



DISEÑO WEB

Guía 4

DAVID VALENCIA SANDOVAL
INGENIERO EN SISTEMAS
david.valencia@ciaf.edu.co

In the bottom left corner, there is a large blue circle with a smaller, lighter blue circle partially overlapping it. In the bottom right corner, there is a large, light blue curved shape with a small blue circle attached to its inner curve.

¿Qué es un formulario?

Se conoce como formulario a los mecanismos para enviar información por parte del usuario, a través de unos campos visuales de forma sencilla e intuitiva, que determinan la naturaleza del tipo de información que se va a enviar y como hacerla llegar a otro extremo donde se procesará esa información.

Nuestros objetivos al crear un formulario HTML deben ser los siguientes:

Hacer lo más sencillo posible el proceso de inserción de datos por parte del usuario.

Intentar que la experiencia de usuario sea lo más agradable y cómoda posible.

Intentar que los datos introducidos estén en un formato predecible y esperado.

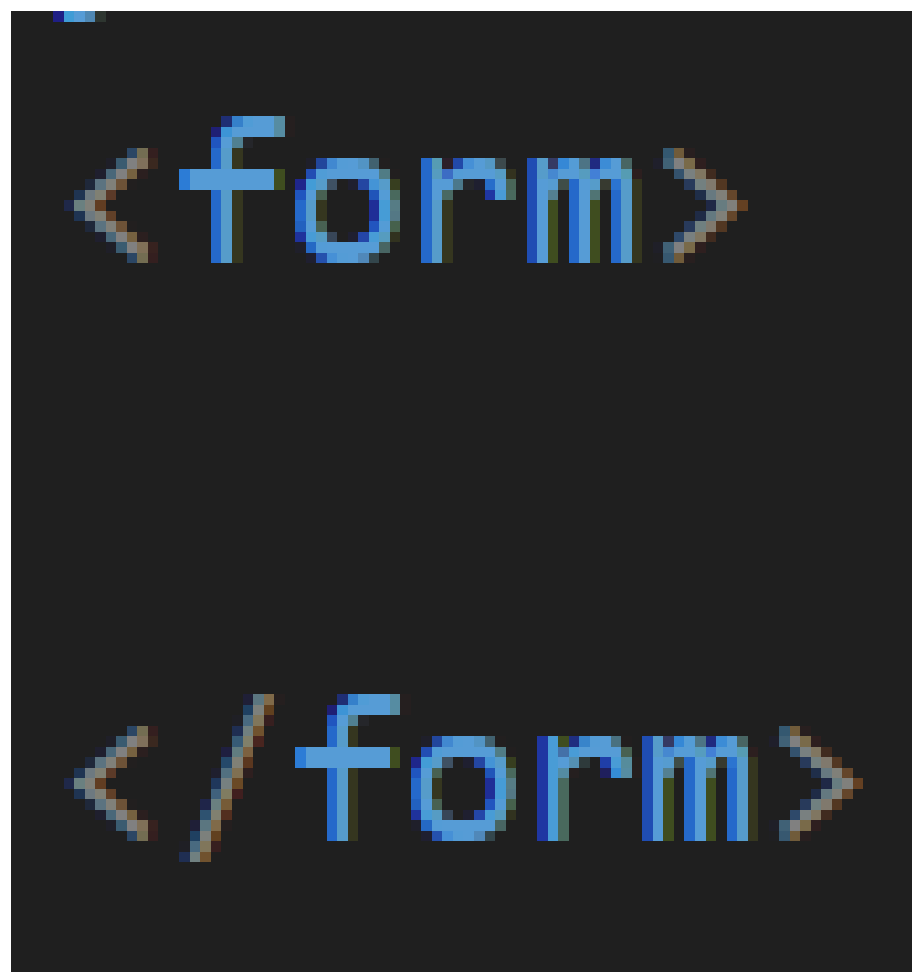
Reducir los errores al introducir datos en la medida de lo posible

Comunicar clara y rápidamente si ocurren errores al introducir datos.

