

			Clase 10
--	--	--	--------------------

Tópico

HTML + CSS

Metas de comprensión

- ✓ Los alumnos comprenderán porque es tan importante el uso y el desarrollo de formularios en HTML
- ✓ Los alumnos desarrollaran la comprensión de los distintos controles de formulario como así también el uso de sus propiedades.
- ✓ Los alumnos desarrollaran la comprensión de los tópicos dados hasta el momento codificando de los distintos ejercicios expuestos y ejemplificados en las clases.

Desempeño de Exploración

Se expone en clase el uso de formularios, sus controles y el uso de los mismos. Los alumnos exponen el uso de esta herramienta en los distintos sitios web. Se analiza la maquetación en general y los estilos que se pueden aplicar. Se proponen ejercicios para desarrollar.

Introducción

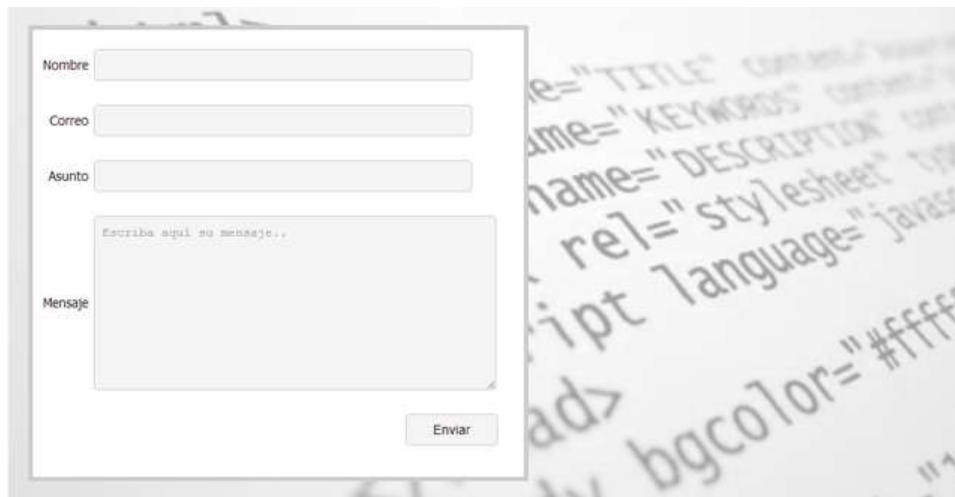
¡Cada clase según su contenido puede tener un numero variable de páginas, a leer, no te asustes!! La materia esta lo más actualizado posible al año 2022, de manera tal que tengas todos los temas en forma completa, de allí su extensión.

Algunas recomendaciones que te pueden ayudar a la hora de comprender el material de estudio:

- ✓ Lee varias veces la clase si fuera necesario.
- ✓ Subraya, destaca o resume los conceptos que creas principales o de importancia en cada tema.
- ✓ Puedes grabar la lectura de la clase (no necesaria puedes ser tú, puede ser un familiar, amigo etc.) para poder escucharla luego en el colectivo, recreo, relax o fin de semana etc.
- ✓ Muchas veces los números o medidas (velocidades, tamaños, resoluciones, latencia, potencia, capacidad, etc.) no son tan importantes ya que la informática evoluciona día a día y esos son bastantes cambiantes en la guerra de las empresas y fabricantes para sobresalir en el mercado, por eso no es necesario que los memorices a todos a menos que el tutor te lo indique.
- ✓ Puedes realizar grafica con cuadros,
- ✓ Puedes ampliar tu conocimiento con investigaciones adiciones en la web o viendo videos en youtube que refuercen los conceptos
- ✓ Puedes consultarle a tu tutor por cualquiera de las vías indicadas en el campus por el temario si tienes alguna duda, consulta o inquietud.
- ✓ Trata de organizar tu tiempo para la lectura y la comprensión del material
- ✓ Este texto va a acompañar a todas las clases para recordarte como leer, estudiar y comprender el material de la materia
- ✓ Cada clase en el campus podrá estar acompañada de material adicional como profundizar los temas para la mejor interpretación de los mismos con videos, encuestas, foros, actividades individuales y/o grupales.
- ✓ Busca algún compañero de estudio para poder compartir conocimiento, apuntes y metodología de estudio
- ✓ No esperes hasta último momento para realizar tus consultas, leer o ponerte al tanto con la materia

¿Dónde se produce la interacción del blogger con sus seguidores? En el formulario de comentarios. ¿Dónde puedo contarte una sugerencia sobre tu web? En el formulario de contacto. ¿Cómo puedo saber cuánto me costará el proyecto que tengo en mente? En el formulario de petición de presupuesto...

En todos los sitios web existen acciones que ejecuta el usuario que se resuelven mediante formularios: hacer una búsqueda, completar encuestas, realizar consultas, enviar un adjunto, etc. Y el caso más habitual es el formulario de contacto que permite enviar un mensaje sin necesidad de salir del sitio.

A contact form with a light gray border. It contains four input fields: 'Nombre' (Name), 'Correo' (Email), 'Asunto' (Subject), and 'Mensaje' (Message). The 'Mensaje' field is a larger text area with the placeholder text 'Escriba aquí su mensaje..'. Below the fields is a button labeled 'Enviar'. The background of the image shows a blurred snippet of HTML code, including tags like 'TITLE', 'KEYWORDS', 'DESCRIPTION', 'stylesheet', and 'language'.

¡No te pierdas esta clase, comencemos!

Formularios

La Web se ha convertido en una poderosa arma para las empresas que se dedican a realizar encuestas o compras, los formularios han sido una de las herramientas que han ayudado a este auge.

