



CEU

| *Centro de Estudios  
Profesionales*

LENGUAJE DE MARCAS  
1º DAW – 1º DAM

---

# TEMA 16

## Formularios

### 1/3

# Elemento form

El elemento “form” se utiliza para crear formularios de entrada de datos para poder interactuar con el usuario.

El elemento “form” es un contenedor que incluirá dentro de su bloque diferentes tipos de elementos tales como: campos de texto, checkbox, radio buttons, submit butons, labels, etc.

Para crear un bloque formulario se emplea la directiva “<form>”:

```
<form>
```

```
.
```

Elementos dentro del bloque form ....

```
.
```

```
</form>
```

# Elemento <input>

El elemento “input” es de los más usados en los formularios, sirve para poder obtener la entrada de datos de un usuario, el campo “input” puede ser mostrado y puede tener varios usos o tipos representados con el atributo “type”, entre los más importantes destacamos:

<code>&lt;input type="text"&gt;</code>	Muestra un campo de entrada simple de texto.
<code>&lt;input type="radio"&gt;</code>	Muestra un campo de selección, para seleccionar una entre varias opciones.
<code>&lt;input</code>	Muestra un campo de selección, para



<code>type="checkbox"&gt;</code>	seleccionar una o varias opciones “multilista”
<code>&lt;input type="submit"&gt;</code>	Muestra un botón de envío del formulario a un servicio (“en el lado del servidor”).
<code>&lt;input type="button"&gt;</code>	Muestra un botón para realizar alguna acción.

Vamos a crear un archivo html denominado tema16.html con el siguiente contenido:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <!-- ETIQUETA PARA HACER RESPONSIVE LA PÁGINA, ADAPTACIÓN DE ESCALAS -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tema 16 html</title>
  <!-- ICONO DE VENTANA -->
  <link rel="icon" type="image/x-icon" href="imagenes/favicon.ico">
  <!-- HOJA DE ESTILOS EXTERNA-->
  <link rel="stylesheet" href="estilos/estilostema16.css">
</head>
<body>
  <h2> CAMPOS DE ENTRADA DE TEXTO</h2>
  <!-- FORMULARIO DE ENTRADA DE DATOS (CAMPOS DE TEXTO)-->
  <form>
    <!-- ETIQUETA LABEL PARA PODER ESCRIBIR ETIQUETAS DE TEXTO-->
    <label>Nombre:</label><br>
    <!-- TAG INPUT PARA PODER INTRODUCIR TEXTO EL USUARIO -->
    <input type="text" name="idNombre"><br>
    <label>Apellidos:</label><br>
    <input type="text" name="idApellidos">
  </form>
</body>
```

En el archivo estilostema16.css escribiremos:



```
body {  
  font-family: Arial, sans-serif;  
  margin-left: 10px;  
  padding: 5px;  
}  
  
h2 {  
  color: #4d77ca;  
  text-align: center;  
}
```

El resultado del ejercicio sería:

#### CAMPOS DE ENTRADA DE TEXTO

Nombre:

Apellidos:

Podemos observar que el ancho de los campos de texto por defecto es de 22 caracteres sin no especificamos la anchura del campo (ya lo veremos más adelante).

El atributo “name” nos va a servir para poder recoger el valor del campo de texto que haya introducido el usuario. Es obligatorio ponerlo, porque si enviamos un formulario y emitimos el campo “name” no se enviará ningún valor.

#### Uso de “for” e “id”:

Actualiza tu documento y vamos a agregar un atributo “for” en nuestros label y un atributo “id” en nuestros campos input. Deben coincidir.