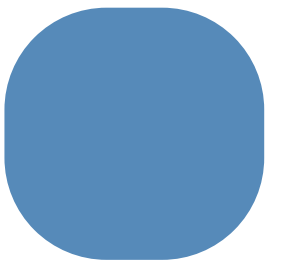


La etiqueta <form>

Para comenzar con la creación de un formulario, el primer paso es indicar una etiqueta contenedora <form>, que incluirá toda la información que se quiere recoger en ese formulario:



De esta forma, incluso se pueden crear varias etiquetas <form> por página, pudiendo así crear varios formularios con diferentes objetivos, como por ejemplo, uno para realizar una búsqueda en la página, otro para dejar un comentario y otro para enviar un email al autor, etc.



Atributo	Descripción	Más información
action	Dirección URL del backend al cuál se enviará la información obtenida en el formulario.	
method	Método HTTP de envío. <code>GET</code> a través de URL, <code>POST</code> para envío extenso de información.	
name	Nombre del formulario. Útil para identificar el formulario y procesar posteriormente.	
target	Nombre del lugar donde se abrirá el formulario. <code>_blank</code> para nueva pestaña.	
enctype	Codificación para el envío del formulario. Importante para envío de archivos.	Envío de archivos
accept-charset	Fuerza a utilizar una codificación en los parámetros de texto del formulario.	
autocomplete	Activa <code>on</code> o desactiva <code>off</code> el autocompletado para todos los campos del formulario.	
novalidate	BOOLEAN Desactiva la validación HTML5 del formulario.	

Ejemplo básico 1

Así pues, vamos a ver un primer formulario base de ejemplo (aún vacío, sin campos de entrada de datos, sólo con la etiqueta contenedora) para ejemplificar su utilización:

```
<form name="formulario" method="post" action="/send/">  
  
</form>
```

Nótese que en este primer ejemplo se definen 3 atributos base en <form>:

- El atributo **name** define el nombre del formulario para diferenciarlo de otros.
- El atributo **method** define el método HTTP a utilizar. Usaremos GET cuando nos interese enviar datos a través de la URL. En caso contrario, POST.
- El atributo **action** se usa para indicar la URL a donde se enviarán los datos cuando el usuario pulse el botón de enviar. En caso de omisión, se enviarán a la propia página actual, volviéndola a cargar. En este caso, se estarían enviando a la ruta absoluta <https://dominio.com/send/>.

Ejemplo básico 2

```
<form method="get" action="/search/" target="_blank" accept-charset="utf-8">  
  
</form>
```

En este caso enviamos a través de método GET a la URL <https://dominio.com/search/>:

- Se ha indicado el atributo **target** con valor `_blank`, esto significa que se abrirá en una nueva pestaña.
- Se ha indicado el atributo **accept-charset**. Esto forzará a que el texto enviado en el formulario sea interpretado como codificación UTF8. Útil si tienes problemas de caracteres extraños o similar.