- ✓ Los *formularios* nos van a permitir, desde dentro de una presentación web, solicitar información al visitante. Estarán compuestos por tantos campos como informaciones queramos obtener. Una vez introducidos los valores en estos campos serán enviados a una URL donde se procesará toda esta información.
- ✓ Los *formularios* permiten obtener información de un cliente que visita una Web. Utilizando esta herramienta un usuario puede enviar datos para que luego sean procesados en el servidor.
- ✓ Un *formulario* contiene dos tipos de elementos: campos de datos y control. Los primeros permiten al usuario introducir información, y los segundos indicar qué debe hacerse con esos datos. Los datos son ingresados por el usuario, y el control son procesos que en cierta ocasión se realizan para permitir procesar dicha información.
- ✓ El *formulario* no es más que una página escrita en **HTML** como cualquier otra. Lo que la hace diferente es que permite capturar datos de varias maneras: directamente desde el teclado, escogiendo un valor de una lista desplegable o seleccionando una opción desde botones.

Una vez terminada la captura de datos, estos forman un paquete de información, que es enviado a un servidor http (web) que lo procesará y devolverá una respuesta. Pueden utilizarse también para enviar la información por correo electrónico o como un archivo html.

En el caso de que el resultado de un formulario sea un correo o un html, es sencillo de manejar. La complicación se produce cuando la información debe ser procesada en el servidor, porque debe existir un programa que sepa cómo manejar la información de ese formulario, esto quiere decir que se debe hacer un programa para que sirva de puente entre el formulario y el servidor, normalmente estas aplicaciones se denominan **CGI** (Common Gateway Interface). Los programas **CGI** pueden estar escritos en cualquier lenguaje de programación que sea soportado por el sistema operativo del servidor, y pueden estar diseñados para cualquier función: desde una simple captura de datos, que serán guardados en un archivo, hasta la más sofisticada consulta a una base de datos.

Formulario Básico

Todo formulario debe estar encerrado entre el par de etiquetas **<FORM>** y **</FORM>**, y debe ser ubicado en el cuerpo de cualquier documento HTML, es decir, entre el par de etiquetas **<BODY>** y **</BODY>**. Esta etiqueta **<FORM>** presenta tres atributos posibles

ACTION	El valor de este parámetro es la URL del programa o guion en el <i>Servidor Web</i> utilizado para procesar la información recolectada.
METHOD	Puede asumir el valor GET o el valor POST , y definen la manera en la cual los datos son transferidos al servidor.

Los dos primeros atributos de la tabla son de uso obligatorio para cualquier formulario que generemos, ya que establecen dónde enviar la información y cómo enviarla. Cuando definimos un formulario lo hacemos de la siguiente manera:

<form> controles del formulario... </form>

Destino de la información

El formulario comienza con esta línea:

<FORM ACTION="mailto:alumno@issd.edu.ar" METHOD=POST>

Como dijimos anteriormente todo formulario debe comenzar con la etiqueta **<FORM>** y finalizar con **</FORM>**. El parámetro **ACTION** define qué es lo que debemos hacer con la información obtenida, en este caso le indicamos al navegador que nos envíe los datos a nuestra dirección de correo electrónico (en el ejemplo la mandan a issd). La mayoría de los formularios en Internet realizan esta tarea, es decir, recolectan información del usuario y la envían por correo electrónico hacia algún destino.

Controles de Formularios

Todos los controles que pueden usarse a la hora de crear un formulario permitirán al navegante insertar información. En algunas oportunidades esta información consistirá en la inserción de un texto, que podrá ser de una sola línea o varias y que podrá estar oculto o no. En otras ocasiones se ofrecerá entre una serie de opciones entre las que simplemente hay que elegir una o varias. Todo ello se podrá realizar de diferentes formas que ahora mismo comenzamos a estudiar.

Para empezar, vamos a ver una serie de controles todos los cuales se crean con la etiqueta **INPUT**. Para diferenciar entre unos controles y otros se hará uso del atributo **TYPE** que puede tomar los siguientes valores: text, password, radio, checkbox, submit, image, reset, file y hidden.

Además del atributo **TYPE**, esta etiqueta consta de algunos más cuya función variará del tipo de control estemos insertando, es decir, en función del valor del atributo **TYPE**. Uno común a todos es **NAME**, ya mencionado, con el que damos un nombre al control, y que es recomendable usarlo siempre. Una vez hecho este comentario, podemos empezar a estudiar cada uno de los tipos de control uno a uno.

Cuadro de texto

Esta misma etiqueta será la utilizada para insertar la mayoría de controles acompañada de **TYPE** por ejemplo:

<INPUT TYPE="text" SIZE=15 MAXLENGTH=15 NAME="Nombre">

- ✓ **Size:** Determina la *anchura* de la caja de texto. El valor por defecto de este atributo es 20 caracteres. En ocasiones convendrá decrementar o incrementar este valor por defecto, pero en todo caso siempre será conveniente mantener un valor menor de 50 caracteres de manera que la caja de texto quepa en la mayoría de pantallas
- ✓ **Maxlength:** Con este atributo limitamos el número *máximo de caracteres* que pueden ser escritos en una caja de texto. El valor de este atributo puede ser mayor o menor que el especificado en **SIZE**, y que es totalmente independiente. Si es mayor cuando lleguemos

al final de la caja de texto los nuevos caracteres que insertemos irán desplazando hacia la izquierda a los primeros, que dejarán de estar a la vista.

```
<INPUT TYPE="text" SIZE="20" MAXLENGTH="20">
<INPUT TYPE="text" SIZE="20" MAXLENGTH="5">
<INPUT TYPE="text" SIZE="20" MAXLENGTH="40">
```

