MINISTERIO DE EDUCACION

CENTRO EDUCATIVO BILINGÜE BELLAS LUCES

SEGUNDO TRIMESTRE

MATERIA: INFORMATICA NIVEL: DUODECIMO GRADO

NOMBRE: ERIC MARTÍNEZ B.

TEMA 2: FORMULARIOS – HTML

FORMULARIOS – HTML

Un formulario dentro de una página web permite al usuario introducir datos los cuales son enviados a un servidor para ser procesados. Los usuarios pueden llenar dichos formularios usando casillas de selección, botones de opción, o campos de texto.

Además de servir como plantillas para nueva información, los formularios web también pueden ser usados para consultar y mostrar información existente.

Lo primero a tener en cuenta hace referencia a la estructura general. Así, todas las etiquetas que vamos a utilizar para el formulario tienen que estar ubicadas dentro de las etiquetas del formulario <form> y </form>.

<form action="nombrearchivo" method="post">

</form>

En donde: action define un fichero .jsp, .php, .asp, etc., que enviará los datos introducidos en el formulario al servidor para ser procesados y al final, con method indicamos el método "post" que garantiza una mayor seguridad para los datos enviados.

CAJAS DE TEXTO (Input)

Las cajas de texto son etiquetas que se utilizan para introducir información y son colocadas por medio de la etiqueta <input>.

<input type="text" size="valor" maxlength="valor" title="mensaje..." name="nom\_campo" autofocus required />

* Con type indicamos qué tipo de datos estamos pidiendo (en este caso 'texto normal').
* Si queremos un campo de texto más largo o extenso podemos especificarlo con size.
* Si queremos restringir la cantidad de caracteres podemos especificarlo con maxlength.
* Con title podemos incluir un texto adicional que aparecerá cuando el usuario coloque el cursor encima. El name es importante ya que es la manera en la que se identificará este valor al enviar la información (es nombre del campo).

Los nuevos atributos para las etiquetas <input> más habituales e importantes son:

* required: El campo es obligatorio.
* autofocus: Inicialmente el campo está seleccionado (sólo puede haber uno por formulario).
* placeholder: Se muestra un valor por defecto en gris que se elimina cuando el usuario introduce algún valor.
* autocomplete: Permite el rellenado automático del <input> en base al texto que vaya escribiendo el usuario, según los valores introducidos en ocasiones anteriores.

Ejemplo:

<input type="text" size="40" title="Escribe tu nombre completo..." name="nombre" autofocus required/>

CAJA DE CONTRASEÑA (PASSWORD)

<input type="password" size="7" title="Escribe la contraseña que desee..." name="password" required/>

Al detectar que en type se especifica el valor 'password' lo que el usuario escribe se enmascara con círculos de forma automática.

CORREO ELECTRÓNICO (EMAIL)

<input type="email" placeholder="usuario@dominio.com" size="40" title="Escribe tu correo electrónico..." name="mail" required/>

Como en type se especifica 'email' la autovalidación de este dato incluirá la búsqueda de una @ y un mínimo formato apropiado.

En este caso observamos el parámetro placeholder, que es el responsable de que inicialmente aparezca ya escrito en gris flojo (usuario@dominio.com). Automáticamente cuando el usuario escribe cualquier valor dicho texto se elimina, igualmente si el usuario borra todo el contenido, el texto especificado con placeholder vuelve a aparecer.

CAJAS TIPO DATE

<input type="date" value="2000-01-01" name="nacimiento" title="La fecha en la que naciste, ya que..."/>

El valor 'date' dentro de type identifica que el valor a utilizar es una fecha y que por lo tanto aparecerá el acceso a un calendario, que se puede ubicar directamente en una fecha inicial utilizando el parámetro 'value'.

Así, value permite especificar la fecha inicial por defecto que aparecerá inicialmente en el calendario. Sino se indica ningún valor aparecerá la fecha actual.

CAJAS DE RANGO

<input type="range" min="1" max="7" value="2" step="1" title="Necesitamos saber..." name="adiccion"/>

Al indicar en type el valor 'range' aparece una barra (slider) con la que el usuario puede introducir un valor numérico arrastrando la barra hasta el valor que corresponda.

Con min y max se especifica el valor mínimo y máximo que el usuario podrá introducir.

Para especificar un valor inicial utilizamos el parámetro value.

Con step podemos especificar el valor de incremento (o decremento) de cada punto del slider. Por ejemplo, un valor de "1" quiere decir que cada vez que el usuario mueve un espacio del slider se aumenta o se disminuye el valor de 1 en 1, y que además no serán válidos los decimales.