```
<!-- FORMULARIO DE ENTRADA DE DATOS (CAMPOS DE TEXTO)-->
<form>
  <!-- ETIQUETA LABEL PARA PODER ESCRIBIR ETIQUETAS DE TEXTO-->
  <label for="idNombre">Nombre:</label><br>
  <!-- TAG INPUT PARA PODER INTRODUCIR TEXTO EL USUARIO -->
  <input type="text" id="idNombre" name="idNombre"><br>
  <label for="idApellidos">Apellidos:</label><br>
  <input type="text" id="idApellidos" name="idApellidos">
</form>
</body>
```

No es obligatorio poner el atributo “for” en los label, pero es una buena práctica y recomendable. Cada campo “for” va unido con un campo “input”, para que cuando realices click en el label, automáticamente el foco del ratón se va directamente a su campo input referenciado.

Si pulsáis en la etiqueta “Nombre” se posiciona el foco de entrada en el campo input del nombre y lo mismo pasa con la etiqueta “Apellidos”.

Las etiquetas “label” son también útiles para los lectores de pantalla “para personas invidentes” porque lee en voz alta la etiqueta cuando está posicionado en el campo de entrada.

## Radio buttons

El elemento `<input type="radio">` define un radio button.

Permite seleccionar una única entrada de una lista disponible de opciones.

Añadamos el siguiente código a nuestra página html:



```
<label for="idApellidos">Apellidos:</label><br>
<input type="text" id="idApellidos" name="idApellidos">
<!-- USO DE RADIO BUTTON -->
<br><br>
<label>Selecciona tu especialidad:</label>
<br>
<input type="radio" id="idDAW" name="idCICLO" value="DAW">
<label for="idDAW">DAW</label><br>
<input type="radio" id="idDAM" name="idCICLO" value="DAM">
<label for="idDAM">DAM</label><br>
<input type="radio" id="idASIR" name="idCICLO" value="ASIR">
<label for="idASIR">ASIR</label>
</form>
</body>
```

Podemos observar que el resultado de salida es:

Selecciona tu especialidad:

- ☐ DAW
- ☐ DAM
- ☒ ASIR

La etiqueta “for” del label y la etiqueta “id” de cada input, sirve como dijimos para emparejarlos, si pulsas en el texto de cada opción se pone chequeado su respectivo radio button, prueba a cambiar el id y verás que no tiene ese efecto.

El atributo “value” es el valor que se devuelve cuando hagamos referencia al campo de radio button.

El atributo “name” vemos que es el mismo en todos los input que componen el listado de opciones, es el identificador al que haremos referencia par obtener el valor que se ha seleccionado en el radio button.



# Check Boxes

El elemento `<input type="checkbox">` define un campo de chequeo. Este campo permite al usuario obtener uno o más opciones de un listado de opciones previamente definido.

Escribamos el siguiente código en nuestro archivo html:

```
<input type="checkbox" id="idDAM" name="idDAM" value="DAM">
<label for="idDAM">DAM</label><br>
<input type="radio" id="idASIR" name="idCICLO" value="ASIR">
<label for="idASIR">ASIR</label>
<!-- USO DE CHECKBOXES -->
<br><br>
<label>Selecciona las opciones que correspondan:</label>
<br>
<input type="checkbox" id="idCarnet" name="idCarnet" value="carnet">
<label for="idCarnet">Tengo carnet de conducir</label><br>
<input type="checkbox" id="idVehiculo" name="idVehiculo" value="vehiculo">
<label for="idVehiculo">Tengo vehículo propio</label><br>
<input type="checkbox" id="idIngles" name="idIngles" value="ingles">
<label for="idIngles">Domino el Inglés</label>
</form>
</body>
```

Podemos comprobar que la relación de los label con los campos input sigue igual.

Con respecto al atributo “name”, ya no es el mismo, cada input tiene el suyo propio, porque no están entrelazados, puede haber uno o más chequeados. El atributo “value” será el valor del campo que tendrá en caso de ser chequeado “veremos más adelante las comprobaciones de lo que hemos ido introduciendo en cada campo”.



# Botón Submit

El botón submit, define un botón para enviar los datos de un formulario a un controlador de formulario (Suele ser un script para procesar los datos que llegan como entrada), este script puede estar escrito en cualquier lenguaje (PHP, Python, Java, etc...)

Todos los datos que se envían son de campos de entrada, el campo de entrada que no tenga su atributo “name” no será enviado.

Agreguemos el siguiente código al final del body de nuestro archivo html:

