

Introducción a la Programación

Semana N° 4

Formulario HTML

Los formularios HTML son elementos de una página web que permiten a los usuarios introducir datos. Estos datos pueden ser enviados a un servidor web para su procesamiento, o pueden utilizarse para realizar acciones en el lado del cliente.

Los formularios HTML están formados por uno o más controles de formulario. Los controles de formulario son elementos que permiten a los usuarios introducir datos de diferentes tipos, como texto, números, fechas, etc.

Diseño de un formulario

Al diseñar un formulario HTML, hay una serie de factores a tener en cuenta:

La finalidad del formulario: ¿Qué datos desea recopilar del usuario? ¿Cómo se utilizarán estos datos?

El público objetivo: ¿Quién utilizará el formulario? ¿Qué nivel de alfabetización tecnológica tienen?

La usabilidad: El formulario debe ser fácil de usar y comprender.

Ejemplo de formulario

```
<form action="example.php" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" placeholder="Nombre">
  <label for="email">Correo electrónico:</label>
  <input type="email" name="email" id="email" placeholder="Correo electrónico">
  <label for="edad">Edad:</label>
  <input type="number" name="edad" id="edad" placeholder="Edad">
  <input type="submit" value="Enviar">
</form>
```

Etiqueta de formularios

La etiqueta <form>

La etiqueta <form> en HTML5 define un formulario web. Un formulario web es un elemento de una página web que permite a los usuarios introducir datos. Estos datos pueden ser enviados a un servidor web para su procesamiento, o pueden utilizarse para realizar acciones en el lado del cliente.

La etiqueta `<form>` tiene los siguientes atributos obligatorios:

`action`: Especifica la URL del archivo en el servidor web al que se enviarán los datos del formulario.

`method`: Especifica el método HTTP que se utilizará para enviar los datos del formulario. Los métodos HTTP más comunes son `post` y `get`.

La etiqueta `<form>` también tiene los siguientes atributos opcionales:

`enctype`: Especifica el tipo de codificación de datos que se utilizará para enviar los datos del formulario.

`name`: Especifica el nombre del formulario. Este nombre se utiliza para identificar el formulario en el código del servidor.

`autocomplete`: Especifica si el navegador debe rellenar automáticamente los campos del formulario.

La etiqueta `<label>`

La etiqueta `<label>` de HTML se utiliza para asociar una etiqueta con un elemento de formulario.

La etiqueta `<label>` se utiliza para proporcionar una etiqueta para un control de entrada, lo que ayuda a los usuarios a comprender la función del control de entrada.

La etiqueta `<label>` tiene los siguientes atributos obligatorios:

`for`: Especifica el id del elemento de formulario al que se asocia la etiqueta. La etiqueta `<label>` también tiene los siguientes atributos opcionales:

`accesskey`: Especifica una tecla de acceso para la etiqueta.

`tabindex`: Especifica el índice de tabulación de la etiqueta.

Para asociar una etiqueta con un control de entrada, colocamos la etiqueta `<label>` junto al control de entrada y especificamos el id del control de entrada en el atributo `for` de la etiqueta `<label>`.

Por ejemplo, el siguiente código crea un checkbox con el nombre "intereses" y el valor "deporte" y lo asocia con una etiqueta con el texto "Deportes":

```
<input type="checkbox" name="intereses" value="deporte" id="deportes">  
<label for="deportes">Deportes</label>
```

La etiqueta "Deportes" se muestra junto al checkbox. Cuando el usuario hace clic en el checkbox, la etiqueta "Deportes" también se activa.

El atributo `for` de la etiqueta `<label>` es obligatorio para que la etiqueta `<label>` funcione correctamente. Si no especificamos el atributo `for`, la etiqueta `<label>` no se asociará con ningún elemento de formulario.

La etiqueta `<label>` es una herramienta importante para mejorar la accesibilidad de los formularios web. Al asociar una etiqueta con un control de entrada, podemos proporcionar una etiqueta para el control de entrada, lo que ayuda a los usuarios a comprender la función del control de entrada.

La etiqueta `<input>`

La etiqueta `<input>` de HTML se utiliza para crear controles interactivos para formularios web. Los controles de entrada pueden ser campos de texto, botones de radio, casillas de verificación, listas desplegables, etc.