CAJA ENUMERADA (NUMBER)

<input type="number" min="100" max="9000" step="1" value="100" title="De momento el ganador..." name="comics"/>

Con **min** y **max** se especifica el valor mínimo y máximo que serán tomados como válidos.  
Para especificar un valor inicial utilizamos el parámetro **value**.  
Con **step** podemos especificar el intervalo o 'de cuánto en cuánto' aumentarán o disminuirán los datos introducidos. Por ejemplo, un valor de "10" quiere decir que cada vez que el usuario aumenta o reduce el valor éste se mueve de 10 en 10 y que además únicamente serán válidos los datos que sigan este intervalo (100, 110, 120, 130, etc...).

BOTONES

**1. Enviar (submit)**

<input type="submit" value="Enviar los datos"/>

Con type="submit" estamos añadiendo el típico botón "Enviar", en el que a través del parámetro value podemos indicar el texto que se mostrará dentro del botón.

En este caso la acción de enviar la información se realizará de la manera automática, enviando los datos a la dirección que se ha indicado en <form>.

**2. button**: Crea un botón 'vacío'. La acción que se ejecutará cuando el usuario haga clic en él. La acción a ejecutar se tendrá que añadir posteriormente con un lenguaje de programación indicando su nombre de id.

<input type="button" value="Operacion " id=”boton1”/>

**3. checkbox**: Crea cuadros de selección en el que se puede seleccionar más de un valor. Todos los checkbox tienen que tener el mismo valor en name.

Además, si queremos que uno de los valores esté marcado inicialmente por defecto, le añadiremos la propiedad checked

<input type="checkbox" name="vacaciones" value="Damasco" /> Damasco (Siria)

<input type="checkbox" name="vacaciones" value="Kabul" checked/> Kabul (Afganistán)

<input type="checkbox" name="vacaciones" value="Belgrado" /> Belgrado (Serbia)



**4. radio**: Crea círculos de selección en el que únicamente se puede seleccionar uno de ellos. Todos los radio button tienen que tener el mismo valor en name.

Si además queremos que un valor esté seleccionado inicialmente, añadiremos el parámetro 'checked' a la opción deseada (como 'Mujer' en el siguiente ejemplo).

<input type="radio" name="sexo" value="Mujer" checked/> Mujer

<input type="radio" name="sexo" value="Hombre"/> Hombre y sus derivados

<input type="radio" name="sexo" value="NS/NC"/> No estoy muy seguro/a



**5. Botón de resetear la información**

Es posible que el usuario una vez avanzado el formulario, quiera reiniciarlo desde el principio y volver a escribir la información. Para ello es muy útil el botón para borrar la información. El botón se escribe cambiando el type, que en el caso de estos botones es “reset”. El ejemplo sencillo de este botón tendría el siguiente código:

<input type="reset" value="Borrar información">

LISTAS DESPLEGABLES

La etiqueta <select> sirve para crear listas desplegables ( obligatoriamente no tiene que ser desplegable), y cada uno de los elementos de esa lista son especificados por las etiquetas OPTION u OPTGROUP dentro de ella.

Los atributos de la etiqueta select son:

* name: Representa el nombre asociado con la casilla de texto, que permite su identificación en el par nombre/valor.
* disabled: Crea un lista desactivada, que aparece atenuada
* size: Si un elemento SELECT se presenta como una lista con desplazamiento (“scrolled list box”), este atributo especifica el número de filas de la lista que deberían ser visibles al mismo tiempo.No es preciso que los agentes visuales presenten un elemento SELECT como una lista (“list box”); pueden usar cualquier otro mecanismo, como por ejemplo un menú desplegable (“drop-down menu”).
* multiple: Permite al usuario seleccionar varios campos de la lista
* Por cada opción a desplegar en la lista se utiliza la etiqueta <option> la cual puede tener como propiedades selected y value además de cualquiera de los atributos inherentes de los objetos html, que puedes revisar en esta entrada.

Ejemplo:



TABLAS

Tanto en HTML como en otros programas las tablas se utilizan para organizar datos, pero también nos puede ayudar en organizar la información en la pantalla, dado a que una celda de tabla puede contener cualquier elemento HTML, como por ejemplo un párrafo o un campo de texto de un formulario.

En HTML una tabla está formada por un conjunto de directivas, unas obligatorias y otras opcionales. La directiva principal que delimita e incluye las otras directivas es la directiva <table> y </table>.