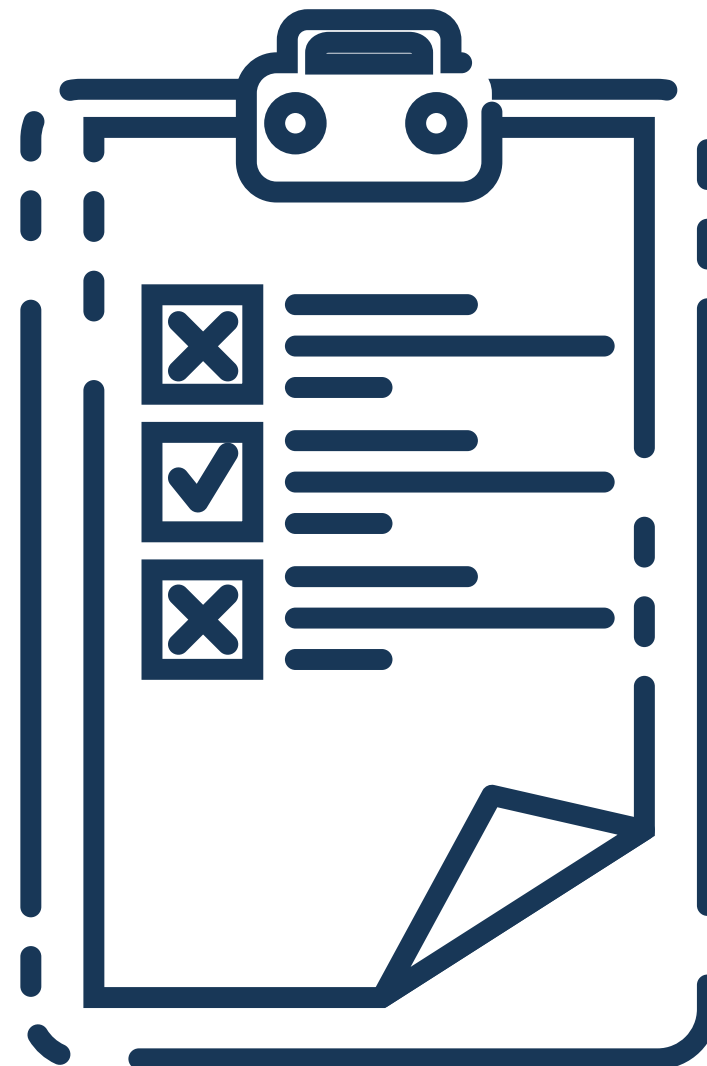
Ejemplo básico 3

```
<form method="post" action="/saveform/" autocomplete="off" novalidate>  
  
</form>
```

- El atributo **autocomplete** indica el valor off. Por defecto, este atributo está a on. Desactivado, todos los campos del formulario no mostrarán sugerencias anteriores de autocompletado.
- El atributo **novalidate** hará que el usuario pueda saltarse todas las validaciones HTML del mismo. Lo veremos más adelante en la sección de validaciones HTML.

Información de un formulario

Para que un usuario pueda introducir información en un formulario, requiere que se le indiquen una serie de campos que le permitan introducir la información de la forma más cómoda y rápida posible. Para ello, debemos saber a priori que tipo de dato le vamos a pedir al usuario (texto, cantidades numéricas, fechas, etc...), y así saber que tipo de campo de entrada de datos es más conveniente utilizar.



La etiqueta <input>

La etiqueta <input> tiene una gran cantidad de atributos (algunos dependen del valor del atributo type).

Veamos los atributos que podemos utilizar de forma general para prácticamente cualquier campo de entrada de datos:

Atributo	Descripción
name	Indica el nombre del campo. Útil para hacer referencia a la información almacenada.
type	Indica la naturaleza de datos que se espera del usuario (texto, número, fechas, etc...).
value	Indica el valor inicial que tendrá este campo de datos. Si se omite, está vacío.
form	Asocia este campo con un formulario específico. Útil si necesitamos tenerlo fuera del <form>.
list	En algunos <input> permite combinar con otra etiqueta para personalizarla. Lo veremos más adelante.
placeholder	Indica una sugerencia al usuario antes de escribir el valor en el campo.
size	Tamaño visual (aprox: número de caracteres) del campo de datos.
autocomplete	Activa on o desactiva off el autocompletado para este campo. Activado por defecto.
autofocus	Establece el foco (coloca el cursor) en este campo al cargar la página.