La etiqueta `<input>` tiene los siguientes atributos obligatorios:

`type`: Especifica el tipo de control de entrada. Los tipos de control de entrada más comunes son `text`, `password`, `radio`, `checkbox`, `submit`, `reset` y `file`. `name`: Especifica el nombre del control de entrada. Este nombre se utiliza para identificar el control de entrada en el código del servidor. La etiqueta `<input>` también tiene los siguientes atributos opcionales:

`value`: Especifica el valor inicial del control de entrada.

`checked`: Especifica si el control de entrada está marcado o no.

`disabled`: Especifica si el control de entrada está deshabilitado o no.

`readonly`: Especifica si el control de entrada es de solo lectura o no.

`size`: Especifica el tamaño del control de entrada.

`maxlength`: Especifica la longitud máxima del valor del control de entrada.

`placeholder`: Especifica un texto que se muestra en el control de entrada cuando está vacío.

`autocomplete`: Especifica si el navegador debe rellenar automáticamente el control de entrada.

`autofocus`: Especifica el control de entrada que debe tener el foco cuando la página se carga.

Además de estos atributos, la etiqueta `<input>` también admite una serie de atributos `aria` para mejorar la accesibilidad de los controles de entrada para las personas con discapacidades.

Ejemplos de la etiqueta `input`:

Campo de texto

Descripción:

El campo de texto permite al usuario introducir una línea de texto sin formato. Es ideal para nombres, direcciones, correos electrónicos y otros datos que no requieren formato adicional.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "text".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

placeholder: Muestra un texto dentro del campo que indica al usuario qué tipo de información debe introducir. En este caso, el placeholder es "Nombre".

label: Etiqueta el campo de texto y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

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

Atributos adicionales:

value: Establece un valor predeterminado para el campo de texto.

size: Define el ancho del campo de texto en caracteres.

maxlength: Limita la cantidad de caracteres que se pueden introducir en el campo.

required: Indica que el campo es obligatorio y debe ser completado antes de enviar el formulario.

readonly: Deshabilita la edición del campo de texto.

Contraseña

Descripción:

El campo de contraseña permite al usuario introducir una contraseña sin que se muestre en la pantalla. Es ideal para proteger información sensible como datos de acceso a cuentas, información financiera y otros datos confidenciales.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "password".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

placeholder: Muestra un texto dentro del campo que indica al usuario qué tipo de información debe introducir. En este caso, el placeholder es "Contraseña".

label: Etiqueta el campo de contraseña y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="password" name="contraseña" id="contraseña" placeholder="Contraseña">
<label for="contraseña">Contraseña</label>
```

Atributos adicionales:

value: No se recomienda establecer un valor predeterminado para un campo de contraseña.

size: Define el ancho del campo de contraseña en caracteres.

maxlength: Limita la cantidad de caracteres que se pueden introducir en el campo.

required: Indica que el campo es obligatorio y debe ser completado antes de enviar el formulario.

autocomplete: Indica si el navegador debe autocompletar el campo de contraseña. Se recomienda establecer este atributo como "off" para mejorar la seguridad.

Botones de radio

Descripción:

Los botones de radio permiten al usuario seleccionar una opción de un grupo de opciones mutuamente excluyentes. Es decir, solo se puede seleccionar un botón de radio dentro de un grupo con el mismo nombre. Son ideales para preguntas con respuestas de tipo sí/no, género, preferencias, etc.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "radio".

name: Especifica el nombre del grupo de botones de radio. Todos los botones de radio que deseen formar parte del mismo grupo deben tener el mismo nombre.

value: Especifica el valor que se envía al servidor cuando se selecciona el botón de radio.

checked: Indica si el botón de radio está seleccionado por defecto. Solo un botón de radio dentro de un grupo puede tener este atributo.

label: Etiqueta el botón de radio y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="radio" name="sexo" id="hombre" value="hombre">
<label for="hombre">Hombre</label>
<input type="radio" name="sexo" name="mujer" value="mujer">
<label for="mujer">Mujer</label>
```

Atributos adicionales:

disabled: Deshabilita el botón de radio, impidiendo que el usuario lo seleccione.

readonly: Hace que el botón de radio sea de solo lectura, pero el usuario aún puede verlo.

tabindex: Define el orden de tabulación del botón de radio.

Separar bloques de radio button HTML para que trabajen independientes

Existen dos métodos principales para separar bloques de radio button HTML y lograr que funcionen de forma independiente:

Agrupar los botones con fieldset y legend:

Agrupar los radio buttons relacionados dentro de un elemento fieldset. Esto define visualmente un bloque independiente.

