

Formularios. HTML.



El formulario: <form>

Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.

```
<form action="ejemplo.php" method="get">
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
  <p>Año de nacimiento: <input type="number" name="nacido" min="1900"></p>
  <p>Sexo:
    <input type="radio" name="hm" value="h"> Hombre
    <input type="radio" name="hm" value="m"> Mujer
  </p>
  <p>
    <input type="submit" value="Enviar">
    <input type="reset" value="Borrar">
  </p>
</form>
```



La etiqueta que delimita un formulario es la etiqueta <form> ...</form>. Los atributos más importantes de la etiqueta <form> son:

- **action:** contiene el nombre del agente que procesará los datos remitidos al servidor (por ejemplo, un script de PHP)
- **method:** define la manera de enviar los datos al servidor. Los valores posibles son:
 - **get:** los valores enviados se añaden a la dirección indicada en el atributo action
 - **post:** los valores se envían de forma separada

Si el atributo method no está establecido, el formulario se comporta como si el valor fuera get.

La etiqueta <form> es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.

Los etiquetas que crean los controles en los formularios son <input>, <button>, <select>, <optgroup>, <option> y <textarea>. Además, se pueden estructurar los controles con las etiquetas <fieldset> y <legend>. Por último, la etiqueta <label> permite mejorar la accesibilidad de los controles.

El navegador envía únicamente los datos de los controles contenidos en el formulario. En una misma página puede haber varios formularios que envíen datos al mismo o a diferentes agentes.

Atributos comunes de los controles

Algunos atributos se pueden utilizar en varios tipos de controles.

El atributo type de la etiqueta <input> indica el tipo de control de que se trata.

En HTML 4.01, los tipos de <input> eran: botón para enviar [submit](#), botón para reiniciar [reset](#), texto [text](#), contraseña [password](#), archivo [file](#), imagen [image](#) y oculto [hidden](#).

En HTML 5 se añadieron los tipos de <input>: botón [button](#), número [number](#), búsqueda [search](#), teléfono [tel](#), dirección web [url](#), dirección de correo [email](#), fecha [date](#), hora [time](#), color [color](#) y rango [range](#).

En HTML 5.1 se añadieron los tipos de <input>: mes [month](#), semana [week](#) y fecha y hora local [datetime-local](#).

El atributo name

El atributo name identifica al control.

Todos los controles de un formulario deben tener el atributo name porque, en general, sólo se envían los controles que lo tienen (aunque hay excepciones, como los controles de tipo imagen o los botones, que no los necesitan).

En general, los atributos name no se deben repetir, porque no se recibiría toda la información al enviarla. La única excepción es el [botón radio](#), como se explica en el apartado correspondiente.

El atributo value

El atributo value permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta y algunos controles no lo admiten (área de texto, archivo o imagen).

En algunos casos (botones) el valor se muestra al usuario y el usuario no puede modificarlo. En otros casos (cajas de texto, contraseña) el valor se muestra en la página y el usuario puede modificarlo. En otros casos (casillas de verificación, botones radio, opciones de menú, oculto) el valor no se muestra en la página y el usuario no puede modificarlo.

En algunos casos (botones, oculto, botones radio) el atributo value es necesario. En otros casos (casilla de verificación, opciones de menú) es conveniente. En otros casos (cajas de texto, contraseña, imagen) puede omitirse. En el caso del selector de archivo, los navegadores no lo admiten por motivos de seguridad.

El atributo required

El atributo required permite indicar qué controles es obligatorio rellenar para enviar el formulario.

```
<p>Nombre: <input type="text" name="nombre" required></p>
```

```
<p>Sexo:
  <input type="radio" name="hm" value="h" required> Hombre
  <input type="radio" name="hm" value="m" required> Mujer
</p>
```

Si alguna de las opciones de un botón radio tiene establecido el atributo required, el control es obligatorio.

```
<p>Sexo:
  <input type="radio" name="hm" value="h" required> Hombre
  <input type="radio" name="hm" value="m"> Mujer
</p>

<p>Edad:
  <input type="radio" name="edad" value="menor"> Menor
  <input type="radio" name="edad" value="adulto" required> Adulto </p>
```

El atributo placeholder

El atributo placeholder permite mostrar en los controles de texto un texto que desaparece al escribir en el control.

<p>Apellidos:
<input type="text" name="apellidos" size="40" placeholder="Escriba sus dos apellidos"> </p>

El atributo size

El atributo size permite establecer la longitud de los controles de texto (text, password, search, etc.). Por omisión, las cajas suelen tener 20 caracteres de longitud.

<p>Nombre: <input type="text" name="nombre"></p>
<p>Apellidos: <input type="text" name="apellidos" size="40"></p>
<p>Nacionalidad: <input type="text" name="nacionalidad" size="20"></p>

Los atributos maxlength y minlength

Los atributos maxlength y minlength permiten establecer la longitud máxima y mínima, respectivamente, que puede escribir el usuario en un control de texto (text, password, search, etc.).

<p>Usuario: <input type="text" name="usuario" maxlength="6"></p>
<p>Contraseña: <input type="password" name="password" minlength="6"></p>

El atributo autofocus

El atributo autofocus permite indicar el control que debe tener el foco al cargarse el formulario.

Este atributo facilita la accesibilidad del formulario pues permite que el usuario pueda empezar a rellenar el formulario sin necesidad de hacer clic en el elemento. Como el navegador desplaza automáticamente la página hasta la posición del elemento, se recomienda que el elemento con atributo autofocus se encuentren al principio de la página, para que el desplazamiento no confunda al usuario.