La etiqueta <input> - El atributo value

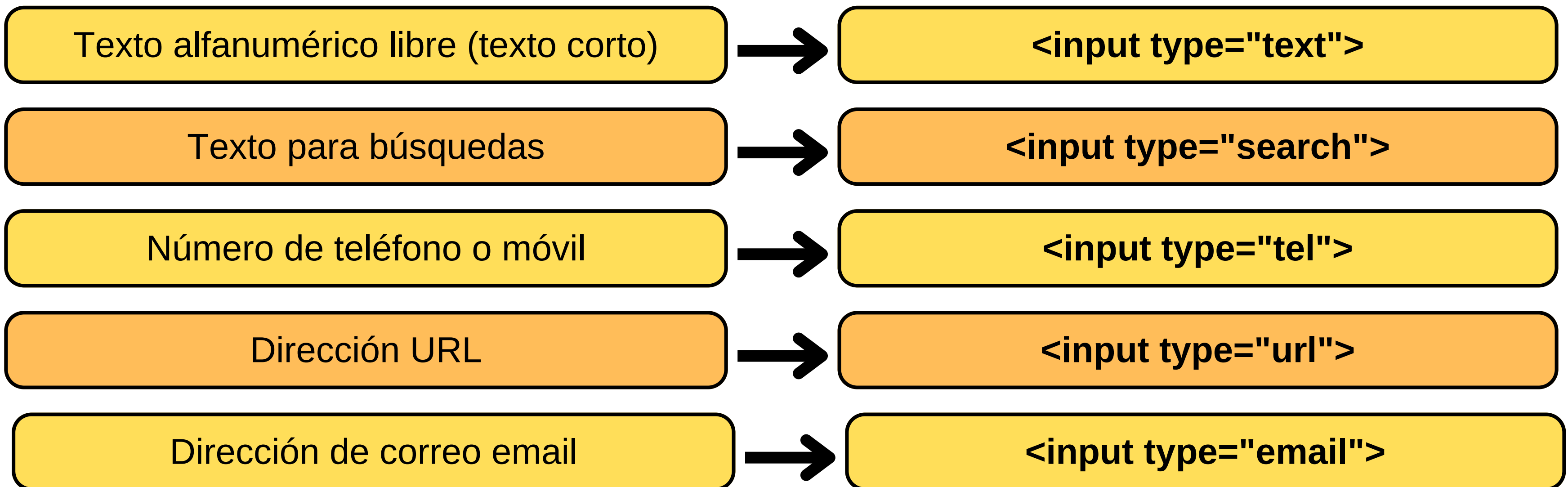
El atributo **value** se utiliza para dar un valor por defecto a un campo <input>. Tanto en este tipo de campos de entrada como en otros que veremos más adelante, podemos utilizar el atributo value para establecer un valor por defecto:

```
<form method="post" action="/send/">  
  <input value="Talento Tech">  
</form>
```

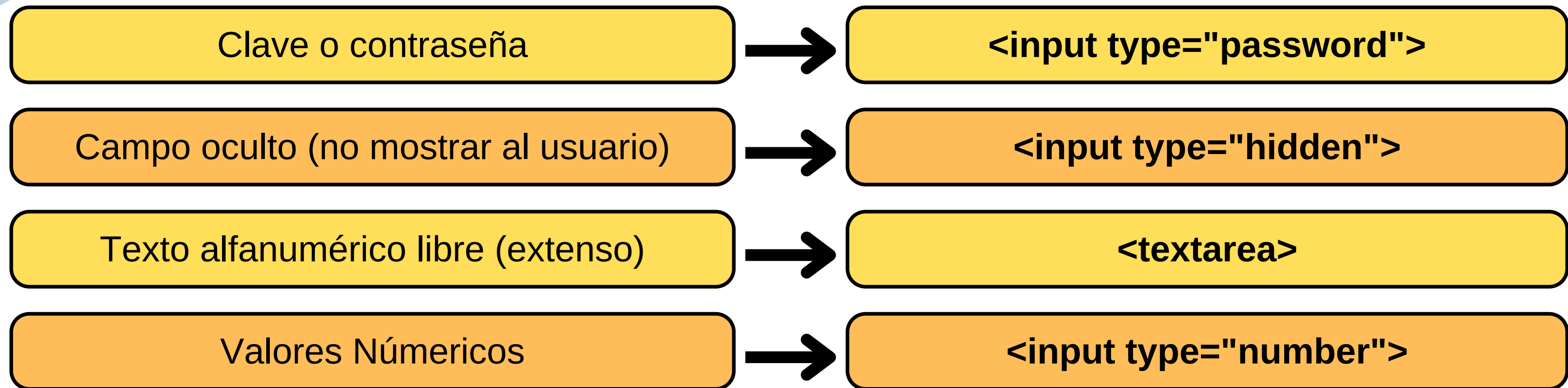
Esto puede ser realmente útil cuando un campo, probablemente, vaya a tener un valor específico. Podemos indicárselo para ahorrar tiempo al usuario, ya que en el caso de querer otra opción, sólo tiene que borrar y corregirla.

