

# HTML-CSS

*Angel González*



# Formularios Avanzados





## Formularios Web

`<form>..</form>`

/ Encierra todos los componentes del formulario

/ **Atributos:**

- Sigue teniendo los atributos **name**, **action** y **method**

```
<form name="miformulario" action="recoge.php" method="get">  
</form>
```

- También **enctype**, con sus posibles valores:

- **application/x-www-form-urlencoded**: Se codifican todos los caracteres convirtiendo los espacios en blanco en '+' y los demás caracteres especiales se convierten en valores hexadecimales ASCII.
- **multipart/form-data**: No se codifican los caracteres, y solo se usa cuando el formulario envía/sube ficheros al servidor.
- **text/plain**: Solamente se convierten los espacios en blanco en caracteres '+'

## Formularios Web



### `<input>`

- / Genera un campo de entrada para que el usuario introduzca datos en él.
  - HTML5 incorpora nuevos valores para el atributo **type**
  - Estos nuevos tipos:
    - Especifican qué clase de entrada es esperada
    - Le dicen al navegador qué debe hacer con la información recibida.
  - El navegador procesará los datos ingresados de acuerdo al valor del atributo **type** y validará o no la entrada.



## Formularios Web - <input>

- / Antiguos valores del atributo **type**, que siguen en uso:
- **text**: Para insertar texto
  - **password**: Genera un campo para introducir contraseñas. Ocultando los caracteres en el cuadro de texto.
  - **checkbox**: Genera casillas de verificación
  - **radio**: Genera un botón radio para seleccionar una opción entre varias propuestas
  - **submit**: Genera un botón para enviar el formulario
  - **button**: Genera un botón para realizar alguna acción que se indicará en código javascript
  - **reset**: Limpia el formulario



## Formularios Web - <input>

- **hidden**: Para enviar información complementaria, oculta al usuario
- **file**: Permite agregar archivos adjuntos

```
<form name="miformulario" action="recoge.php" method="get" enctype="multipart/form-data">
  Nombre: <input type="text" name="nombre"/><br/>
  Contraseña: <input type="password" name="pass"/><br/>
  Deportes: <br/>
    Correr: <input type="checkbox" name="correr" value="correr"/>
    Nadar: <input type="checkbox" name="nadar" value="nadar"/>
    Patinar: <input type="checkbox" name="patinar" value="patinar"/><br/>
  Género: <br/>
    Masculino: <input type="radio" name="genero" value="masculino"/>
    Femenino: <input type="radio" name="genero" value="femenino"/><br/>
    <input type="hidden" name="oculto" value="1"/>
  Archivo Adjunto: <input type="file" name="adjunto"/><br/>
    <input type="submit" value="Enviar"/>
    <input type="reset" value="Borrar"/>
</form>
```

Nombre:

Contraseña:

Deportes:

Correr: ☐ Nadar: ☐ Patinar: ☐

Género:

Masculino: ☐ Femenino: ☐

Archivo Adjunto:   No se ha seleccionado ningún archivo.





## Formularios Web - <input>

/ Nuevos valores para el atributo **type**:

- Search
- Url
- Date
- Datetime
- Datetime-local
- Month
- Time
- Week
- Datalist
- Keygen
- Output
- Number
- Range
- Tel
- Email
- Color



## Formularios Web - <input>

/ Nuevos valores para el atributo **type**:

- **number**: Limita la entrada del input a solo números
  - **min**: Valor mínimo que se aceptará en el campo
  - **max**: Valor máximo que se aceptará en el campo
  - **step**: Determina el valor en el que el campo aumentará o disminuirá
    - \* El tipo number, no admite atributos de estilo como size, deben aplicarse desde css

```
Puntos:<input type="number" name="puntos" min="0" max="10" step="2"/>
```

Puntos:



## Formularios Web - <input>



- **tel**: Para introducir números de teléfono
  - En un dispositivo móvil se detecta que el input es de tipo 'tel' y al pulsar en él automáticamente sale el teclado telefónico del móvil para insertar el número en cuestión.

```
Teléfono:<input type="tel" name="telefono" /><br/><br/>
```

Teléfono:

- **email**: Permite introducir una dirección de correo electrónico.
  - El texto introducido será comprobado por el navegador y validado si es una dirección correcta.
  - Si falla la validación, el formulario no se enviará
  - La respuesta del navegador a una entrada inválida no está determinada, dependerá del navegador..

```
email:<input type="email" name="correo" />
```

email:

! Incluye un signo "@" en la dirección de correo electrónico. La dirección "evahotmail.com" no incluye el signo "@".