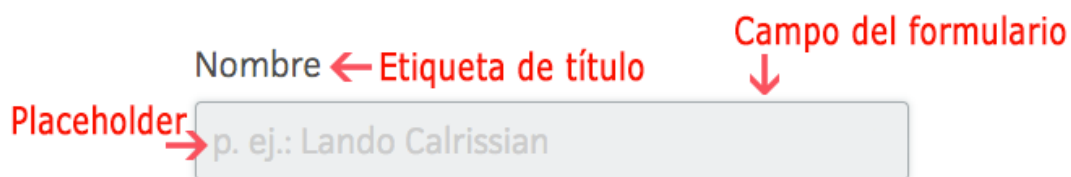
Por último, queda decir que si no usamos el atributo **MAXLENGTH** el número de caracteres que pueden introducirse en la caja de texto no tendrá límite.

- ✓ **Value:** Sirve para especificar un texto que debe aparecer por defecto en la caja de texto, antes de que el usuario escriba nada. Este texto suele ser, en general, o bien instrucciones o bien la respuesta más probable. Veamos un ejemplo que ilustra estos dos casos:

```
<INPUT TYPE="text" VALUE="Introduzca aquí su nombre">
```

- ✓ **Name:** define el nombre que se le da al control, esto puede servir para que el lenguaje dinámico lo tome como lectura y lo pueda manipular como propio.
- ✓ **Placeholder:** Por ejemplo, para dar más información sobre el tipo esperado de información que queremos que el usuario introduzca, que desaparece cuando el control de formulario adquiere el foco.

```
<input type="text"... placeholder="Tu nombre es..">
```



- ✓ **Autofocus:** este atributo hace que un campo de formulario adquiriera el foco de forma automática cuando se carga una página. Su sintaxis es la siguiente:

```
<input type="text" autofocus ... >
```

El atributo “**autofocus**” es de tipo booleano (o es verdadero o es falso) .

Introduzca su nombre:

Introduzca su edad:

Cuadro de texto para claves

En ocasiones puede ser conveniente pedir al usuario algún tipo de información confidencial como puede ser una clave de acceso o *password*.

Debemos tener en cuenta que algunos usuarios pueden estar en algún lugar público o, al menos, con gente alrededor que pueden leer lo que escriba en la pantalla y por tanto no le interesará que lo que escribe sea mostrado en ésta.

<H2>Formulario de autenticación</H2>

FORM>

<P>Introduzca su nombre: <INPUT TYPE="text" NAME="nombre">

<P>Introduzca su clave: <INPUT TYPE="password" NAME="clave">

</FORM>

Los cuadros de texto para claves también tienen los atributos **SIZE**, **MAXLENGTH**, **VALUE** y **ALIGN** que vimos en las cajas de texto. Al igual que hicimos con estas, veamos un resumen de todos los atributos de las cajas de texto para claves:

<INPUT TYPE="password" NAME="clave" SIZE="tamaño"

MAXLENGTH="longitud_máxima" VALUE="clave por defecto" ALIGN="alineamiento">

Rellene los campos del siguiente formulario con los valores deseados para generar los archivos .htaccess y .htpasswd:

Usuario:

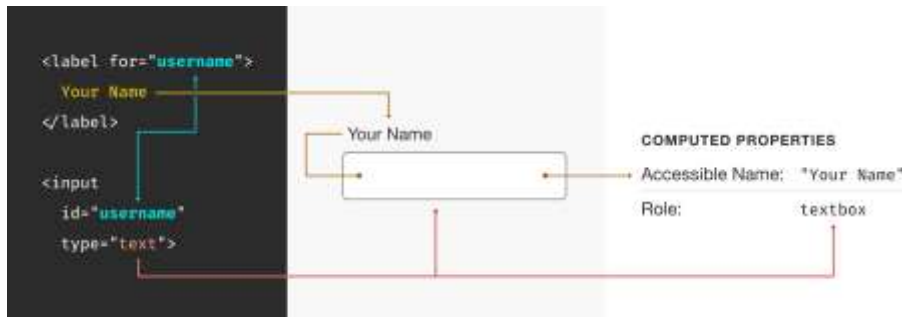
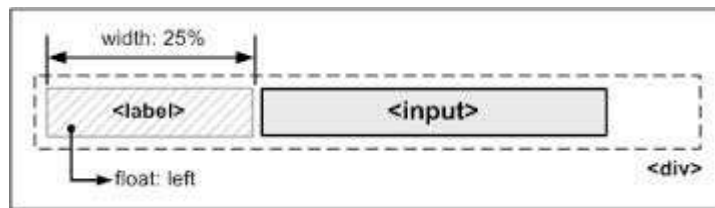
Contraseña:

Etiquetar controles

En el caso de querer incluir una etiqueta a un elemento utilizamos la etiqueta **<LABEL>...</LABEL>**, esta etiqueta no sólo permite darle un texto a un elemento de un formulario, sino que comparte el control del mismo. Esto quiere decir que podemos seleccionar no sólo al elemento, sino que también a su rótulo.

<label for="nombre">Nombre:</label>

<input type="text" id="nombre" name="fnombre" />



Botón de Envío

Este control envía automáticamente el formulario en que se encuentra cuando es presionado. Es decir, cuando se pulsa el botón el navegador nos lleva a la dirección web indicada en el campo *action* y al mismo tiempo envía a esa dirección los datos que contiene el formulario.

