12°C</td>
  </tr>
  <tr>
    <td>E 13 km/h</td>
    <td>E 11 km/h</td>
    <td>S 16 km/h</td>
  </tr>
</table>
```

Hoy	Mañana	Miércoles
Soleado	Mayormente soleado	Parcialmente nublado
19°C	17°C	12°C
E 13 km/h	E 11 km/h	S 16 km/h

Unificación de celdas

La unificación de celdas es un mecanismo mediante el cual un autor puede combinar o fusionar dos o más celdas adyacentes ([td](#) y [th](#)) en una tabla. Este efecto puede lograrse utilizando los atributos `colspan`, para unificación horizontal, y `rowspan`, para unificación vertical. Estos atributos pueden tomar un valor entero mayor a cero, que representa el número de celdas que participarán en la unificación.

La idea detrás de este mecanismo se explica por sí misma: cuando el atributo `colspan` es establecido en una celda, indica a los intérpretes que la misma debe abarcar, no solamente su propio espacio, sino también el espacio de un número de celdas contiguas hacia su derecha. Lo mismo sucede para el atributo `rowspan` pero hacia abajo. Debido a la estructura de las tablas en HTML, La unificación sólo puede hacerse hacia la derecha o hacia abajo. Hacerlo hacia arriba o hacia la izquierda consistiría un intento de abarcar espacio ya utilizado.

En el siguiente ejemplo, una celda de la tabla está ocupando el espacio de tres celdas, como lo indica su atributo `colspan`. Nota cómo la declaración de las dos celdas a la derecha de esta celda particular han sido omitidas en el código. Esto es algo muy apropiado, ya que el espacio de estas celdas ha sido tomado por la anterior.

```
<table>
  <tr>
    <th></th>
    <th>Día 1</th>
    <th>Día 2</th>
    <th>Día 3</th>
    <th>Día 4</th>
  </tr>
  <tr>
    <th>Mike (lastimado)</th>
    <td colspan="3">0 km</td>
    <td>4 km</td>
  </tr>
  <tr>
    <th>Susan</th>
    <td>23 km</td>
    <td>18 km</td>
    <td>19 km</td>
    <td>15 km</td>
  </tr>
</table>
```

	Día 1	Día 2	Día 3	Día 4
Mike (lastimado)	0 km			4 km
Susan	23 km	18 km	19 km	15 km

Ahora probaremos el atributo `rowspan`. Aunque la idea de unificación es la misma (excepto por la dirección), la estructura de las tablas en HTML hace que la unificación vertical sea un tanto diferente. En la unificación horizontal todo sucedía en la misma fila (la declaración del atributo `colspan` y la omisión de las celdas contiguas). En la unificación vertical, el atributo `rowspan` se establece en la celda de una fila en particular y la omisión de las celdas unificadas se realiza en las filas subsiguientes, una a una. Esto significa que si una celda en la primera fila abarca el espacio de tres celdas verticalmente, las filas dos y tres tendrán una celda faltante cada una. Veámoslo con un ejemplo.

```
<table>
  <tr>
    <th></th>
    <th>Básico</th>
    <th>Completo</th>
  </tr>
  <tr>
    <th>Instalación</th>
    <td rowspan="2">Gratis!</td>
    <td>$49.99</td>
  </tr>
  <tr>
    <th>Primer año</th>
    <td>$19.99</td>
  </tr>
</table>
```