## Formularios Web - <input>

- **url**: Funciona igual que email, pero para direcciones web.
  - Acepta solo direcciones absolutas
  - Devuelve un error si el valor es inválido

```
Página Web:<input type="url" name="web"/>
```

Página Web:

 Introduce una URL

- **search**: Indica que el campo es un buscador
  - No controla la entrada, es solo una indicación al navegador
  - El teclado del dispositivo móvil mostrará el botón 'buscar'.

```
<input type="search" name="buscar"/>
```

## Formularios Web - <input>



- **range**: Permite seleccionar una valor en un rango de números
  - \* **min**: establece el valor mínimo del rango
  - \* **max**: establece el valor máximo del rango
  - \* **value**: valor a mostrar inicialmente
  - \* **step**: valor en el que el número aumenta o disminuye en cada paso del rango
- Se utiliza javascript para mostrar los números en la pantalla

```
rango:<input type="range" name="rango" min="0" max="10" value="6" step="2"/>
```

rango:

## Formularios Web - <input>



- **date**: Proporciona una mejor forma de introducir fechas.
  - Los navegadores implementan esta función mostrando un calendario al hacer clic en el campo

```
Fecha Nacimiento:<input type="date" name="fechaNac"/><br/><br/>
```

Fecha Nacimiento:



## Formularios Web - <input>

- **week**: similar a date, pero solo recoge semanas enteras
  - El valor esperado tiene la sintaxis 2016-W35, donde 2016 es el año y 35 la semana

```
Semana:<input type="week" name="semana"/>
```

Semana: Semana 10, 2016 x ▴ ▾ ▼

- **month**: Permite recoger un mes entero
  - El valor esperado tiene la sintaxis año-mes

```
Mes:<input type="month" name="mes"/><br/><br/>
```

Mes: marzo de 2016 x ▴ ▾ ▼



## Formularios Web - <input>

- **time**: similar a date, recoge horas, minutos y segundos
  - El valor esperado es horas:minutos o horas:minutos:segundos, dependiendo del navegador se recogerán los segundos o no

```
hora:<input type="time" name="tiempo"/>
```

hora: 23:05 x

- **datetime**: Para introducir fecha y hora incluyendo zona horaria
  - **min**: Fecha mínima
  - **max**: Fecha máxima
  - **value**: Fecha por defecto
  - **step**: El intervalo de avance.
  - **disabled**: Determina si está desactivado el input para la introducción de datos (disable="disabled"), por defecto activado.

```
Fecha y Hora: <input type="datetime" name="fechahora" step="1"
min="2016-01-01T00:00Z" max="2016-12-31T12:00Z"
value="2016-01-01T12:00"/>
```

Fecha y Hora: 2016-01-01T12:00

- **datetime-local**: igual que datetime sin zona horaria

```
Fecha y Hora: <input type="datetime-local" name="fechahora"/>
```