Para insertar un botón usamos la etiqueta `<input ... />`. Para indicar que se trata de un botón de envío dentro de esta etiqueta escribiremos `type = "submit"`.

```
<!DOCTYPE HTML>
<body>
<form method="post" action="enviar.php">
<input type="submit" value="Enviar este formulario" />
</form>
</body>
</html>
```

Al pulsar sobre el botón 'Enviar este formulario' es que el navegador irá a la página 'enviar.php' y recibirá los datos que pudiera llevar el formulario.

Botón de borrado

Este tipo de botones reestablecen los controles en un formulario a sus valores iniciales cuando es presionado. Son insertados con la etiqueta (tag) HTML input con el valor `"reset"` en su atributo `"type"`.

```
<!DOCTYPE HTML>
<body>
<form method="post" action="enviar.php">
<input type="text" name="texto1" value="Valor por defecto" ><br >
<input type="checkbox" name="condiciones" checked="checked" >Acepto las condiciones
<br>
<input type="reset" value="Limpiar campos">
</form>
</body>
```

Botones genéricos

Los botones genéricos pueden ser usados como botones de envío o restablecimiento, o bien pueden no tener ninguna acción preestablecida (dependiendo del valor de su atributo "type"). Su característica es que se puede insertar contenido HTML dentro de ellos.

Los botones de contenido se insertan con el tag HTML button, usando el valor "button", "submit" o "reset" en su atributo "type". Un valor button supondrá que el botón no hará nada en respuesta a un click, a no ser que se especifique una acción de otra manera.

```
<!DOCTYPE HTML >
<body>
<form method="post" action="enviar.php">
  <button type="submit">
    <b>Boton de Tipo Submit</b>
  </button>
</form>
</body>
</html>
```

Hemos escrito <button type="submit">, lo que supondrá que el botón funcione como un botón de envío. Si hubiéramos escrito <button type="reset"> el botón funcionaría como un botón de restablecimiento de valores por defecto, y si hubiéramos escrito <button type="button"> el botón no daría lugar a ninguna acción. En caso de no especificarse type para el botón, en general se entenderá que es un botón tipo submit.

Otro ejemplo del uso de estos botones es asignándole código JavaScript, que en este caso permitirá cerrar la ventana del navegador.

```
<INPUT TYPE="button" VALUE="Cerrar ventana"
  onClick="window.close();">
<INPUT TYPE="button" VALUE="Cargar otra ventana"
  onClick="window.location.replace ('button2.html');">
```

Botones de elección

Estos controles reciben también el nombre de botones de radio, como traducción directa de su denominación inglesa *radio buttons* y porque en general los navegadores suelen darles una forma circular. Para insertar un botón de elección usaremos de nuevo la etiqueta INPUT, pero esta vez el valor del atributo TYPE será radio:

```
<INPUT TYPE="radio">
```

Este tipo de controles tiene dos estados o posiciones: seleccionado (ON) o no seleccionado (OFF), estando inicialmente todos en la posición **OFF**. También es posible especificar que un botón determinado esté seleccionado inicialmente insertando el atributo **CHECKED** en la etiqueta **INPUT** de la siguiente manera:

```
<INPUT TYPE="radio" CHECKED>
```

Los botones de elección suelen ser insertados en los formularios en grupos, dando al usuario la posibilidad de elegir entre una serie de opciones.

Cuando pulsamos sobre un botón de radio le pasamos a la posición ON y permanecerá en ese estado hasta que pulsemos en otra opción del mismo grupo. Esto es así porque entre los botones de radio de un mismo grupo sólo uno de ellos puede estar seleccionado, por tanto, cuando seleccionamos uno, aquel que estuviese seleccionado previamente dejará de estarlo.



Para indicar que una serie de botones de elección pertenecen a un mismo grupo debemos incluir el mismo valor en el atributo **NAME** en todos ellos. Además, debemos usar el atributo **VALUE** para dar un nombre distinto a cada uno de los botones. Veamos un ejemplo:

```
<FORM>
<P><INPUT TYPE="radio" NAME="musica" VALUE="Jazz">Jazz
<P><INPUT TYPE="radio" NAME="musica" VALUE="Pop">Pop
<P><INPUT TYPE="radio" NAME="musica" VALUE="Rock">Rock
<P><INPUT TYPE="radio" NAME="musica" VALUE="Country">Country
</FORM>

<form>
<input type="radio" id="radio1" name="Grupo">Opcion 1
<br>
<input type="radio" id="radio2" name="Grupo">Opcion 2
<br>
<input type="radio" id="radio3" name="Grupo">Opcion 3
<br><br>
<input type="button" value="Mostrar" onClick="mostrarSeleccionado()">
</form>
```

Cuadro de selección

Las cajas de selección guardan ciertos parecidos con los botones de radio, pero además permitirán seleccionar varias opciones en una lista. Al igual que dichos botones tienen dos posiciones, seleccionados o no seleccionados, estando en esta última posición inicialmente, a no ser que hayamos usado el atributo **CHECKED**. Cada caja de selección es independiente del resto, y por tanto el valor del atributo **NAME** debe ser diferente en cada una.

Para insertar una caja de selección debemos usar de nuevo la etiqueta **INPUT**, pero esta vez con **TYPE="checkbox"**. Veamos un ejemplo de uso de este tipo de controles:

```
<FORM>
<P><INPUT TYPE="checkbox" NAME="medico">Médico
<P><INPUT TYPE="checkbox" NAME="programador" CHECKED>Programador
<P><INPUT TYPE="checkbox" NAME="abogado">Abogado
<P><INPUT TYPE="checkbox" NAME="ingeniero">Ingeniero
<CENTER><INPUT TYPE="submit" VALUE="Enviar"></CENTER>
</FORM>
```

Pulsando sobre cada caja de selección se cambia de seleccionado a no seleccionado, independientemente de la posición del resto. En el código podemos apreciar que es posible indicar que una caja de selección esté seleccionada por defecto.