Utiliza legend dentro de fieldset para proporcionar un título descriptivo al bloque.

Asigna el mismo valor al atributo name de cada radio button dentro del mismo bloque. Esto define un grupo de botones mutuamente excluyentes, donde solo se puede seleccionar uno a la vez.

Ejemplo:

```
<form action="/action_page.php">
  <fieldset>
    <legend>Elige tu género:</legend>
    <input type="radio" id="male" name="gender" value="male" checked>
    <label for="male">Hombre</label><br>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Mujer</label><br>
  </fieldset>

  <fieldset>
    <legend>Elige tu color favorito:</legend>
    <input type="radio" id="red" name="color" value="red">
    <label for="red">Rojo</label><br>
    <input type="radio" id="blue" name="color" value="blue">
    <label for="blue">Azul</label><br>
    <input type="radio" id="green" name="color" value="green">
    <label for="green">Verde</label><br>
  </fieldset>

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

En este ejemplo, se crean dos bloques independientes de radio buttons: uno para elegir género y otro para elegir color favorito. Cada bloque tiene su propio título descriptivo y los botones dentro de cada bloque comparten el mismo nombre de atributo, lo que asegura que solo se pueda seleccionar una opción por bloque.

Casilla de verificación

Descripción:

Las casillas de verificación permiten al usuario seleccionar una o más opciones de una lista. Son ideales para preguntas con respuestas múltiples, como preferencias, intereses, habilidades, etc.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "checkbox".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Especifica el valor que se envía al servidor cuando se selecciona la casilla de verificación.

checked: Indica si la casilla de verificación está seleccionada por defecto.

label: Etiqueta la casilla de verificación y la asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="checkbox" name="deporte" id="deporte" value="deporte">  
<label for="deporte">Deportes</label>  
<input type="checkbox" name="cine" id="cine" value="cine">  
<label for="cine">Cine</label>
```

Atributos adicionales:

disabled: Deshabilita la casilla de verificación, impidiendo que el usuario la seleccione.

readonly: Hace que la casilla de verificación sea de solo lectura, pero el usuario aún puede verla.

tabindex: Define el orden de tabulación de la casilla de verificación.

Botón de envío

Descripción:

El botón de envío permite al usuario enviar los datos del formulario al servidor. Es un elemento esencial en cualquier formulario web.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "submit".

value: Especifica el texto que se muestra en el botón.

label: Etiqueta el botón de envío y lo asocia con el nombre del campo. Aunque no es obligatorio, se recomienda para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="submit" value="Enviar" id="enviar">  
<label for="enviar">Enviar</label>
```

Atributos adicionales:

name: Especifica un nombre para el botón de envío. Se puede usar para identificar el botón en el código JavaScript.

disabled: Deshabilita el botón de envío, impidiendo que el usuario lo envíe.

form: Especifica el ID del formulario que se envía al hacer clic en el botón.

formaction: Especifica la URL a la que se envía el formulario.

formmethod: Especifica el método HTTP que se utiliza para enviar el formulario. Los valores posibles son "get" y "post".

formenctype: Especifica el tipo de codificación que se utiliza para enviar los datos del formulario. Los valores posibles son "application/x-www-form-urlencoded" y "multipart/form-data".

Botón de reinicio

Descripción:

El botón de reinicio permite al usuario borrar todos los datos introducidos en un formulario y volver a su estado inicial. Es útil para que los usuarios puedan corregir errores o empezar de nuevo sin tener que recargar la página.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "reset".

value: Especifica el texto que se muestra en el botón.

label: Etiqueta el botón de reinicio y lo asocia con el nombre del campo. Aunque no es obligatorio, se recomienda para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="reset" value="Restablecer" id="restablecer">  
<label for="restablecer">Restablecer</label>
```


Atributos adicionales:

name: Especifica un nombre para el botón de reinicio. Se puede usar para identificar el botón en el código JavaScript.

disabled: Deshabilita el botón de reinicio, impidiendo que el usuario lo use.