Fecha y Hora: dd/mm/aaaa --:--

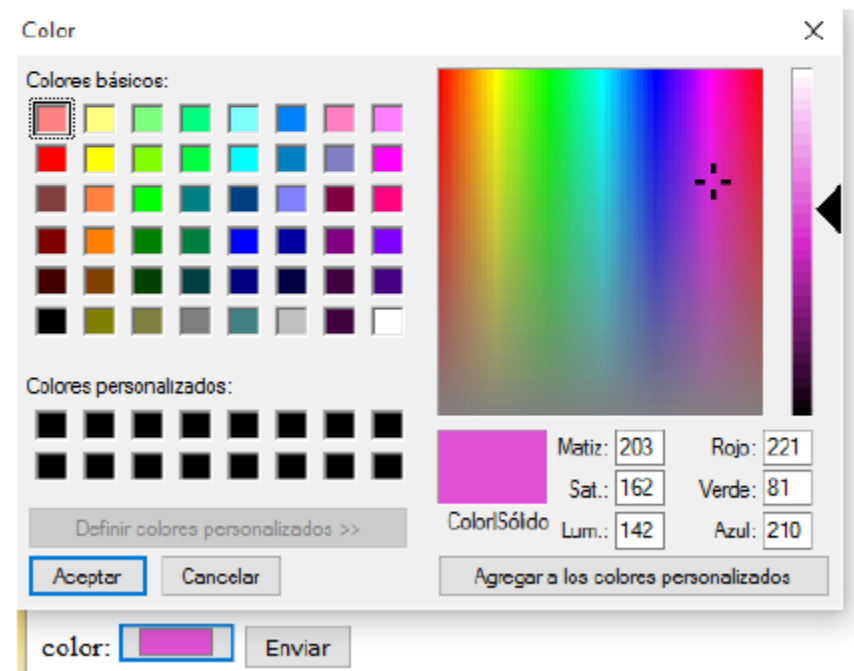
## Formularios Web - <input>



### – color:

- Permite seleccionar un color
- El valor esperado, es el código del color en hexadecimal

```
color: <input type="color" name="color"/>
```





## Formularios Web - <input>



- **datalist**: Facilita el autocompletado de un campo **tipo input**.
  - Es decir, según el usuario va escribiendo, el navegador abre una lista desplegable con las diferentes opciones a seleccionar.
  - Es muy parecido a un combo o lista.
  - El atributo '**list**' del input y el '**id**' del **datalist** tienen que coincidir, de lo contrario no se relacionarían

```
Escribe un color (Azul,rojo,amarillo,negro,verde):  
<input list="listacolores" name="color">  
  <datalist id="listacolores">  
    <option value="Azul">  
    <option value="Rojo">  
    <option value="Amarillo">  
    <option value="Negro">  
    <option value="Blanco">  
  </datalist>
```

Escribe un color (Azul,rojo,amarillo,negro,verde): a

Azul
Amarillo
Blanco

## Nuevos atributos



- **autocomplete** Este es un viejo atributo que se ha vuelto estándar en esta especificación.
  - \* Puede tomar dos valores: **on** y **off**.
  - \* El valor por defecto es **on**.
  - \* Cuando es configurado como **off** los elementos **<input>** pertenecientes a ese formulario tendrán la función de autocompletar desactivada, sin mostrar entradas previas como posibles valores.
  - \* Puede ser implementado en el elemento **<form>** (afecta a todos los **input**) o en cualquier elemento **<input>** independientemente.

```
<input type="search" name="buscar" autocomplete="off"/>
```

## Nuevos atributos



### – placeholder:

- Permite que aparezca un texto de ayuda en el campo
- Este texto desaparece cuando el usuario se posiciona dentro de el.

```
DNI:<input type="text" name="dni" placeholder="P.e. 11111111D"/>
```

DNI:



## Nuevos atributos

### – required

- Establece el campo como obligatorio
- Si no se rellena el formulario no se enviará

```
email<input type="email" name="correo" required/>
```

email

Rellene este campo.



## Nuevos atributos

### – multiple:

- Se utiliza en algunos tipos como **email** o **file**, para permitir múltiples entradas de un mismo campo
- Los valores insertados debe separarse con comas, para se válidos

```
email<input type="email" name="correo" multiple/>
```

email



## Nuevos atributos

### / autofocus:

- Pone el foco sobre el elemento seleccionado
- No mueve el foco cuando ya ha sido seleccionado otro elemento

```
<form >
<fieldset>
  <legend>autofocus</legend>
  DNI:<input type="text" name="dni" placeholder="P.e. 11111111D"/>
  email<input type="email" name="correo" required autofocus/><br/><br/>

  <input type="submit" name="enviar" value="Enviar" />
</fieldset>
</form>
```

autofocus

DNI:  email

## Nuevos atributos



### / pattern:

- Permite añadir validaciones a un campo, además de las previstas por defecto.
- Su valor será una expresión regular

```
código postal<input pattern="[0-9]{5}" name="codpostal" title="Escriba los 5 dígitos de su código postal"/>
```

código postal

Ajústese al formato solicitado: Escriba los 5 dígitos de su código postal.