Seleccione su profesión (escoja todas las que procedan):

☐ Médico
☒ Programador
☐ Abogado
☒ Ingeniero

Enviar

<INPUT TYPE="checkbox" NAME="ingeniero" VALUE="SI">Ingeniero

```
<html>
<head></head>
<body>
<input type="checkbox" class="someone" name="any" disabled>
<input type="checkbox" class="someone" name="any" disabled>
<input type="checkbox" class="someone" name="any" disabled>
</body>
</html>
```

```
<form>
<input type="checkbox" name="male" value="m">CONTENT
<input type="checkbox" name="female" value="f">CONTENT
</form>
```

Datos numéricos

<input type="number" ... >

Esto crea un tipo especial de campo de entrada para la entrada de números. En la mayoría de navegadores que lo soportan se presenta como un campo de entrada de texto con un control que permite incrementar y disminuir su valor.

- ✓ **min y max:** Como su propio nombre indica, este par de atributos permiten establecer un límite inferior y superior para los valores que se pueden introducir en un campo de formulario numérico, como los tipos de entrada numéricos, de rango, fecha y hora (sí, hasta se pueden utilizar para establecer los límites superior e inferior para las fechas: por ejemplo, en un formulario de reserva de viajes podrías limitar el selector de fechas para que sólo permita al usuario seleccionar fechas futuras). Para entradas **range**, de hecho, **min** y **max** son necesarios para definir los valores que se devuelven cuando se envía el formulario.



```
<input type="number" ... min="1" max="10">
```

El siguiente ejemplo admite valores pares comprendidos entre 2 y 12.

```
<input type="number" min="2" max="12" step="2" >
```

```
<form>
  <input type="number"
    min="0"
    max="10"
    step="2"
    value="6">
```

Sliders

```
<input type="range">
```

Crear un control deslizante que permita elegir entre una gama de valores. Ten en cuenta que, por defecto, esta entrada no suele mostrar el valor seleccionado, ni siquiera el rango de valores que cubre. A continuación, vemos otro ejemplo:



```
<input type="range" name="edad" min="18" max="99" step="1" value="30">
```

- ✓ **máx y min** para establecer el rango de valores máximo y mínimo.
- ✓ **value** para especificar un valor por defecto.
- ✓ **step** para establecer el valor numérico de avance del slider o barra de desplazamiento

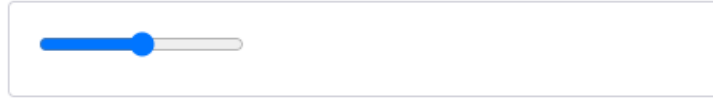
```
<div id="slider">
5% <input id="slide" type="range"
  min="5" max="200" step="5" value="100"
  onchange="updateSlider(this.value)" />
200%
</div><br/>
```

Atributos válidos para este control:

- ✓ **name:** Nombre del input (name="nombre del input").
- ✓ **disabled:** Atributo que determina si está desactivado el input para la introducción de datos (disabled="disabled"), por defecto activado.
- ✓ **autocomplete:** Con autocomplete activado el input se autorrellenará con contenidos anteriormente enviados por él mismo (autocomplete="off", autocomplete="on")
- ✓ **autofocus:** Este atributo hace que el cursor se sitúe en el input (autofocus="autofocus").
- ✓ **list:** Asocia mediante id una lista de valores predefinidos (list="nombre lista")
- ✓ **min y max:** Fijan los valores máximos y mínimos que puede tener el **range input** (max="5" y/o min="0").
- ✓ **step:** Valor numérico con el cual varías el intervalo del input (step="2", sumaría o restaría de 2 en 2 en el slider).

- ✓ **value:** Valor por defecto del input.

```
<input type="range" min="5" max="10" step="0.01">
```



Fecha y Hora

HTML5 tiene diferentes tipos de entrada para la creación de selectores complejos de fecha y hora, por ejemplo, como los selectores de fecha que se ven en casi todos los sitios de reservas de vuelos o trenes. Veremos los siguientes tags:

- ✓ **datetime:** permite elegir una fecha y una hora
- ✓ **month:** permite elegir un mes, almacenado internamente como un número entre 1 y 12, aunque los diferentes navegadores pueden dar mecanismos más elaborados de selección, como listas de desplazamiento con los nombres de los meses
- ✓ **week:** permite elegir semana, almacenada internamente con el formato 2010-W37 (semana 37 del año 2010), mediante un selector de fechas.

A tener en cuenta para estos controles:

- ✓
- ✓ El **formato de fecha** devuelto por defecto es AAAA-MM-DDTHH:MM.
- ✓ Se puede configurar un mínimo de fecha a seleccionar con el atributo '**min**': min="2013-10-08T23:59:60Z".
- ✓ Se puede configurar un máximo de fecha a seleccionar con el atributo '**max**': max="2014-10-08T23:59:60Z".
- ✓ Con el atributo '**value**' se puede poner el calendario con una fecha concreta por defecto: value="2013-10-08T23:59:60Z".
- ✓ Con el atributo '**step**' puedes establecer el avance o retroceso de días con las flechas del calendario.