La etiqueta `<input>` - El atributo `type`

Para determinar la naturaleza de la información que se pretende obtener en cada campo de entrada de datos, se aconseja utilizar el atributo `type`, donde podremos indicar si los datos que esperamos por parte del usuario serán datos de texto, datos numéricos, fechas, etc. Si se omite el atributo `type`, se asume que es `type="text"`.



La etiqueta <input> - El atributo type



Como puedes ver, son múltiples tipos de campos de datos y muy variados, y aunque en principio no parecen tener diferencias, hay algunos matices que iremos explicando poco a poco. La etiqueta `<textarea>` también la explicaremos más adelante.

La etiqueta <input> - El atributo type

```
<form method="post" action="/send/">
  <input name="nombre_cliente" type="text" placeholder="Ingrese Su nombre">
  <input name="documento_cliente" type="text" placeholder="Ingrese Su documento">
  <input name="telefono_cliente" type="tel" placeholder="Ingrese Su Teléfono">
  <input name="correo_cliente" type="email" placeholder="Ingrese Su Correo">
  <input name="contrasena_cliente" type="password" placeholder="Ingrese Su Contraseña">
  <input name="tipo_cliente" type="text" value="CLIENTE">
</form>
```

La etiqueta <input> - El atributo placeholder

Placeholder: Es el texto que aparecerá a modo de sugerencia/ayuda para el usuario (en gris, ligeramente menos visible que el texto del campo). En cuanto el usuario escriba en el mismo, el mensaje desaparecerá. Sólo volvería a aparecer si el usuario decide borrar nuevamente la información que ha escrito en ese campo.

La etiqueta <input> - Comprobación de ortografía

En todos estos campos, tanto etiquetas <input> como las etiquetas <textarea> que veremos más adelante, podemos utilizar un atributo denominado spellcheck mediante el cuál indicamos al navegador si debe revisar o no la ortografía del texto escrito en dicho elemento (en el cuál aparece una línea roja inferior cuando la palabra no se encuentra en el diccionario):

```
<input name="nombre_cliente" type="text" spellcheck="true" placeholder="Ingrese Su nombre">
```

La etiqueta <textarea>

Esta etiqueta tiene algunos atributos que podría ser interesante destacar:

Atributo	Descripción
cols	Número de caracteres que caben en horizontal (columnas).
rows	Número de caracteres que caben en vertical (filas).
wrap	Con <code>soft</code> (por defecto) o <code>hard</code> puedes indicar si el texto debe enviarse ajustado o tal cuál se escribió.

La etiqueta <textarea>

```
<form method="post" action="/send/">  
  <textarea name="text" cols="32" rows="8">  
    Este es el valor de texto por defecto</textarea>  
</form>
```

Observa que en este caso, la etiqueta <textarea> tiene un valor por defecto, pero no lo hemos indicado en el atributo value como hacíamos en la etiqueta <input>, sino que aquí lo incluimos en el interior de la etiqueta <textarea>, ya que esta etiqueta si tiene etiqueta de cierre.