Nota: En estos apuntes (lecciones y ejercicios) no se utiliza este atributo porque como en las páginas se incluyen varios formularios en distintas posiciones de la página, el desplazamiento automático a un elemento con atributo autofocus puede confundir al lector.

El atributo disabled

El atributo disabled permite deshabilitar el control. Una vez deshabilitado, el control ni siquiera puede coger el foco.

<input type="submit" value="Enviar" disabled>

<input type="text" name="texto" disabled>

El atributo readonly

El atributo `readonly` hace que el control no sea modificable, aunque el control puede coger el foco.

```
<input type="text" name="texto" value="¡A que no me cambias!" readonly>
```

```
<input type="text" name="texto" placeholder="¡A que no me cambias!" readonly>
```

El atributo `tabindex`

El atributo `tabindex` permite controlar el orden en que el foco pasa de un elemento a otro mediante el tabulador (Tab para avanzar y Shift+Tab para retroceder). Los valores de `tabindex` pueden ser números naturales (incluido el cero), no necesariamente consecutivos. Si no está presente, los controles se visitan en el orden en que aparecen en el texto. Si está presente, los controles se visitan de menor a mayor.

```
<input type="text" name="texto1" placeholder="Texto 1">
<input type="text" name="texto2" placeholder="Texto 2">
<input type="text" name="texto3" placeholder="Texto 3">
<input type="text" name="texto4" placeholder="Texto 4">
```

```
<input type="text" name="texto1" placeholder="Texto 1" tabindex="5"> <input
type="text" name="texto2" placeholder="Texto 2" tabindex="8"> <input type="text"
name="texto3" placeholder="Texto 3" tabindex="7"> <input type="text"
name="texto4" placeholder="Texto 4" tabindex="6">
```

El atributo `accesskey`

El atributo `accesskey` establece la tecla que coloca el foco en un elemento de un formulario, pero los navegadores Google Chrome y Firefox no utilizan los mismos atajos de teclado, como se comenta en la [lección de diferencias entre navegadores](#).

- Firefox utiliza la combinación Alt+Shift+tecla
- Google Chrome utilizan normalmente la combinación Alt+tecla, pero cuando esa combinación ya está asociada a alguna acción usa la combinación Alt+Shift+tecla

En el ejemplo siguiente Firefox utiliza la combinación Alt+Shift+tecla para situar el cursor en las cajas de texto, mientras que Google Chrome utiliza la combinación Alt+tecla:

```
<input type="text" name="texto1" placeholder="Acceso con a" accesskey="a"> <input
type="text" name="texto2" placeholder="Acceso con i" accesskey="i"> <input type="text"
name="texto3" placeholder="Acceso con o" accesskey="o"> <input type="text"
name="texto4" placeholder="Acceso con u" accesskey="u">
```

Chrome utiliza la combinación Alt+tecla, que tiene el inconveniente que entra en conflicto con combinaciones de teclas ya definidas por el navegador o el sistema operativo. Firefox utiliza la combinación Alt+Shift+tecla, más incómoda de utilizar, pero que no interfiere con las combinaciones ya definidas por el navegador o el sistema operativo.

En el ejemplo siguiente, Chrome no puede acceder a la caja de texto con la combinación Alt+d, ya que esta combinación traslada el foco a la barra de dirección, pero pueden hacerlo con la combinación Alt+Shift+d:

```
<input type="text" name="texto1" placeholder="Acceso con d" accesskey="d">
```

En el ejemplo siguiente, Chrome no puede acceder a la caja de texto con la combinación Alt+e, ya que esta combinación abre el menú de herramientas, pero puede hacerlo con la combinación Alt+Shift+e:

```
<input type="text" name="texto2" placeholder="Acceso con e" accesskey="e">
```

En el caso de las teclas de acceso que no son las letras del alfabeto (los acentos no pueden ser teclas de acceso):

- Firefox utiliza la combinación Alt+Shift+tecla
- Chrome utiliza la combinación Alt+tecla

```
<input type="text" name="texto1" placeholder="Acceso con 1" accesskey="1"> <input  
type="text" name="texto2" placeholder="Acceso con 2" accesskey="2"> <input type="text"  
name="texto3" placeholder="Acceso con 3" accesskey="3"> <input type="text"  
name="texto4" placeholder="Acceso con 4" accesskey="4">
```

```
<input type="text" name="texto1" placeholder="Acceso con ," accesskey=","> <input  
type="text" name="texto2" placeholder="Acceso con ." accesskey="."> <input type="text"  
name="texto3" placeholder="Acceso con +" accesskey="+">
```

Si hay varios elementos con la misma tecla de acceso, o si hay varios elementos con la misma tecla de acceso en mayúsculas y minúsculas:

- Google Chrome no avanza de uno a otro
- Firefox avanza de uno a otro con la combinación Alt+Shift+tecla

```
<input type="text" name="texto1" placeholder="Acceso con z" accesskey="z"> <input  
type="text" name="texto2" placeholder="Acceso con z" accesskey="z"> <input type="text"  
name="texto3" placeholder="Acceso con z" accesskey="z">
```

```
<input type="text" name="texto1" placeholder="Acceso con w" accesskey="w"> <input  
type="text" name="texto2" placeholder="Acceso con W" accesskey="W"> <input  
type="text" name="texto3" placeholder="Acceso con W" accesskey="W">
```