Fecha y hora actual: `<input type="datetime" name="fechahora" step="1" min="2014-01-01T00">`

Los valores usados en este ejemplo son:

- ✓ **min:** He puesto que la fecha mínima sea el 1 de enero 2021 a las 00:00.
- ✓ **max:** He puesto que la fecha máxima sea el 31 de diciembre 2021 a las 12 am.
- ✓ **value:** Fecha por defecto el 1 de enero 2021 a las 12.
- ✓ **step:** Que avance en intervalos de 1 en 1

```
<input type="date" ... >
```

```
<input type="time" ... >
```

Estos atributos crean un selector de fecha plenamente funcional y una entrada de texto que contiene un separador para horas, minutos y segundos que sólo permite introducir un valor de tiempo.

```
<input type="time" ... step="1800">
```

- ✓ **Step:** El atributo step se puede utilizar con un valor de entrada numérico para dictar la granularidad de los valores que se pueden introducir. Por ejemplo, es posible que desees

que los usuarios introduzcan una hora determinada, pero sólo en incrementos de 30 minutos. En este caso, podemos usar el atributo *step*, teniendo en cuenta que para entradas time el valor del atributo está en segundos:

```
<form>
  <div>
    <label for="bday">Escriba su cumpleaños:</label>
    <input type="date" id="bday" name="bday">
  </div>
</form>
```

```
<form>
  <div>
    <label for="party">Elija la fecha de festejo que prefiera:</label>
    <input type="date" id="party" name="party" min="2017-04-01" max="2017-04-30">
  </div>
</form>
```

```
<HTML>
<BODY>
<label for="hr">Horario de Reunion:</label>
<input type="time" name="hr" min="09:00" max="18:00" required>
</BODY>
</HTML>
```

Horario de Reunion:

16	22
17	23
18	24
19	25
20	26
21	27
22	28

```
<FORM name="ejemplo1" id="ejemplo1">
Nombre <input type="text" name="nombre" id="nombre" required />
Edad <input type="number" name="edad" id="edad" max="50" min="10" step="2" />
Fecha <input type="date" id="nacimiento">
</FORM>
```

Color

Este tipo de entrada muestra un selector de color. La implementación de Opera permite al usuario elegir entre una variedad de colores, introducir valores hexadecimales directamente en un campo de texto o invocar el selector de color nativo del sistema operativo.

`<input type="color">`



Búsqueda

`<input type="search">`

Podría discutirse que el tipo de entrada de búsqueda no es más que una entrada con un estilo diferente. Por ejemplo, un navegador podría ofrecer al usuario una opción para crear automáticamente una búsqueda personalizada para un sitio específico.

HTML5 search input

Introduce el término de búsqueda:

Teléfono

A la hora de completar un input de tipo tel, un smartphone como el iPhone convierte su teclado a números de teléfono.

`<input type="tel" name="telefono">`

```
<label for="phone">Enter your phone number:
</label>

<input type="tel" id="phone" name="phone"
       pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"
       required>

<small>Format: 123-456-7890</small>
```

Mail

El nuevo tipo `<input type="email">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una dirección de email *válida*, pero no comprueba si la dirección existe o no, sólo si el formato es válido. Como ocurre con el resto de campos de entrada, puede enviar este campo vacío a menos que se indique que es obligatorio.

El atributo *múltiple* indica que el valor de este campo, puede ser una lista de emails válidos, separados por comas.

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

Nombre:

Email:

Url:

Introduzca una dirección de correo.

Comentario:

Url

`<input type="url">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una URL correcta. Algunos navegadores ofrecen ayudas al usuario, como Opera que añade el prefijo *http://* a la URL si el usuario no lo ha introducido.

`<input type="url" name="url">`

En este campo, el teclado del iPhone es qwerty pero en modo "url", ya que ofrece teclas fundamentales para escribir una dirección web como son el punto, la barra "/" o la tecla ".com".

URL:

Introduce una URL

El atributo autocomplete

El atributo *autocomplete* especifica que el formulario o el campo de entrada de formulario debe tener una función de autocompletado.

Funciona para las etiquetas `<form>`, y con los siguientes tipos de `<input>`: **text**, **search**, **url**, **telephone**, **email**, **password**, **datepickers**, **range**, y **color**.

Cuando el usuario empieza a escribir en un campo de autocompletar, el navegador debe mostrar las opciones para rellenar el campo:

Ejemplo:

```
<form action="enviar.asp" method="post" autocomplete="on">
  Nombre: <input type="text" name="txtnom" ><br >
  Apellido: <input type="text" name="txtape" ><br >
  E-mail: <input type="email" name="email" autocomplete="off"><br >
  <input type="submit">
</form>
```

Texto multilínea

si deseamos que el usuario pueda escribir el texto de un *mail*, o escribir una serie de comentarios largos, las áreas de texto serán tremendamente útiles. La etiqueta usada para insertar este nuevo control es **TEXTAREA**, que consta de cuatro atributos:

- ✓ **NAME**: El nombre que queremos asignarle al control. Como siempre, este nombre será enviado junto con los datos del área de texto al mandar el formulario.
- ✓ **ROWS**: El número de líneas de la caja de texto.
- ✓ **COLS**: El número de caracteres visibles de cada línea. Este atributo es similar al atributo **SIZE** que vimos en las cajas de texto convencionales.

La etiqueta **TEXTAREA** está compuesta por una instrucción de inicio y una instrucción de fin, siendo obligatorio el uso de ambas. Entre ellas únicamente podrá insertarse texto llano (sin ninguna etiqueta HTML) y éste será mostrado como contenido inicial del área de texto. Veamos un ejemplo:

```
<TEXTAREA NAME="texto" ROWS="10" COLS="50">
Es la mejor que he visto nunca.</TEXTAREA>
```

Si no pulsamos ENTER y llegamos al final de la línea el navegador debe decidir qué hacer.