Botón de archivo

Descripción:

El botón de archivo permite al usuario seleccionar un archivo de su dispositivo para subirlo al servidor. Es ideal para cargar imágenes, documentos, videos y otros tipos de archivos.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "file".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

accept: Especifica los tipos de archivos que se pueden seleccionar. Se puede utilizar una lista de extensiones separadas por comas, como ".jpg,.png,.gif".

multiple: Permite al usuario seleccionar varios archivos al mismo tiempo.

size: Define el tamaño máximo del archivo que se puede subir.

label: Etiqueta el campo de archivo y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="file" name="archivo" id="archivo">  
<label for="archivo">Archivo</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de archivo, impidiendo que el usuario seleccione un archivo.

readonly: Hace que el campo de archivo sea de solo lectura, pero el usuario aún puede ver el archivo seleccionado.

tabindex: Define el orden de tabulación del campo de archivo.

Color

Descripción:

El campo de color permite al usuario seleccionar un color a través de un selector de color integrado en el navegador. Es ideal para situaciones en las que el usuario necesita elegir un color específico para personalizar una página web, un producto o cualquier elemento que requiera selección de color.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "color".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define el color inicial seleccionado en el selector. Se debe especificar en formato hexadecimal (por ejemplo, "#FF0000" para rojo).

label: Etiqueta el campo de color y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

```
<input type="color" name="color" id="color">  
<label for="color">Color</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de color, impidiendo que el usuario seleccione un color.

readonly: Hace que el campo de color sea de solo lectura, pero el usuario aún puede ver el color seleccionado.

tabindex: Define el orden de tabulación del campo de color.

Fecha

Descripción:

El campo de fecha permite al usuario seleccionar una fecha a través de un calendario integrado en el navegador. Es ideal para situaciones en las que el usuario necesita elegir una fecha específica, como para reservar una cita, registrar un evento o completar un formulario.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "date".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la fecha inicial seleccionada en el calendario. Se debe especificar en formato YYYY-MM-DD (por ejemplo, "2024-03-08" para hoy).

min: Define la fecha mínima que el usuario puede seleccionar. Se debe especificar en formato YYYY-MM-DD.

max: Define la fecha máxima que el usuario puede seleccionar. Se debe especificar en formato YYYY-MM-DD.

step: Define el intervalo de fechas que se puede seleccionar. Por ejemplo, un valor de "7" permite al usuario seleccionar solo fechas que sean múltiplos de 7 días a partir de la fecha mínima.

label: Etiqueta el campo de fecha y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="date" name="fecha" id="fecha">  
<label for="fecha">Fecha</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de fecha, impidiendo que el usuario seleccione una fecha.

readonly: Hace que el campo de fecha sea de solo lectura, pero el usuario aún puede ver la fecha seleccionada.

tabindex: Define el orden de tabulación del campo de fecha.

Hora

Descripción:

El campo de hora permite al usuario seleccionar una hora a través de un selector de hora integrado en el navegador. Es ideal para situaciones en las que el usuario necesita elegir una hora específica, como para programar una alarma, registrar un horario o completar un formulario.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "time".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la hora inicial seleccionada en el selector. Se debe especificar en formato HH:MM (por ejemplo, "12:00" para el mediodía).

min: Define la hora mínima que el usuario puede seleccionar. Se debe especificar en formato HH:MM.

max: Define la hora máxima que el usuario puede seleccionar. Se debe especificar en formato HH:MM.

step: Define el intervalo de tiempo que se puede seleccionar. Por ejemplo, un valor de "60" permite al usuario seleccionar solo horas que sean múltiplos de 60 minutos a partir de la hora mínima.

label: Etiqueta el campo de hora y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="time" name="hora" id="hora">  
<label for="hora">Hora</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de hora, impidiendo que el usuario seleccione una hora.

readonly: Hace que el campo de hora sea de solo lectura, pero el usuario aún puede ver la hora seleccionada.

tabindex: Define el orden de tabulación del campo de hora.

Fecha y hora local

Descripción:

El campo de fecha y hora local permite al usuario seleccionar una fecha y hora específica a través de un selector integrado en el navegador. Es ideal para situaciones en las que se necesita elegir una fecha y hora precisa, como para reservar un vuelo, registrar una entrada o completar un formulario.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "datetime-local".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la fecha y hora inicial seleccionada en el selector. Se debe especificar en formato YYYY-MM-DDTHH:MM (por ejemplo, "2024-03-08T12:00" para hoy a las 12:00 PM).

min: Define la fecha y hora mínima que el usuario puede seleccionar. Se debe especificar en formato YYYY-MM-DDTHH:MM.

max: Define la fecha y hora máxima que el usuario puede seleccionar. Se debe especificar en formato YYYY-MM-DDTHH:MM.

step: Define el intervalo de tiempo que se puede seleccionar. Por ejemplo, un valor de "60" permite al usuario seleccionar solo fechas y horas que sean múltiplos de 60 minutos a partir de la fecha y hora mínima.

label: Etiqueta el campo de fecha y hora local y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="datetime-local" name="fecha_hora" id="fecha_hora">  
<label for="fecha_hora">Fecha y hora</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de fecha y hora local, impidiendo que el usuario seleccione una fecha y hora.

readonly: Hace que el campo de fecha y hora local sea de solo lectura, pero el usuario aún puede ver la fecha y hora seleccionada.

tabindex: Define el orden de tabulación del campo de fecha y hora local.

Semana

Descripción:

El campo de semana permite al usuario seleccionar una semana específica del año a través de un selector integrado en el navegador. Es ideal para situaciones en las que se necesita elegir una semana para realizar un seguimiento de tareas, registrar eventos o completar un formulario.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "week".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la semana inicial seleccionada en el selector. Se debe especificar en formato YYYY-W## (por ejemplo, "2024-W10" para la décima semana de 2024).

min: Define la semana mínima que el usuario puede seleccionar. Se debe especificar en formato YYYY-W##.

max: Define la semana máxima que el usuario puede seleccionar. Se debe especificar en formato YYYY-W##.

step: Define el intervalo de semanas que se puede seleccionar. Por ejemplo, un valor de "1" permite al usuario seleccionar solo semanas que sean múltiplos de 1 semana a partir de la semana mínima.

label: Etiqueta el campo de semana y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="week" name="semana" id="semana">  
<label for="semana">Semana</label>
```

