



# Diseño Web Responsive HTML5 y CSS3

Unidad 6: Multimedia y Formularios



## *Indice*

### **Unidad 6: Multimedia y Formularios**

#### Formularios en HTML5





## Objetivos

### Que el alumno logre:

- Conocer los nuevos elementos semánticos incorporados en la versión HTML5.





## Formularios en HTML 5

Los formularios son elementos que han tenido una gran evolución a lo largo del desarrollo de la web. Su misión primordial siempre ha sido permitir que el usuario pueda ingresar información para que sea procesada por un servidor.

Esta operación puede ir desde el envío de un mail hasta la actualización en una base de datos, entre muchas otras posibilidades.

Los formularios pueden potenciarse con el uso de Javascript, ya que es posible utilizar esta tecnología para validar los datos antes de que lleguen al servidor.

### Etiqueta form

El elemento **form** es el que agrupa cada uno de los formularios.

Actúa como contenedor y a la vez permite que se ejecuten las entradas de datos.

Tiene algunos atributos que permiten que el usuario interactúe de formas distintas:

- **action:** Es uno de los más importantes, ya que define la dirección URL a la que se enviará la información del formulario (habitualmente una página en PHP o ASP que procesará el requerimiento del formulario).
- **accept-charset:** es posible indicar el juego de caracteres que puede manejar el servidor que recibe información del formulario. Por ejemplo, se puede utilizar el varlo UTF-8.
- **autocomplete:** Permite sólo *on* y *off* (por defecto *on*) y permite que el navegador rellene aquellos campos *input* de forma automática.
- **enctype:** Indica la codificación de la información que se enviará al servidor.
  - Tiene 3 posibles estados:
    - *application/x-www-form-urlencoded*
      - por defecto si no se indica.
    - *multipart/form-data*
      - no se codifican los caracteres, debe emplearse cuando se utiliza en el formulario algún campo con opción de upload)
    - *text/plain*



- los espacios se convierten en el símbolo +, y el resto de los caracteres especiales se mantienen sin codificar.
- **method**: Indica el tipo de cabecera por el que se enviará la información.
  - Permite 2 estados:
    - **GET**
      - Con get pasamos la información (nombre=valor) por medio de la URL y será visible en la barra del navegador. No es recomendable utilizar este método para información privada, pero puede ser de utilidad en los casos en los que se quiera que el usuario guarde la dirección en el historial o en favoritos.
    - **POST**
      - Es recomendable para las transacciones con el servidor donde no deseamos que los datos pasen por la URL, y no pone un límite de caracteres para el traspaso de datos.
- **name**: Es el nombre del formulario. Si se indica no puede estar vacío, pero puede no ponerse (y el sistema lo creará automáticamente de forma interna).
- **novalidate**: Un atributo booleano que indica si la información se validará o no.
- **target**: Al igual que los enlaces, permite los valores:
  - `_blank`,
  - `_self` (por defecto),
  - `_parent`
  - `_top`.

## Etiqueta input

El principal elemento para la introducción de datos sigue siendo el **input**, y en esta ocasión llega con muchísimos atributos y con muchos tipos de entradas.

El principal atributo del sistema de entrada de datos es el **type**.

Hasta ahora los valores más habituales eran *hidden* o *text* que básicamente eran textos alfanuméricos, pero ahora eso ha cambiado, y la lista de tipos es bastante grande:



- **hidden**: Es un contenido oculto que suele corresponder a una serie de caracteres alfanuméricos. Este valor no puede ser manipulado por el usuario.
- **text**: Campo de texto que no permite saltos de línea. Si se añade el atributo **value** será el contenido original del campo.
- **search**: Campo de texto que no permite saltos de línea y que indica que será usado como contenido de una búsqueda. Si se añade el atributo **value** será el contenido original del campo.
- **tel**: Campo de texto que indica un número de teléfono. No tiene ningún formato en particular, por lo que se pueden incluir espacios o guiones, por ejemplo.
- **url**: Campo de texto en el que se pueden incluir direcciones URL absolutas. Se puede hacer una especie de *autosuggest* junto a otros elementos.
- **email**: Campo de texto que permite incluir correo electrónico. Principalmente si se incluye el atributo **value** permite una única dirección, y si es el **múltiple** puede incluir varias.
- **password**: Campo de texto, pero que aparece oculto para el usuario y así proteger la información.
- **datetime**: Permite una fecha y hora válida en un formato como *0037-12-13T00:00Z*, *1979-10-14T12:00:00.001-04:00* o *8592-01-01T02:09+02:09*. Si se indica un número, serán los milisegundos transcurridos desde el 01/01/1970.
- **date**: Permite indicar una fecha. El atributo **value** indica la fecha en cuestión y el **min** y **max** los límites del rango de fecha. El atributo **step** marca los milisegundos (por defecto 1 día) entre un valor y otro. El formato sería *2010-03-09*.
- **month**: Permite indicar un mes. Tiene los mismos atributos que **date**. El formato sería *2010-03*.
- **week**: Permite indicar la semana del año. Tiene los mismos atributos que **date**. El formato sería *2010-W03* (tercera semana).
- **time**: Permite indicar una hora. Tiene los mismos atributos que **date**. El formato sería *17:23* y opcionalmente los segundos, *17:23:59*.
- **datetime-local**: Es igual que el **datetime** pero no permite indicar ningún tipo de desfase horario.
- **number**: Representa un valor numérico. Este puede ser negativo (si empieza con un guión -), seguido de un número y con decimales (si se le indica un punto .) y para acabar, un exponente, empezando con una letra e, seguida de un símbolo positivo + o negativo - y la cantidad exponencial. Un ejemplo completo



sería: `-12345.678e+90`.

- **range**: El tipo *range* permite un *slide* en donde el número (valor) no es lo 100% importante, sino que es el formato en el que se muestra. Un poco más abajo, un ejemplo.
- **color**: Este elemento representa un color. El formato es del estilo `#A0F59E` (o sea, el RGB).
- **checkbox**: Este elemento ya existía, y es el que permite elegir unas cuantas opciones de un grupo de ellas.
- **radio**: Al igual que el anterior, permite elegir de un grupo, pero sólo una de las opciones.
- **file**: Éste permite la opción de subir archivos. Con él se pueden seleccionar archivos que luego serán enviados al sistema. Si se le indica el atributo *multiple* se podrán seleccionar varios ficheros a la vez. Además, si se indica el atributo *accept* se puede limitar ficheros según su MIME-type, por ejemplo con `audio/*`, `video/*` o `image/*`.
- **submit**: Este es el botón que permite el envío de información al sistema. Puede ir acompañado de otros atributos que hacen que “ese botón” haga cosas que no son las generales del formulario. Los parámetros son: *formaction* (que hace ese botón), *formenctype* (la codificación), *formmethod* (la forma de envío), *formnovalidate* (la validación) y *formtarget* (en qué “ventana” se abre)
- **image**: Es igual que el *submit* pero permite ser una imagen en vez de un botón en sí. Para ello ha de ir acompañado del atributo *src* que indica la dirección URL de la imagen.
- **reset**: Otro botón que si pulsas deja los valores del formulario en su valor inicial.
- **button**: Un botón que “no hace nada”. Suele utilizarse habitualmente con funciones JavaScript que son los que realizan la acción.

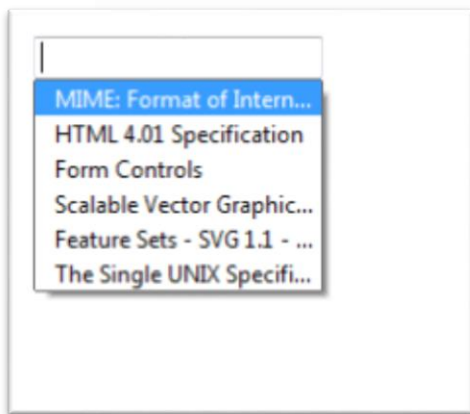
#### Ejemplo de *autosuggest* gracias al `type="url"`

```
<input type="url" name="location" list="urls">
<datalist id="urls">
  <option label="MIME: Format of Internet Message Bodies"
value="http://www.ietf.org/rfc/rfc2045">
  <option label="HTML 4.01 Specification" value="http://www.w3.org/TR/html4/">
  <option label="Form Controls" value="http://www.w3.org/TR/xforms/slice8.html#ui-
commonelems-hint">
  <option label="Scalable Vector Graphics (SVG) 1.1 Specification"
value="http://www.w3.org/TR/SVG/">
```



```
<option label="Feature Sets - SVG 1.1 - 20030114"
value="http://www.w3.org/TR/SVG/feature.html">
  <option label="The Single UNIX Specification, Version 3" value="http://www.unix-
systems.org/version3/">
</datalist>
```