Cuadros de selección

Su función será dar a elegir entre una serie de opciones de manera que el usuario pueda elegir una o varias de ellas. Existen dos formas de mostrar estas opciones, como una *lista con desplazamiento*, o mediante una *Lista desplegable*. Para insertar un cuadro de selección usaremos la etiqueta **SELECT**, que consta de una instrucción de inicio y de una instrucción de fin, entre las cuales introduciremos las diferentes opciones para el usuario. Para insertar estas opciones usaremos una nueva etiqueta, **OPTION**, que consta de una única instrucción. Veamos un ejemplo que ilustra el uso de uso de este tipo de control:

Ejemplo 1

```
<FORM METHOD="POST" ACTION="mailto:alumno@issd.edu.ar">
  <INPUT TYPE="SUBMIT" VALUE="Enviar">
  <INPUT TYPE="RESET" VALUE="Borrar">
  <SELECT NAME="lista" MULTIPLE SIZE=3>
    <OPTION>Valor 1
    <OPTION SELECTED>Valor 2
    <OPTION>Valor 3
    <OPTION value="Valor 4">Valor 4 especial
```



```

        <OPTION>Valor 5
        <OPTION>Valor 6
        <OPTION>Valor 7
    </SELECT>
</FORM>

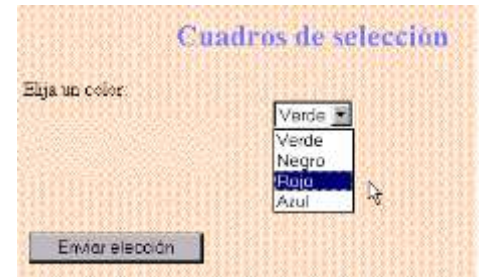
```

Ejemplo 2

```

<FORM>
Elija un color:<BR>
<SELECT NAME="color">
    <OPTION>Verde
    <OPTION>Negro
    <OPTION>Rojo
    <OPTION>Azul
</SELECT>
<INPUT TYPE="submit" VALUE="Enviar elección">
</FORM>

```



Cuando se pulse el botón 'Enviar elección' se mandará al servidor el texto que sigue a la etiqueta **OPTION**, que se haya escogido (Por ejemplo '*color=Rojo*').

Existe un atributo, *SELECTED*, que nos permitirá elegir la opción que debe estar seleccionada por defecto. Por ejemplo, si en el código anterior queremos que el color negro aparezca seleccionado inicialmente tendremos que escribir:

```

<SELECT NAME="color">
    <OPTION>Verde
    <OPTION SELECTED>Negro
    <OPTION>Rojo
    <OPTION>Azul
</SELECT>

```

Al principio hemos comentado que los cuadros de selección pueden mostrarse como lista desplegable, o bien como listas con desplazamiento.

Hasta ahora sólo hemos visto el primer caso en el que únicamente la opción seleccionada es visible. En ocasiones puede ser conveniente que todas, o al menos algunas de las posibilidades sean mostradas directamente, sin tener que pulsar un botón. Para conseguir esto el lenguaje **HTML** ofrece el atributo *SIZE*, que determina el número de opciones que pueden ser vistas simultáneamente. Si este valor es 1 (o no se usa el atributo **SIZE**), se mostrará una persiana desplegable, como hemos visto hasta ahora. Si es mayor, se mostrará una lista, que podrá tener barras de desplazamiento si no caben todas las opciones disponibles. Veamos un ejemplo:

```

<FORM>
Lista de la compra:<BR>
<SELECT NAME="compra"
    SIZE="6">
    <OPTION VALUE="Fruta"> Fruta
    <OPTION VALUE="Verdura">Verdura
    <OPTION VALUE="Azucar">Azucar
    <OPTION VALUE="Mantequilla">Mantequilla
    <OPTION VALUE="Yerba">Yerba
    <OPTION VALUE="Pasta"> Pasta
</SELECT>
<P>

```



```
<INPUT TYPE="submit" VALUE="Enviar lista de la compra">
</FORM>
```

En este caso el número de opciones es igual al tamaño que le hemos dado al control, por lo que no han sido necesarias barras de desplazamiento.

Si no indicamos lo contrario, los cuadros de selección actúan como los botones de radio, es decir, sola una opción puede ser elegida simultáneamente. Afortunadamente, es posible cambiar este comportamiento, y permitir la selección de varias opciones, usando el atributo *MULTIPLE* en el interior de la etiqueta **SELECT**. En el ejemplo de la lista de la compra esta posibilidad es de gran utilidad, añadiendo este atributo al código anterior queda:

```
<FORM>
Lista de la compra:<BR>
<SELECT NAME="compra" SIZE="6" MULTIPLE>
  <OPTION VALUE="Fruta"> Fruta
  <OPTION VALUE="Verdura">Verdura
  <OPTION VALUE="Ternera">Ternera
  <OPTION VALUE="Mantequilla">Mantequilla
  <OPTION VALUE="Salchichas">Salchichas
  <OPTION VALUE="Pasta"> Pasta
</SELECT>
<P>
  <INPUT TYPE="submit" VALUE="Enviar lista de la compra">
</FORM>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>_</head>
  <body>
    <select> == $0
      <option value="value1">opcion 1</option>
      <option value="value2">opcion 2</option>
      <option value="value3">opcion 3</option>
    </select>
  </body>
</html>
```