Atributos adicionales:

disabled: Deshabilita el campo de semana, impidiendo que el usuario seleccione una semana.

readonly: Hace que el campo de semana sea de solo lectura, pero el usuario aún puede ver la semana seleccionada.

tabindex: Define el orden de tabulación del campo de semana.

Teléfono

Descripción:

El campo de teléfono permite al usuario ingresar un número de teléfono. Es ideal para situaciones en las que se necesita recopilar información de contacto, como en formularios de registro, reservas o encuestas.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "tel".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define el número de teléfono inicial que se muestra en el campo.

pattern: Define una expresión regular que el número de teléfono debe cumplir. Puedes usar esta función para validar el formato del número, por ejemplo, para asegurarte de que solo se introduzcan números y ciertos caracteres especiales.

placeholder: Define un texto que se muestra en el campo como indicación para el usuario.

required: Indica que el campo es obligatorio y el usuario debe ingresar un número de teléfono.

disabled: Deshabilita el campo de teléfono, impidiendo que el usuario ingrese un número.

readonly: Hace que el campo de teléfono sea de solo lectura, pero el usuario aún puede ver el número de teléfono ingresado.

maxlength: Define el número máximo de caracteres que el usuario puede ingresar en el campo.

minlength: Define el número mínimo de caracteres que el usuario puede ingresar en el campo.

size: Define el tamaño del campo de entrada en caracteres.

label: Etiqueta el campo de teléfono y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="tel" name="teléfono" id="teléfono">  
<label for="teléfono">Teléfono</label>
```

Ejemplo de uso de pattern:

```
<input type="tel" name="teléfono" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
```

En este ejemplo, el número de teléfono debe tener el formato ###-###-####, donde # representa un dígito numérico.

Ejemplo de uso de placeholder:

```
<input type="tel" name="teléfono" placeholder="Ingrese su número de teléfono">
```

En este ejemplo, se muestra un texto informativo dentro del campo de teléfono que indica al usuario qué tipo de información debe ingresar.

Ejemplo de uso de required:

```
<input type="tel" name="teléfono" required>
```

En este ejemplo, el usuario no podrá enviar el formulario sin ingresar un número de teléfono en el campo.

URL

Descripción:

El campo de URL permite al usuario ingresar una dirección web. Es ideal para situaciones en las que se necesita recopilar información sobre sitios web, como en formularios de registro, comentarios o encuestas.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "url".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la URL inicial que se muestra en el campo.

pattern: Define una expresión regular que la URL debe cumplir. Puedes usar esta función para validar el formato de la URL, por ejemplo, para asegurarte de que comienza con "http://" o "https://".

placeholder: Define un texto que se muestra en el campo como indicación para el usuario.

required: Indica que el campo es obligatorio y el usuario debe ingresar una URL.

disabled: Deshabilita el campo de URL, impidiendo que el usuario ingrese una URL.

readonly: Hace que el campo de URL sea de solo lectura, pero el usuario aún puede ver la URL ingresada.

size: Define el tamaño del campo de entrada en caracteres.

label: Etiqueta el campo de URL y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="url" name="sitio_web" id="sitio_web">  
<label for="sitio_web">Sitio web</label>
```