```
<label for="idIngles">Domino el Inglés</label>
</form>
<br><br>
<h2> FORMULARIO DE ENVÍO DE 2 CAMPOS</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idNombre">Nombre:</label><br>
  <input type="text" id="idNombre" name="idNombre" value="María"><br>
  <label for="idApellido">Apellidos:</label><br>
  <input type="text" id="idApellido" name="idApellido" value="Dorado Sanz"><br><br>
  <!-- BOTÓN SUBMIT-->
  <input type="submit" value="Submit">
</form>
</body>
```

Vemos el atributo “action” del formulario que es dónde se especifica la ruta del archivo o servicio que queremos ejecutar cuando se pulse en el botón “submit” de nuestro formulario. Vemos que hemos repetido identificadores porque están en otro formulario.

El resultado será:





#### CAMPOS DE ENTRADA DE TEXTO

Nombre:

Apellidos:

Selecciona tu especialidad:

- ☐ DAW
- ☐ DAM
- ☐ ASIR

Selecciona las opciones que correspondan:

- ☐ Tengo carnet de conducir
- ☐ Tengo vehículo propio
- ☐ Domino el Inglés

#### FORMULARIO DE ENVÍO DE 2 CAMPOS

Nombre:

Apellidos:

Como ejercicio, incluye en el primer formulario de nuestra página html el mismo atributo “action” que hemos visto en el formulario anterior e incluye un botón “submit”, de esta forma podremos ver los valores que hemos rellenado en dicho formulario.

### Atributo “target”

Este atributo que ya lo hemos visto en otros elementos html, nos va a decir dónde queremos abrir la página de destino del atributo “action” de nuestro formulario:

Ejemplo:

```
<form action="/action.php" target="_blank">
```

Lo abriría en otra pestaña.

Por defecto, si no ponemos nada lo abre en la misma ventana en la que tenemos el formulario. Los valores posibles que puede tener este atributo son:



<code>_blank</code>	Lo abre en una nueva ventana o pestaña
<code>_self</code>	Lo abre en la actual pestaña o ventana
<code>_parent</code>	Lo abre en la ventana padre
<code>_top</code>	Lo abre en la ventana completa
<code>framename</code>	Lo abre en el iframe que le indiquemos

Podemos probar a cambiar los actions de nuestros dos formularios y ponerlos que los abra en otra pestaña. Aunque lo normal en estos casos es que se abra en la misma.

### Atributo “Method”

Va a indicar el método http que se usa en la llamada al servicio. Por defecto es un método “get”.

Ejemplo:

`<form action="/action_page.php" method="get">` “Método de llamada a servicio sin paso de parámetros por formulario”. Son servicios sin envío de datos confidenciales. Los parámetros en caso de existir se envía en la propia url y son visibles.



`<form action="/action_page.php" method="post">` “Método de llamada a servicio con paso de parámetros por formulario”. No se ven los datos de envío, se utiliza para el envío de datos confidenciales.

## Atributo “autocomplete”

El atributo autocomplete puede tener dos valores posibles “on” “off”, en caso de estar a “on” los campos a rellenar en el formulario pueden mostrar valores por defecto que el usuario ha introducido previamente.

El valor por defecto si no se especifica dicho atributo es “on”.

En nuestro ejercicio html, pongamos el atributo en “off” en el último formulario que hemos creado y veamos su comportamiento:

```
<!-->
<h2> FORMULARIO DE ENVÍO DE 2 CAMPOS</h2>
<form action="https://www.w3schools.com/action_page.php" autocomplete="off">
  <label for="idNombre">Nombre:</label><br>
  <input type="text" id="idNombre" name="idNombre" value="María"><br>
  <label for="idApellido">Apellidos:</label><br>
  <input type="text" id="idApellido" name="idApellido" value="Dorado Sanz"><br><br>
  <!-- BOTÓN SUBMIT-->
  <input type="submit" value="Submit">
</form>
/body>
```

Ahora, si podéis observar el primer formulario, si pulsáis en los campos input, os saldrán los valores recomendados que habéis introducido antes, pero en el segundo formulario no saldrá ningún valor.

## PRIMER FORMULARIO



Cuando el foco del cursor se posa en el primer campo, muestra los valores que hemos introducido antes:

The screenshot shows a web form with the following elements:

- Nombre:** A text input field.
- Apellidos:** A text input field.
- Selección:** A dropdown menu that is open, showing a list of names: María, Fernando, adf, and adsf.
- Selección de opciones:** A section with the text "Selecciona las opciones que correspondan:" followed by three checkboxes:
  - ☐ Tengo carnet de conducir
  - ☐ Tengo vehículo propio
  - ☐ Domino el Inglés
- Submit:** A button at the bottom of the form.

Red rectangles highlight the "Nombre:" field, the "Apellidos:" field, and the dropdown menu.

## SEGUNDO FORMULARIO:

Cuando el foco del cursor se posa en el primer campo, no muestra ningún contenido que se haya introducido anteriormente.



Nombre:

Apellidos:

Submit

## Elemento <select>

El elemento “select” define una campo de formulario que mostrará una lista desplegable.

Escribe el siguiente ejemplo en nuestro archivo html:

```
</form>
<br><br>
<h2> FORMULARIO CON CAMPO DE LISTA DESPLEGABLE</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idLista">Seleccione asignatura:</label><br>
  <select id="idLista" name="idLista">
    <option value="lm">Lenguaje de Marcas</option>
    <option value="pr">Programación</option>
    <option value="so">Sistemas operativos</option>
    <option value="en">Entornos</option>
  </select>
  <br><br>
  <input type="submit" value="Enviar">
</form>
</body>
```

Si probáis a enviar el formulario, veréis como muestra los valores de los atributos “value” que son los que se envían de cada elemento de la lista.



Habéis podido comprobar que por defecto, se selecciona el primer elemento de la lista:

#### FORMULARIO CON CAMPO DE LISTA DESPLEGABLE

Seleccione asignatura:  
Lenguaje de Marcas ▼

Enviar

Podemos cambiar el valor del elemento seleccionado, utilizando el atributo:

**“selected”**

Hagamos el siguiente cambio en nuestro archivo html:

```
<h2> FORMULARIO CON CAMPO DE LISTA DESPLEGABLE</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idLista">Seleccione asignatura:</label><br>
  <select id="idLista" name="idLista">
    <option value="lm">Lenguaje de Marcas</option>
    <option value="pr">Programación</option>
    <option value="so">Sistemas operativos</option>
    <option value="en" selected>Entornos</option>
  </select>
  <br><br>
  <input type="submit" value="Enviar">
</form>
</body>
```

Podemos comprobar que ahora al ejecutar nuestra página html, el valor seleccionado por defecto es el que hemos fijado con el atributo “selected”, en este caso la opción de Entornos:



Seleccione asignatura:

Entornos ▼

Enviar

Podemos hacer que se muestren más valores de inicio, actualmente se muestra el valor preseleccionado y se tiene que desplegar para mostrar los demás valores. Vamos a utilizar el atributo “size” para poder cambiar esto.

### Atributo “size”

Vamos a poner lo siguiente en nuestro ejercicio html:

```
<h2> FORMULARIO CON CAMPO DE LISTA DESPLEGABLE</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idLista">Seleccione asignatura:</label><br>
  <select id="idLista" name="idLista" size="4">
    <option value="lm">Lenguaje de Marcas</option>
    <option value="pr">Programación</option>
    <option value="so">Sistemas operativos</option>
    <option value="en" selected>Entornos</option>
  </select>
  <br><br>
  <input type="submit" value="Enviar">
</form>
</body>
```

Ahora podemos ver que hemos dejado 4 líneas visibles, que coincide con las 4 opciones, por eso no está activo el scroll:



Seleccione asignatura:

Enviar

Si colocáramos de tamaño 2 opciones, el scroll estaría activo, pruébalo.

Con size="2":

Seleccione asignatura:

Enviar

Por defecto hemos visto que nuestra lista desplegable es de una única selección, pero podemos necesitar una lista desplegable que contenga varios elementos seleccionables, para ello utilizamos el atributo:

**“multiple”**

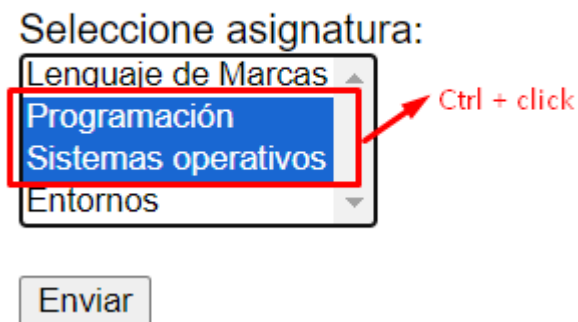
Escribamos en nuestro código dicho atributo para ver su comportamiento:





```
<h2> FORMULARIO CON CAMPO DE LISTA DESPLEGABLE</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idLista">Seleccione asignatura:</label><br>
  <select id="idLista" name="idLista" size="4" multiple>
    <option value="lm">Lenguaje de Marcas</option>
    <option value="pr">Programación</option>
    <option value="so">Sistemas operativos</option>
    <option value="en" selected>Entornos</option>
  </select>
  <br><br>
  <input type="submit" value="Enviar">
</form>
/body>
```

De esta forma, podremos utilizar la tecla “Ctrl” con el cursor del ratón para poder seleccionar más de una opción.



Si pulsamos en “Enviar” el formulario, podremos ver cómo se envían los valores seleccionados (pr y so) → Programación y Sistemas operativos:



## Submitted Form Data

Your input was received as:

idLista=pr&idLista=so

The server has processed your input and returned this answer.

**Note:** This tutorial will not teach you how servers are processing input. Processing input is explained in our [PHP tutorial](#).

## Atributo “optgroup”

Este atributo sirve para agrupa elementos por una categoría dentro de la lista desplegable.

Vemos el ejemplo, escribamos el siguiente código en nuestro archivo html:

```
<h2> FORMULARIO CON CAMPO DE LISTA DESPLEGABLE (AGRUPADA POR CATEGORÍAS)</h2>
<form action="https://www.w3schools.com/action_page.php">
  <label for="idLista">Seleccione asignatura:</label><br>
  <select id="idLista" name="idLista" size="7" multiple>
    <optgroup label="Asignaturas de primero">
      <option value="lm">Lenguaje de Marcas</option>
      <option value="pr" selected>Programación</option>
      <option value="so" selected>Sistemas operativos</option>
      <option value="en" selected>Entornos</option>
    </optgroup>
    <optgroup label="Asignaturas de segundo">
      <option value="dm">Dispositivos móviles</option>
      <option value="sa">Sistemas avanzados</option>
      <option value="in">Inglés</option>
    </optgroup>
  </select>
  <br><br>
  <input type="submit" value="Enviar">
</form>
```

Podemos observar que hemos creado 2 grupos, con asignaturas de primero y asignaturas de segundo, también hemos puesto 3 asignaturas preseleccionadas al ser multi seleccionable. El resultado sería:



CEU

Centro de Estudios  
Profesionales

LENGUAJE DE MARCAS  
1º DAW – 1º DAM

---

FORMULARIO CON CAMPO DE LISTA DESPLEGABLE (AGRUPADA POR CATEGORÍAS)

Seleccione asignatura:

<b>Asignaturas de primero</b>	
Lenguaje de Marcas	
Programación	
Sistemas operativos	
Entornos	
<b>Asignaturas de segundo</b>	
Dispositivos móviles	

Enviar