La etiqueta <textarea> Tamaño del <textarea>

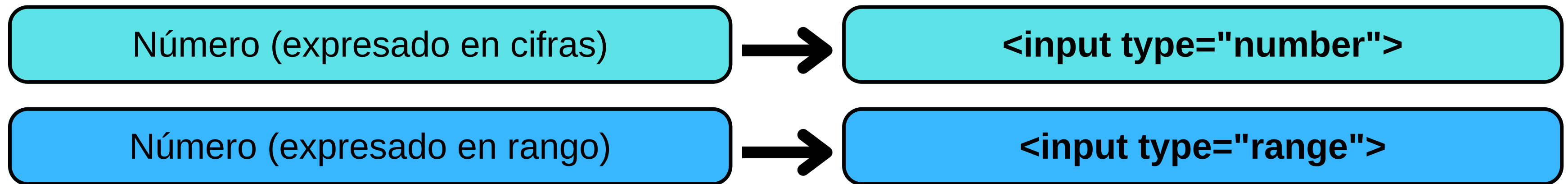
Si queremos modificar las dimensiones del elemento <textarea> utilizando única y exclusivamente HTML, podemos utilizar los atributos cols y rows. Estos atributos nos permiten aumentar el tamaño de ancho y alto del área de texto. Los valores numéricos de estos atributos equivalen al tamaño aproximado de un carácter en este campo, de modo que si indicamos cols="20" significa que habrá un espacio de ancho de aproximadamente 20 caracteres.

Veamos un ejemplo básico con una etiqueta HTML <textarea>:

```
<form method="post" action="/send/">  
  <textarea name="text" cols="32" rows="10"></textarea>  
</form>
```

El atributo type="number"

Para permitir al usuario introducir números, lo podemos hacer de dos formas, utilizando la etiqueta `<input>` con diferentes valores en el atributo `type`. Son las dos siguientes:



`<input type="number">`, una etiqueta mediante la cuál el usuario puede indicar la cifra numérica que quiere enviar en el formulario. La diferencia principal respecto a los campos de texto es que:

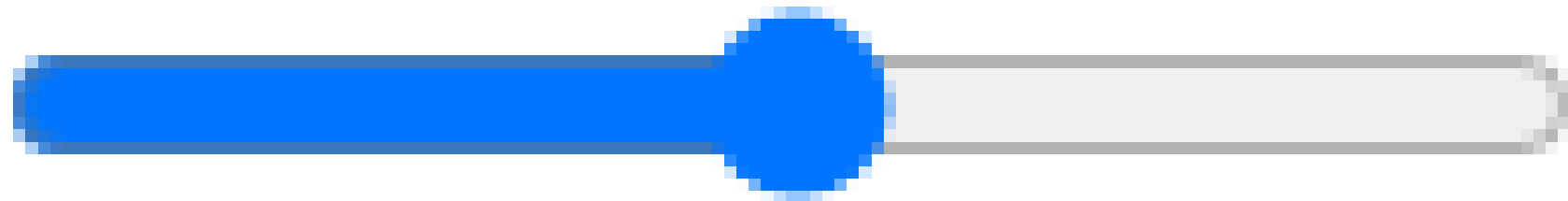
- Al escribir la cifra, nos aparecen flechitas de aumentar y disminuir la cantidad.
- Sólo se permite escribir números y caracteres permitidos.

El atributo type="number"

Carácteres	Descripción
Dígitos 0-9	Para indicar las diferentes cifras.
Signo +	Para indicar números positivos.
Signo -	Para indicar números negativos.
Número e	Para indicar números relacionados con el número e.
Signos . o ,	Para indicar la parte decimal.

El atributo type="range"

Por otro lado, tenemos la etiqueta `<input type="range">`, que en lugar de mostrarnos un campo de texto donde podemos insertar los números directamente, podemos hacerlo mediante un «slider» o barra de desplazamiento que representa un número entre un rango numérico específico. Funciona exactamente igual que `<input type="number">`, sólo que se muestra mediante un slider.