Email

Descripción:

El campo de email permite al usuario ingresar una dirección de correo electrónico. Es ideal para situaciones en las que se necesita recopilar información de contacto, como en formularios de registro, suscripciones o boletines informativos.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "email".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define la dirección de correo electrónico inicial que se muestra en el campo.

pattern: Define una expresión regular que la dirección de correo electrónico debe cumplir. Puedes usar esta función para validar el formato del email, por ejemplo, para asegurarte de que contiene un nombre de usuario, un símbolo "@", un nombre de dominio y un sufijo.

placeholder: Define un texto que se muestra en el campo como indicación para el usuario.

required: Indica que el campo es obligatorio y el usuario debe ingresar una dirección de correo electrónico.

disabled: Deshabilita el campo de email, impidiendo que el usuario ingrese una dirección de correo electrónico.

readonly: Hace que el campo de email sea de solo lectura, pero el usuario aún puede ver la dirección de correo electrónico ingresada.

size: Define el tamaño del campo de entrada en caracteres.

multiple: Permite al usuario ingresar varias direcciones de correo electrónico separadas por comas.

label: Etiqueta el campo de email y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="email" name="correo_electrónico" id="correo_electrónico">  
<label for="correo_electrónico">Correo electrónico</label>
```


Número

Descripción:

El campo de número permite al usuario ingresar un valor numérico. Es ideal para situaciones en las que se necesita recopilar información numérica, como en formularios de registro, encuestas o pedidos.

Atributos:

type: Indica el tipo de campo de entrada. En este caso, se establece como "number".

name: Especifica el nombre del campo. Se utiliza para identificar el valor del campo cuando se envía el formulario.

value: Define el valor numérico inicial que se muestra en el campo.

min: Define el valor mínimo que el usuario puede ingresar.

max: Define el valor máximo que el usuario puede ingresar.

step: Define el intervalo de valores que se puede seleccionar. Por ejemplo, un valor de "1" permite al usuario seleccionar solo números enteros, mientras que un valor de "0.1" permite seleccionar números decimales con una precisión de una décima.

placeholder: Define un texto que se muestra en el campo como indicación para el usuario.

required: Indica que el campo es obligatorio y el usuario debe ingresar un número.

disabled: Deshabilita el campo de número, impidiendo que el usuario ingrese un número.

readonly: Hace que el campo de número sea de solo lectura, pero el usuario aún puede ver el número ingresado.

size: Define el tamaño del campo de entrada en caracteres.

label: Etiqueta el campo de número y lo asocia con el nombre del campo. Es importante para la accesibilidad y la usabilidad.

Ejemplo:

```
<input type="number" name="edad" id="edad">  
<label for="edad">Edad</label>
```

Ejemplo de uso de min y max:

```
<input type="number" name="edad" min="18" max="100">
```

En este ejemplo, el usuario solo podrá ingresar un valor entre 18 y 100 en el campo de edad.

Ejemplo de uso de step:

```
<input type="number" name="peso" step="0.1">
```

En este ejemplo, el usuario podrá ingresar valores decimales con una precisión de una décima en el campo de peso.

Ejemplo de uso de placeholder:

```
<input type="number" name="cantidad" placeholder="Ingrese la cantidad">
```

En este ejemplo, se muestra un texto informativo dentro del campo de número que indica al usuario qué tipo de información debe ingresar.

Ejemplo de uso de required:

```
<input type="number" name="código_postal" required>
```

En este ejemplo, el usuario no podrá enviar el formulario sin ingresar un número en el campo de código postal.

Ejemplos de tipos de etiqueta input

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

<head>
  <meta charset="UTF-8">
  <title>Ejemplos de la etiqueta input HTML5</title>
</head>

<body>
  <h1>Ejemplos de la etiqueta input HTML5</h1>

  <!-- Ejemplo de campo de texto con placeholder -->
  <div>
    <label for="nombre">Nombre</label>
    <input type="text" name="nombre" placeholder="Nombre" id="nombre">
  </div>

  <!-- Ejemplo de campo de contraseña con placeholder -->
  <div>
    <label for="contraseña">Contraseña</label>
    <input type="password" name="contraseña" placeholder="Contraseña" id="contraseña">
  </div>
```

```
<!-- Ejemplo de botones de radio -->
<div>
  <input type="radio" name="sexo" value="hombre" id="hombre">
  <label for="hombre">Hombre</label>
  <input type="radio" name="sexo" value="mujer" id="mujer">
  <label for="mujer">Mujer</label>
</div>

<!-- Ejemplo de casillas de verificación -->
<div>
  <input type="checkbox" name="deporte" value="deporte" id="deporte">
  <label for="deporte">Deportes</label>
  <input type="checkbox" name="cine" value="cine" id="cine">
  <label for="cine">Cine</label>
</div>

<!-- Ejemplo de botón de envío -->
<div>
  <input type="submit" value="Enviar" id="enviar">
</div>

<!-- Ejemplo de control de entrada de archivo con placeholder -->
<div>
  <input type="file" name="archivo" placeholder="Archivo" id="archivo">
</div>

<!-- Ejemplo de control de entrada de color -->
<div>
  <input type="color" name="color" id="color">
  <label for="color">Color</label>
</div>

<!-- Ejemplo de control de entrada de fecha -->
<div>
  <input type="date" name="fecha" id="fecha">
  <label for="fecha">Fecha</label>
</div>

<!-- Ejemplo de control de entrada de hora -->
<div>
  <input type="time" name="hora" id="hora">
  <label for="hora">Hora</label>
</div>
```

```

<!-- Ejemplo de control de entrada de semana -->
<div>
  <input type="week" name="semana" id="semana">
</div>

<!-- Ejemplo de control de entrada de teléfono con placeholder -->
<div>
  <input type="tel" name="teléfono" placeholder="Teléfono" id="teléfono">
  <label for="teléfono">Teléfono</label>
</div>

<!-- Ejemplo de control de entrada de URL con placeholder -->
<div>
  <input type="url" name="sitio_web" placeholder="Sitio web" id="sitio_web">
  <label for="sitio_web">Sitio web</label>
</div>

<!-- Ejemplo de control de entrada de número con placeholder -->
<div>
  <input type="number" name="edad" placeholder="Edad" id="edad">
  <label for="edad">Edad</label>
</div>

</body>
</html>

```

La etiqueta <textarea>

La etiqueta <textarea> HTML5 se utiliza para crear un control de entrada de texto de varias líneas. Los textarea input se utilizan a menudo para recopilar comentarios, reseñas, o otro tipo de texto largo.

La etiqueta textarea tiene los siguientes atributos:

name: Especifica el nombre del control de entrada. El nombre del control de entrada se utiliza para identificar el control de entrada en el formulario.

id: Especifica el ID del control de entrada. El ID del control de entrada se utiliza para identificar el control de entrada en el código JavaScript.

placeholder: Especifica un texto predeterminado que se muestra en el control de entrada cuando está vacío.

rows: Especifica el número de filas del textarea input.

cols: Especifica el número de columnas del textarea input.

maxlength: Especifica la longitud máxima del texto que se puede ingresar en el textarea input.

minlength: Especifica la longitud mínima del texto que se puede ingresar en el textarea input.

required: Especifica si el textarea input es obligatorio.

disabled: Especifica si el textarea input está deshabilitado.

readonly: Especifica si el textarea input está bloqueado para escritura.

El siguiente código muestra un ejemplo de textarea input HTML5:

```
<textarea name="comentarios" id="comentarios" placeholder="Ingrese sus comentarios aquí" rows="10" cols="50"></textarea>
```

Este código creará un textarea input con 10 filas y 50 columnas. El usuario podrá ingresar hasta 500 caracteres en el textarea input.

Los textarea input HTML5 son una herramienta útil para recopilar texto largo de los usuarios.

La etiqueta <select>

La etiqueta <select> HTML5 se utiliza para crear un menú desplegable. Los menús desplegables se utilizan a menudo para recopilar una selección de opciones de los usuarios.