### Visualización desde Firefox



### Ejemplo de *range*

```
<input type="range" min="-100" max="100" value="0" step="10" name="power"
list="powers">
<datalist id="powers">
  <option value="0">
  <option value="-30">
  <option value="30">
  <option value="+60">
</datalist>
```





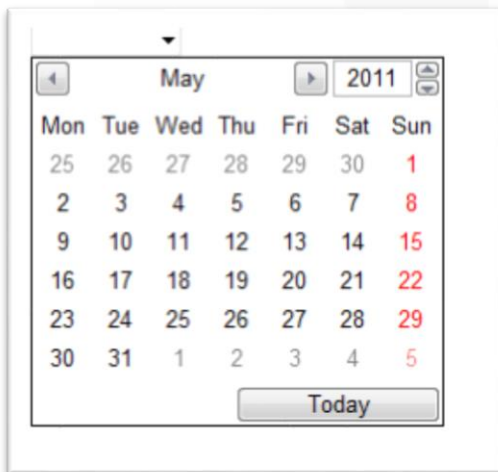
Visualización en Google Chrome



Ejemplo de date:

```
<input name="birthday" type="date">
```

Visualización en Opera:



Ejemplo de caja de búsqueda:

```
<form>
```

```
<input name="q" type="search">
```



```
<input type="submit" value="Find">  
</form>
```

Visualización en Chrome:

A screenshot of a web browser window showing a search interface. It consists of a text input field containing the word 'buscar' and a button labeled 'Find' to its right. The input field has a small 'x' icon inside, likely for clearing the text.

Además de los distintos tipos que puede haber, existen distintos atributos:

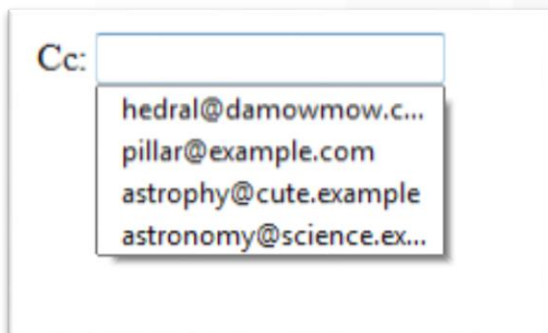
- **autocomplete**: Permite el estado *on* y *off* y hace que el campo se rellene o no automáticamente si el sistema es capaz de identificar la información.
- **list**: Es el valor de una lista identificada por el *datalist*. Con esto se pueden crear sistemas de “autosugerencias”.
- **readonly**: Es un valor booleano, y permite o no que el usuario pueda modificar el contenido.
- **size**: Es el número de caracteres que se muestran por pantalla.
- **required**: Otro atributo booleano que indica que ese elemento ha de incluirse obligatoriamente.
- **multiple**: Permite que algunos elementos seleccionen más de un valor.
- **maxlength**: Es la cantidad máxima de caracteres que se permite.
- **pattern**: Indica el formato de contenido que tiene que tener... Por ejemplo: `[0-9][A-Z]{3}` sería un número seguido de 3 letras en mayúsculas.
- **min y max**: Indican los valores mínimo y máximo que se permiten.
- **step**: Indica el valor de paso cuando hay elementos que permiten saltar.
- **placeholder**: Es un “ejemplo” del valor que se puede incluir. Desaparece automáticamente al hacer clic sobre él.



Ejemplo de selección múltiple de cuentas de correo:

```
<label>Cc: <input type=email multiple name=cc list=contacts></label>
<datalist id="contacts">
  <option value="hedral@damowmow.com">
  <option value="pillar@example.com">
  <option value="astrophy@cute.example">
  <option value="astronomy@science.example.org">
</datalist>
```

Visualización en Firefox:





## Resumen

### En esta Unidad...

Trabajamos con los nuevos elementos multimediales de HTML5.

### En la próxima Unidad...

Comenzaremos a trabajar con Tipografías, Íconos y Animación en CSS3.