```

</tr>
<tr>
  <th>Segundo año</th>
  <td>$9.99</td>
  <td>$29.99</td>
</tr>
</table>

```

	Básico	Completo
Instalación	Gratis!	\$49.99
Primer año		\$19.99
Segundo año	\$9.99	\$29.99

FORMULARIOS

Un formulario HTML es una sección de un documento que contiene contenido normal, código, elementos especiales llamados *controles* (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.), antes de enviar el formulario a un agente para que lo procese (p.ej., a un servidor web, a un servidor de correo, etc.)

Estructura de un formulario

La estructura general de un formulario es:

1. Etiqueta de inicio:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

2. Cuerpo del formulario, con los distintos elementos para poder introducir los datos.

3. Botones de envío y de borrado.

4. Etiqueta de cierre **</FORM>**

Etiqueta de inicio

El atributo **ACTION** indica la acción que se debe efectuar y que es que los datos sean **enviados por e-mail** a la dirección indicada. (Si hiciéramos uso del CGI, sería precisamente aquí donde indicaríamos su localización en el servidor, que habitualmente es el directorio cgi-bin, para que procese los datos).

El atributo **METHOD=POST** indica que los datos sean **inmediatamente** enviados por correo a la dirección de e-mail, nada más pulsar el usuario el botón de envío.

Con el atributo **ENCTYPE="TEXT/PLAIN"** se consigue que las respuestas las recibamos como un **fichero de texto, perfectamente legible y sin codificar**.

Elementos para introducir los datos

Los vamos a dividir en tres clases:

1. Introducción por medio de **texto**
2. Introducción por medio de **menús**
3. Introducción por medio de **botones**

La introducción de los datos se consigue por medio de la etiqueta:

<INPUT TYPE="xxx" NAME="yyy" VALUE="zzz">

En donde:

xxx es la palabra que indica el tipo de introducción.

yyy es el nombre que le asignamos nosotros a la variable de introducción del dato.

zzz es la palabra asociada a un elemento.

1. Introducción por medio de **texto** (una línea)

En este caso es **xxx=text**, es decir, **INPUT TYPE="text"**. El atributo **VALUE** no procede en este caso.

Vamos a poner un ejemplo: solicitamos el apellido del usuario.

**<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">**

Escribe tu apellido:

**
<INPUT TYPE="text" NAME="Apellido">
</FORM>**

Que resulta:

Escribe tu apellido:

Si el usuario introduce su apellido, por ejemplo, Ruiz, y pulsa el botón de envío (que veremos más adelante), recibiremos un e-mail suyo con el siguiente texto:

Apellido=Ruiz

La **longitud** de este formulario es por defecto de 20 caracteres. Se puede variar incluyendo en la etiqueta el atributo **SIZE="número"**. Por otra parte, sea cual sea la longitud del formulario, si no se indica nada, el usuario puede introducir el **número de caracteres** que quiera. Se puede limitar esto, incluyendo en la etiqueta el atributo **MAXLENGTH="número"**.

En el caso que hemos visto, si hubiéramos cambiado la etiqueta correspondiente por:

<INPUT TYPE="text" NAME="Apellido" SIZE="10" MAXLENGTH="12">

Habríamos obtenido:

Escribe tu apellido:

(Se puede comprobar cómo no se pueden introducir más de 12 caracteres).

También se puede hacer que el texto introducido **no sea reconocible**, es decir que todos los caracteres se representen por asteriscos. Basta con cambiar en la etiqueta **INPUT TYPE="text"** por **INPUT TYPE="password"**. En el último ejemplo, si cambiamos la etiqueta correspondiente por:

```
<INPUT TYPE="password" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

Obtenemos:

Escribe tu apellido:

(Se puede comprobar cómo los caracteres introducidos se representan por asteriscos)

Introducción por medio de texto (múltiples líneas)

Cuando el texto a introducir puede alcanzar una **gran longitud**, por ejemplo un comentario, es conveniente utilizar un formulario de texto de **múltiples líneas**.

Esto se consigue con la etiqueta de inicio:

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

(en donde no se utiliza **INPUT TYPE** y donde **ROWS** representa el número de filas, y **COLS** el de columnas).

y la de cierre: **</TEXTAREA>**

Ejemplo: un formulario solicitando los comentarios del usuario:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">
```

Introduce tus comentarios:

```
<BR><TEXTAREA NAME="Comentarios" ROWS="6" COLS="40">
</TEXTAREA>
</FORM>
```

Que resulta:

Introduce tus comentarios:

(El salto de línea del texto introducido no se efectúa automáticamente).

Una vez que el usuario haya escrito sus comentarios dentro del formulario, y haya pulsado el botón de envío, recibiremos un e-mail suyo con el siguiente texto:

Comentarios=Bla bla bla...

Nuevos tipos de input

El nuevo tipo `<input type="email">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una dirección de email *válida*, pero no comprueba si la dirección existe o no, sólo si el formato es válido.

El nuevo tipo `<input type="url">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una URL correcta.

El nuevo tipo `<input type="date">` es de los más esperados y útiles. En muchos de los sitios web es normal disponer de campos específicos de fecha, donde el usuario debe especificar fechas (para un concierto, vuelo, reserva de hotel, etc). Al existir tantos formatos de fecha diferentes (DD-MM-YYYY o MM-DD-YYYY o YYYY-MM-DD), esto puede suponer un inconveniente para los desarrolladores o los propios usuarios.

Este nuevo tipo de campo resuelve estos problemas, ya que es el navegador el que proporciona la interfaz de usuario para el calendario.

El nuevo tipo `<input type="time">` permite introducir una hora en formato 24h, y validarlo. De nuevo, es el navegador el encargado de mostrar la interfaz de usuario .

Tipe week

El nuevo tipo `<input type="week">` permite la selección de una semana del año concreta. La representación interna del mes es un valor entre 1 y 53, pero de nuevo queda en manos del navegador la manera de mostrarlo al usuario. La representación interna de la semana 7, por ejemplo, es la siguiente: 2013-W07.

Tipo number

Como es de esperar, el nuevo tipo `<input type="number">` valida la entrada de un tipo de dato numérico. Este tipo de campo encaja perfectamente con los atributos min y max, que veremos más adelante.

tipo color

El nuevo tipo `<input type="color">`, permite seleccionar un color de una paleta de colores.

Nuevos atributos

Al igual que los nuevos tipos de campo, el elemento `input` ha recibido nuevos atributos para definir su comportamiento y restricciones: `autocomplete`, `min`, `max`, `multiple`, `pattern`, `autofocus`, `placeholder`, `required` y `step`. Existe además un nuevo atributo, `list`, que hace referencia a otro elemento, permitiendo crear un nuevo tipo de entrada de datos.

El atributo *booleano* `autofocus` permite definir que control va a tener el foco cuando la página se haya cargado. Hasta ahora, esto se conseguía a través de JavaScript, utilizando el método `.focus()` en un elemento concreto, al cargarse el documento. Ahora es el navegador el encargado de esta tarea, y puede comportarse de manera más inteligente, como no cambiando el foco de un elemento si el usuario ya se encuentra escribiendo en otro campo (éste era un problema común con JavaScript).

Únicamente debe existir un elemento con este atributo definido en el documento. Desde el punto de vista de la usabilidad, hay que utilizar este atributo con cuidado. Hay que utilizarlo únicamente cuando el control que recibe el foco es el elemento principal de la página, como en un buscador, por ejemplo.

Atributo placeholder

Una pequeña mejora en la usabilidad de los formularios, suele ser colocar un pequeño texto de ayuda en algunos campos, de manera *discreta* y que desaparece cuando el usuario introduce algún dato. Como con el resto de elementos, hasta ahora era necesario utilizar JavaScript para realizar esta tarea, pero el atributo `placeholder` resuelve esta tarea.

Atributo required

Este atributo puede ser utilizado en un `<textarea>` y en la gran mayoría de los elementos `<input>` (excepto en los de tipo `hidden`, `image` o botones como `submit`). Cuando este atributo está presente, el navegador no permite el envío del formulario si el campo en concreto está vacío.

Atributos min y max

Como hemos visto en el campo `<input type="number">`, estos atributos restringen los valores que pueden ser introducidos; no es posible enviar el formulario con un valor menor que `min` o un valor mayor que `max`. También es posible utilizarlo en otro tipo de campos como `date`, para especificar fechas mínimas o máximas. `<input type="date" min="2010-01-01" max="2010-12-31">`

2. Introducción por medio de menús

Si queremos que el usuario, en vez de introducir un texto, como hemos visto en los casos anteriores, **escoja entre varias opciones** que le presentamos nosotros, haremos uso de un formulario en forma de menú.

Se consigue con la etiqueta de inicio **<SELECT NAME="yyy">** y la de cierre **</SELECT>**. Las distintas opciones a escoger se consiguen con la etiqueta **<OPTION>**.

Ejemplo: Pedimos al usuario que elija su color preferido:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">
¿Cuál es tu color preferido?
<BR><SELECT NAME="ColorPreferido">
<OPTION>Rojo
<OPTION>Verde
<OPTION>Azul
<OPTION>Amarillo
</SELECT >
</FORM>
```

Que resulta:

¿Cuál es tu color preferido?

Si el usuario ha escogido, por ejemplo, Azul y ha pulsado el botón de envío, recibiremos un e-mail suyo con el texto: ColorPreferido=Azul.

En el ejemplo anterior, sólo es visible en el formulario una opción. Si queremos que sean **visibles múltiples opciones a la vez**, añadimos en la etiqueta los atributos **MULTIPLE SIZE="número"**, donde especificamos el número de opciones visibles.

Si cambiamos en el ejemplo anterior la etiqueta correspondiente por:

```
<SELECT NAME="ColorPreferido" MULTIPLE SIZE="2">
```

Obtenemos:

¿Cuál es tu color preferido?