La etiqueta <select> tiene los siguientes atributos:

name: Especifica el nombre del control de entrada. El nombre del control de entrada se utiliza para identificar el control de entrada en el formulario.

id: Especifica el ID del control de entrada. El ID del control de entrada se utiliza para identificar el control de entrada en el código JavaScript.

multiple: Especifica si el usuario puede seleccionar más de una opción del menú desplegable.

size: Especifica el número de opciones que se muestran en el menú desplegable de forma predeterminada.

required: Especifica si el menú desplegable es obligatorio.

disabled: Especifica si el menú desplegable está deshabilitado.

readonly: Especifica si el menú desplegable está bloqueado para escritura.

El siguiente código muestra un ejemplo de select input HTML5:

```
<select name="color" id="color">  
  <option value="rojo">Rojo</option>  
  <option value="verde">Verde</option>  
  <option value="azul">Azul</option>  
</select>
```

Este código creará un menú desplegable con tres opciones: rojo, verde y azul. El usuario puede seleccionar una opción del menú desplegable.

También puede utilizar el atributo `multiple` para permitir que el usuario seleccione más de una opción del menú desplegable. Por ejemplo, el siguiente código creará un menú desplegable con tres opciones, y el usuario podrá seleccionar todas las opciones que desee:

```
<select name="colores" id="colores" multiple>
  <option value="rojo">Rojo</option>
  <option value="verde">Verde</option>
  <option value="azul">Azul</option>
</select>
```

Los menús desplegables HTML5 son una herramienta útil para recopilar una selección de opciones de los usuarios.

La etiqueta `<option>`

La etiqueta `<option>` HTML5 se utiliza para crear una opción en un menú desplegable. Las opciones se utilizan para recopilar una selección de opciones de los usuarios.

La etiqueta `<option>` tiene los siguientes atributos:

`value`: Especifica el valor de la opción. El valor de la opción se envía al servidor cuando el usuario selecciona la opción.

`label`: Especifica el texto que se muestra en la opción.

`selected`: Especifica si la opción está seleccionada de forma predeterminada.

`disabled`: Especifica si la opción está deshabilitada.

`readonly`: Especifica si la opción está bloqueada para escritura.

El siguiente código muestra un ejemplo de etiqueta `option` HTML5:

```
<option value="rojo">Rojo</option>
<option value="verde">Verde</option>
<option value="azul">Azul</option>
```

Este código creará tres opciones de menú desplegable: rojo, verde y azul. El atributo `value` especifica el valor de cada opción. El atributo `label` especifica el texto que se muestra en cada opción.

También puede utilizar el atributo `selected` para seleccionar una opción de forma predeterminada. Por ejemplo, el siguiente código seleccionará la opción rojo de forma predeterminada:

```
<option value="rojo" selected>Rojo</option>
<option value="verde">Verde</option>
<option value="azul">Azul</option>
```

Las etiquetas option HTML5 son una herramienta útil para recopilar una selección de opciones de los usuarios.

La etiqueta <button>

La etiqueta <button> HTML5 se utiliza para crear un botón. Los botones se utilizan para realizar acciones, como enviar un formulario, abrir una nueva pestaña o ventana, o iniciar un proceso.

La etiqueta <button> tiene los siguientes atributos:

type: Especifica el tipo de botón. Los tipos de botón comunes incluyen submit, reset, y button.

name: Especifica el nombre del botón. El nombre del botón se utiliza para identificar el botón en el formulario.

id: Especifica el ID del botón. El ID del botón se utiliza para identificar el botón en el código JavaScript.

value: Especifica el valor del botón. El valor del botón se envía al servidor cuando el usuario hace clic en el botón.

disabled: Especifica si el botón está deshabilitado.

El siguiente código muestra un ejemplo de etiqueta <button> HTML5:

```
<button type="submit" value="Enviar">Enviar</button>
```

Este código creará un botón que dice "Enviar". El botón es del tipo submit, lo que significa que enviará el formulario al servidor cuando el usuario haga clic en él. El valor del botón es "Enviar", que se enviará al servidor cuando el usuario haga clic en el botón.

También puede utilizar el atributo type para especificar el tipo de botón. Los tipos de botón comunes incluyen:

submit: Este tipo de botón envía el formulario al servidor cuando el usuario hace clic en él.

reset: Este tipo de botón restablece el formulario a su estado inicial cuando el usuario hace clic en él.

button: Este tipo de botón no tiene una función predefinida. El botón se puede utilizar para realizar cualquier acción deseada.

Los botones HTML5 son una herramienta útil para realizar acciones en una página web.