Sin embargo, este método tiene dos inconvenientes:

- Por defecto, no se muestra el valor elegido.
- Por defecto, los valores mínimo y máximo son 0 y 100.

```
<form method="post" action="/send/">
  <input type="range" min="10000" max="50000000" step="100000"
  | onInput="this.nextElementSibling.value = this.value">
  <output>0</output>
</form>
```

El atributo type="date"

Si nuestra intención es que el usuario introduzca una fecha concreta, en lugar de utilizar un campo de texto genérico `<input>`, lo ideal sería utilizar uno relacionado con las fechas, que desplegará un control denominado frecuentemente date picker. Básicamente, se trata de un campo especial donde el usuario puede seleccionar el día, mes y año que desee, simplemente a golpe de clic, mostrando un calendario adaptado y ahorrándose el escribir la fecha, además de simplificar el formato a utilizar, ya que puede ser muy complicado guardar siempre la fecha en un mismo formato.

Fecha

`<input type="date">`

Mes

`<input type="month">`

Semana

`<input type="week">`

Hora

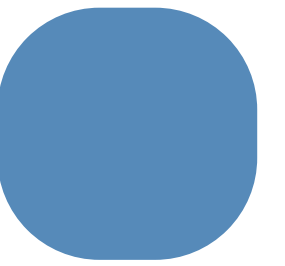
`<input type="time">`

Fecha y hora local

`<input type="datetime-local">`

El atributo type="date"

```
<form method="post" action="/send/">  
  <input type="date" name="fecha_nacimiento"/>  
  <input type="time" name="hora_entrada"/>  
  <input type="week" name="semana_trabajada"/>  
</form>
```



Casillas de verificación

Si queremos definir opciones donde el usuario debe elegir o escoger, en muchos casos lo más apropiado suele ser utilizar uno de los dos tipos de campos siguientes:

- Casillas de verificación, también denominados campos checkbox (verdadero o falso).
- Botones de opción, también denominados botones radio (una opción entre varias posibles).

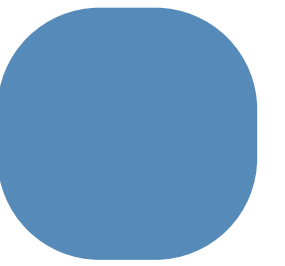
Casilla de verificación (on/off)

`<input type="checkbox">`

Botón radio (opción única)



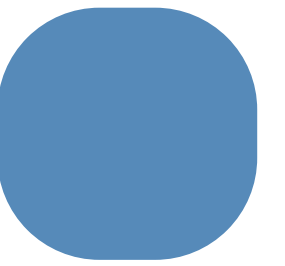
`<input type="radio">`



Casillas de verificación

Las casillas de verificación se pueden indicar utilizando el atributo `type="checkbox"` en un campo de entrada de datos `<input>`. Este tipo de control permite mostrar una casilla al usuario, dándole la posibilidad de marcarlo como activado o desactivado.

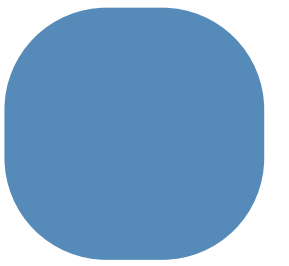
```
<form method="post" action="/send/">  
  <input type="checkbox" name="nombre_usuario">  
  Mostrar nombre real de usuario  
</form>
```



Casillas de verificación El atributo checked

Si indicamos el atributo checked, haremos que por defecto la casilla de verificación esté marcado por defecto, es decir, inicialmente al cargar la página:

```
<form method="post" action="/send/">
  <input type="checkbox" name="nombre_usuario" checked>
    Mostrar nombre real de usuario
  <input type="checkbox" name="nombre_usuario_nick">
    Mostrar nick de usuario
</form>
```



Botones de radio

Mediante la etiqueta `<input type="radio">` podemos crear los denominados botones radio. Este tipo de control se suele utilizar cuando el usuario debe elegir sólo una opción específica de varias disponibles.

Para tener varios botones radio en un mismo grupo y que sólo se active uno de ellos, hay que colocar el mismo valor en el atributo `name` a cada opción. El ejemplo anterior quedaría de esta forma:

```
<form method="post" action="/send/">
  <input type="radio" name="category" value="user" checked> Particular
  <input type="radio" name="category" value="business"> Empresa
</form>
```