Formulario de confirmación (checkbox)

Si queremos que el usuario **confirme** una opción determinada, podemos hacer uso de un formulario de confirmación, o checkbox, que se consigue con la etiqueta:

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

Ejemplo: Solicitamos al usuario que confirme su inclusión en una lista de correo:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">
<INPUT TYPE="checkbox" NAME="Lista">
```


Sí, deseo ser incluido en la lista de correo.

</FORM>

Que resulta:

☐ Sí, deseo ser incluido en la lista de correo.

Si el usuario marca este formulario y pulsa el botón de envío, recibiremos un e-mail suyo con el texto: Lista=On.

Si queremos que el formulario aparezca **inicialmente como marcado** (el usuario no necesitará hacerlo), basta con añadir el atributo **CHECKED** dentro de la etiqueta. En el ejemplo anterior si sustituimos la etiqueta equivalente por:

<INPUT TYPE="checkbox" NAME="Lista" CHECKED>

Obtenemos:

☐ Sí, deseo ser incluido en la lista de correo.

Botones de radio

Cuando queremos que el usuario elija **una única opción entre varias**, podemos hacer uso de los botones de radio, que se consiguen con la etiqueta:

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

Donde **yyy** es el nombre que le ponemos a la variable que se trata de elegir, y **zzz** es el nombre de cada una de las opciones en concreto.

Ejemplo: solicitamos al usuario que defina cuál es su sistema operativo preferido:

**<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">**

¿Cuál es tu sistema operativo preferido?

<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="PC" CHECKED> PC

<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="Mac"> Mac

<INPUT TYPE="radio" NAME="SistemaOperativo" VALUE="Unix"> Unix

</FORM>

Que resulta:

¿Cuál es tu sistema operativo preferido?

☒ PC ☐ Mac ☐ Unix

Obsérvese el atributo opcional **CHECKED** que se ha añadido en la primera etiqueta. Esa será la opción que aparece marcada por defecto.

Obsérvese también que no es posible escoger más de una opción.

Si el usuario ha escogido la opción PC y pulsa el botón de envío, recibiremos un e-mail suyo con el texto: SistemaOperativo=PC.

3. Botones de envío y de borrado

Hasta ahora, en todos los ejemplos que hemos visto, faltaba un elemento esencial en cualquier formulario, y es el botón de envío de los datos, que se consigue con la etiqueta:

<INPUT TYPE="submit" VALUE="zzz">

En donde **zzz** es el texto que queremos que aparezca en el botón.

Vamos a añadirlo al primer ejemplo, en el que se solicitaba el apellido del usuario:

**<FORM ACTION="mailto:dirección_de_email" METHOD="POST"
ENCTYPE="TEXT/PLAIN">**

Escribe tu apellido:

**
<INPUT TYPE="text" NAME="Apellido">
<P><INPUT TYPE="submit" VALUE="Enviar datos">
</FORM>**

Que resulta:

Escribe tu apellido:

Enviar datos

Otro botón interesante es el de **borrado** de los datos introducidos, muy conveniente en un formulario con muchos elementos. Es muy similar al de envío, pues se consigue con la etiqueta:

<INPUT TYPE="reset" VALUE="zzz">

En donde **zzz** es el texto que queremos que aparezca en el botón.

Si añadimos al ejemplo anterior la etiqueta:

<P><INPUT TYPE="reset" VALUE="Borrar datos">

resulta:

Escribe tu apellido:

Enviar datos

Borrar datos

Se puede comprobar su funcionamiento, escribiendo algo en el formulario y pulsando luego el botón de borrado.

Consideraciones finales

Hasta ahora hemos visto uno a uno los diferentes elementos que se pueden utilizar. Pero no hay ningún inconveniente en usar, dentro del mismo formulario, **distintos tipos** de introducción de datos. Al pulsar el usuario el botón de envío recibiríamos en e-mail suyo con las distintas parejas **NAME=VALUE** de cada elemento, encadenadas con el símbolo **&**.

Por ejemplo, con este formulario, con dos elementos distintos:

Escribe tu apellido:

¿Cuál es tu color preferido?

recibiríamos un e-mail con el texto (por ejemplo):

Apellido=Ruiz

ColorPreferido=Azul.

Agrupando elementos

Todos los elementos de un formulario pueden también ser agrupados temáticamente con el [tag HTML fieldset](#). Este tag contiene a un grupo de controles que se relacionan unos con otros de alguna forma (por ejemplo, información personal, información laboral, información financiera, intereses, etc.).

El título para cada grupo de elementos puede establecerse con el [tag HTML legend](#) que debe ser definido justo a continuación del tag de apertura del fieldset, y proveer un título descriptivo para el grupo de controles.