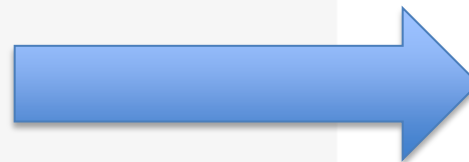
## Mostrar un botón como un enlace

Como ya se vio anteriormente, el estilo por defecto de los enlaces se puede modificar para que se muestren como botones de formulario. Ahora, los botones de formulario también se pueden modificar para que parezcan enlaces.

Las reglas CSS del ejemplo anterior son las siguientes:

```
.enlace {  
  border: 0;  
  padding: 0;  
  background-color: transparent;  
  color: blue;  
  border-bottom: 1px solid blue;  
}
```

```
<input type="button" value="Botón normal" />  
<input class="enlace" type="button" value="Botón como enlace" />
```



Botón normal

Botón como enlace

## Mejoras en los campos de texto

Por defecto, los campos de texto de los formularios no incluyen ningún espacio de relleno, por lo que el texto introducido por el usuario aparece pegado a los bordes del cuadro de texto.

Añadiendo un pequeño padding a cada elemento `<input>`, se mejora notablemente el aspecto del formulario:

The diagram illustrates the improvement of form input fields by adding padding. It consists of two parts, each enclosed in a box. The top part is labeled 'Formulario sin padding en los input' and shows two input fields: 'Nombre' with the text 'lorem ipsum' and 'Contraseña' with masked text '\*\*\*\*\*'. The text is flush with the left and right borders of the input boxes. A large blue arrow points from this top box to the bottom box. The bottom part is labeled 'Formulario con padding en los input' and shows the same two input fields. In this version, the text 'lorem ipsum' and '\*\*\*\*\*' is indented from the borders of the input boxes, creating a more professional and readable appearance.

La regla CSS necesaria para mejorar el formulario es muy sencilla:

```
form.elegante input {  
  padding: .2em;  
}
```

## Mejoras en los campos de texto

Por defecto, los campos de texto de los formularios no incluyen ningún espacio de relleno, por lo que el texto introducido por el usuario aparece pegado a los bordes del cuadro de texto.

Añadiendo un pequeño padding a cada elemento `<input>`, se mejora notablemente el aspecto del formulario:

Formulario sin padding en los input

Nombre

Contraseña

Formulario con padding en los input

Nombre

Contraseña

La regla CSS necesaria para mejorar el formulario es muy sencilla:

```
form.elegante input {  
  padding: .2em;  
}
```

## Labels alineadas y formateadas

Los elementos `<input>` y `<label>` de los formularios son elementos en línea, por lo que el aspecto que muestran los formularios por defecto, es similar al de la siguiente imagen:

```
<form>
  <fieldset>
    <legend>Alta en el servicio</legend>

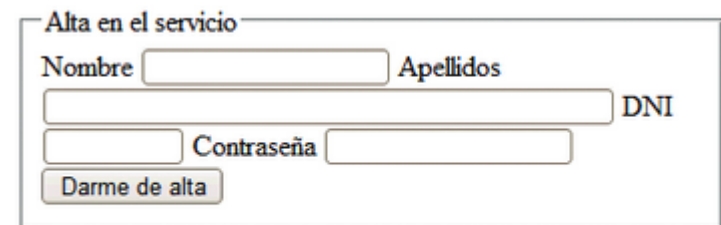
    <label for="nombre">Nombre</label>
    <input type="text" id="nombre" />

    <label for="apellidos">Apellidos</label>
    <input type="text" id="apellidos" size="50" />

    <label for="dni">DNI</label>
    <input type="text" id="dni" size="10" maxlength="9" />

    <label for="contrasena">Contraseña</label>
    <input type="password" id="contrasena" />

    <input class="btn" type="submit" value="Dar de alta" />
  </fieldset>
</form>
```