Los elementos definitorios para una tabla son las filas, las columnas y las celdas. HTML ofrece directivas como <tr> para construir las filas y <td> para celdas de datos y <th> para celdas de títulos (cabeceras). Cada cruce entre las filas y las celdas forman las columnas. Al encontrar la directiva <th> el navegador convierte el texto que lo rodea en negrita y lo centra con respecto a la celda.

Para que el navegador imprime una tabla en la página Web necesita un conjunto básico de directivas: <table>, <tr> y <td>. El siguiente código es una estructura básica de una tabla:

<!doctype html>

<html>

<head>

<meta charset="iso-8859-1"/>

<title>ESTABLECIENDO TABLAS</title>

</head>

<body> <br><br>

</table>

<table border="1">

<thead>

<tr>

<th>Nombre</th>

<th>Apellido</th>

<th>Edad</th>

</tr>

</thead>

<tbody>

<tr>

<td>Juan</td>

<td>Fernandez</td>

<td>25</td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan="3">Informacion sobre el usuario Juan Fernandez</td>

</tr>

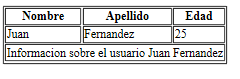
</tfoot>

</table>

</body>

</html>

Salida



VIDEO

Actualmente existen 3 estándares de vídeo que utilizan códecs diferentes y que vienen impulsados por grandes empresas.



El código HTML estándar para visualizar un único formato de vídeo sería:

<video>

<source src="video.ogv" type="video/ogg"/>

</video>

En el tag <video> es posible incorporar un gran número de atributos que nos ayudan a poder controlar mejor el video, como son:

• width: Indica la anchura del vídeo, expresada en píxeles (px) o en porcentaje (%)

• height: Indica la altura del vídeo, expresada en píxeles (px) o en porcentaje (%)

• controls: Establece si tendrá la barra de navegación (play, stop, volumen...) o no

• autoplay: Indica si el vídeo se inicia detenido o ya aparece por defecto reproduciéndose

• loop: Establece si al acabar el vídeo, éste debe volver a empezar o no

• muted: Indica si el audio se reproduce o se reproduce sin audio

• preload: Indica si el vídeo se empieza a cargar cuando el navegador carga la página html

• poster: Da la posibilidad de establecer una imagen (jpg, gif o png) como fotograma inicial del video.

Por lo tanto, si utilizásemos todos los atributos, el código anterior quedaría de la siguiente manera:

<video width="800" height="600" controls autoplay loop poster="fotograma.png">

<source src="video.ogv" type="video/ogg"/>

</video>

Si queremos que el vídeo se pueda visualizar desde cualquier navegador (que soporte HTML5), tenemos que añadir a este código más posibilidades y por lo tanto tenemos que tener el vídeo grabado en diferentes formatos.

El código HTML completo quedaría así:

<video width="800" height="600" controls autoplay loop poster="fotograma.png">

<source src="video.ogv" type="video/ogg"/>

<source src="video.mp4" type="video/mp4"/>

<source src="video.webm" type="video/webm"/>

<p>Tu navegador no soporta HTML5</p>

</video>

Si además queremos asegurarnos de que el vídeo también se podrá visualizar en navegadores antiguos tendremos que añadir la posibilidad de visualizar el vídeo en el formato de Flash (.flv), que es soportado por todos los navegadores, quedando el código de la siguiente manera:

<video width="800" height="600" controls autoplay loop poster="fotograma.png">

<source src="video.ogv" type="video/ogg"/>

<source src="video.mp4" type="video/mp4"/>

<source src="video.webm" type="video/webm"/>

<object width="800" height="600" data="video.flv">

<param name="movie" value="video.flv">

<embed src="video.flv" width="800" height="600">

</object>

<p>Tu navegador no soporta HTML5</p>

</video>

En este ejemplo se han utilizado únicamente los atributos 'width', 'height', 'controls' y 'poster', evitando utilizar la opción 'autoplay' (ya que no nos interesa que el video se reproduzca automáticamente al cargar la página).

Para añadir un vídeo de Youtube lo único que tenemos que hacer es:

1. Ir a la web de www.youtube.com y acceder al vídeo que quieras añadir a tu web.

2. Hacer clic con el botón derecho del ratón sobre el vídeo

3. Seleccionar la opción "Copiar código de inserción".

4. Ya dentro del editor de HTML, crear un DIV y pegar dentro el contenido que se ha copiado en el punto anterior, que será algo parecido al siguiente código:

<iframe width="640" height="360" src="http://www.youtube.com/embed/

\_93kDSPlKIM?feature=player\_detailpage" frameborder="0" allowfullscreen></iframe>