## Labels alineadas y formateadas

Aprovechando los elementos `<label>`, se pueden aplicar unos estilos CSS sencillos que permitan mostrar el formulario con el aspecto de la siguiente imagen:



En primer lugar, se muestran los elementos `<label>` como elementos de bloque, para que añadan una separación para cada campo del formulario. Además, se añade un margen superior para no mostrar juntas todas las filas del formulario:

```
label {  
    display: block;  
    margin: .5em 0 0 0;  
}
```

El botón del formulario también se muestra como un elemento de bloque y se le añade un margen para darle el aspecto final deseado:

```
.btn {  
    display: block;  
    margin: 1em 0;  
}
```

## Labels alineadas y formateadas

Para mostrar un formulario tal y como aparece en la imagen anterior no es necesario crear una tabla y controlar la anchura de sus columnas para conseguir una alineación perfecta. Sin embargo, sí que es necesario añadir un nuevo elemento (por ejemplo un `<div>`) que encierre a cada uno de los campos del formulario (`<label>` y `<input>`). El esquema de la solución propuesta es el siguiente:

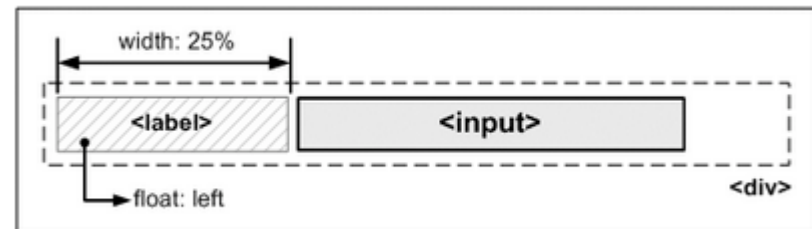
Alta en el servicio

Nombre

Apellidos

DNI

Contraseña



## Labels alineadas y formateadas

Por tanto, en el código HTML del formulario anterior se añaden los elementos `<div>`:

```
<form>
  <fieldset>
    <legend>Alta en el servicio</legend>

    <div>
      <label for="nombre">Nombre</label>
      <input type="text" id="nombre" />
    </div>

    <div>
      <label for="apellidos">Apellidos</label>
      <input type="text" id="apellidos" size="35" />
    </div>
    ...
  </fieldset>
</form>
```

Y en el código CSS se añaden las reglas necesarias p

```
div {
  margin: .4em 0;
}

div label {
  width: 25%;
  float: left;
}
```



## Formulario en varias columnas

Los formularios complejos con decenas de campos pueden ocupar mucho espacio en la ventana del navegador. Además del uso de pestañas para agrupar los campos relacionados en un formulario, también es posible mostrar el formulario a dos columnas, para aprovechar mejor el espacio.

Alta en el servicio	Datos de contacto
Nombre <input type="text"/>	Telefono <input type="text"/>
Apellidos <input type="text"/>	Email <input type="text"/>
DNI <input type="text"/>	Dirección <input type="text"/>
Contraseña <input type="text"/>	Código Postal <input type="text"/>

```
form fieldset {  
  float: left;  
  width: 48%;  
}
```

```
<form>  
  <fieldset>  
    ...  
  </fieldset>  
  
  ...  
</form>
```

## Resaltar el campo seleccionado

Una de las mejoras más útiles para los formularios HTML consiste en resaltar de alguna forma especial el campo en el que el usuario está introduciendo datos. Para ello, CSS define la pseudo-clase `:focus`, que permite aplicar estilos especiales al elemento que en ese momento tiene el foco o atención del usuario.

Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

```
input:focus {  
  border: 2px solid #000;  
  background: #F3F3F3;  
}
```