Agrupando Elementos

Dentro de un formulario podemos agrupar los elementos por tema, permitiendo un mejor aspecto. Para ello se utiliza la etiqueta **<FIELDSET>**... **</FIELDSET>**, que permite agrupar elementos de un formulario, creando una caja alrededor de los mismos. La etiqueta **<LEGEND>**... **</LEGEND>**, nos permite colocar un título dentro del grupo, este título puede colocarse arriba (top), abajo (bottom), izquierda (left) o derecha (right).

```

<FORM METHOD="POST" ACTION="mailto:alumno@issd.edu.ar" >
<FIELDSET>
<LEGEND> Datos del Cliente</LEGEND><BR>
Nombre: <INPUT type="text" name="nombre_entrega" size=40 maxlength=30><BR>
Dirección: <TEXTAREA name="direccion_entrega" rows="3"
cols="33"></TEXTAREA><BR>
Ciudad: <INPUT type="text" name="ciudad_entrega" size=15 maxlength=20>
Provincia: <INPUT type="text" name="provincia_entrega" size=15 maxlength=20><BR>
Teléfono: <INPUT type="text" name="telefono_entrega" size=17 maxlength=17><BR>
</FIELDSET>
<HR>

<FIELDSET>
<LEGEND><STRONG>FORMA DE PAGO:</STRONG></LEGEND><BR>
Tipo de pago: <BR>
<INPUT type="radio" name="forma_pago" value="efectivo" checked id="efect">
<LABEL for="efect">Efectivo</LABEL>
<INPUT type="radio" name="forma_pago" value="tarjeta" id="tarj">
<LABEL for="tarj">Tarjeta </LABEL>
<P>
Seleccione su Tarjeta:
<SELECT name="Banco_tarjeta">
<OPTION selected>Banco Nacional<option>Banco Provincial<option>Banco Privado
</SELECT>
Nº Tarjeta: <INPUT type="text" name="numero_tarjeta" size=20 maxlength=20>
</FIELDSET>
<HR>
<INPUT type="submit" value="Comprar"><INPUT type="reset" value="Borrar
formulario">
</FORM>

```

En este ejemplo podemos observar un formulario que se envía por mail en donde tenemos dos grupos: “Datos del Cliente” y “Forma de Pago”; en el segundo grupo tenemos dos botones con su **LABEL** que nos permite no sólo seleccionar al botón sino también a la leyenda.

```

<FIELDSET>
<LEGEND>Tus datos</LEGEND>
<LABEL>
Nombre: <INPUT TYPE="text" name="nombre">
<br>Edad: <INPUT TYPE="text" name="edad">
</LABEL>
</FIELDSET>

```

Todos los controles de un formulario se pueden desactivar, impidiendo así al usuario que los sobrescriba. Se seguirán mostrando en pantalla, aunque con un aspecto distinto para indicar que no son editables. Para ello sólo tenemos que indicarle el parámetro **DISABLED**:

```

<LABEL DISABLED>Texto:
<INPUT TYPE=TEXT DISABLED>
</LABEL>

```

Texto:

Teclas de acceso

Los *shortcuts*, o atajos de teclado, son accesos rápidos a distintas partes de nuestra web, con sólo tocar una combinación de teclas. Su implementación es sumamente sencilla, y basta con agregar `accesskey="[caracter de acceso rápido]"` dentro de una etiqueta, de la siguiente manera:

```
<input type="submit" accesskey="b" value="Buscar">
```

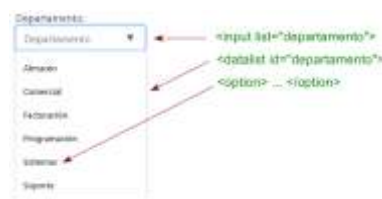
Los siguientes elementos soportan el atributo `accesskey`: **A**, **AREA**, **BUTTON**, **INPUT**, **LABEL**, **LEGEND** y **TEXTAREA**.

Este ejemplo asigna la tecla de acceso "U" a un rótulo asociado con un control input. Al pulsar la tecla de acceso, el foco se dirige hacia el rótulo, el cual a su vez lo dirige al control asociado. El usuario puede entonces introducir texto en el área input.

```
<FORM action="..." method="post">
<P>
<LABEL for="nombre-usuario" accesskey="U">
Nombre de Usuario:
</LABEL>
<INPUT type="text" name="usuario" id="nombre-usuario">
</P>
</FORM>
```

Datalist

La nueva etiqueta `<datalist>` permite automatizar el ingreso de datos. Genera una lista desplegable de opciones para que el usuario realice su selección. La ventaja de este control es que reconoce los primeros caracteres ingresados por el usuario, para "autocompletar" la información.



```
<label>Selecciona tu país:</label>
<input type="text" name="pais" id="pais" list="paises"/>
<datalist id="paises">
<option value="España" />
<option value="México" />
<option value="Argentina" />
<option value="Perú" />
<option value="Colombia" />
<option value="Otro país" />
</datalist>
```

Si el navegador no soporta esta nueva característica, mostrará un casillero de *input* sencillo, sin desplegar las opciones.

```
<input list="listacolores">

<datalist id="listacolores">
<option value="Azul">
<option value="Rojo">
<option value="Amarillo">
<option value="Negro">
<option value="Blanco">
</datalist>
```



Validación de Formularios

Atributo required

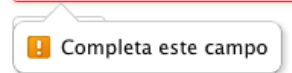
El atributo **required** controla que el campo de entrada tenga contenido. Si no es así, emite un mensaje o señal de error e impide enviar el formulario. Al incluir el atributo **required** dentro de un elemento `<input>`, automáticamente se hace obligatorio su llenado y al ser un atributo tipo booleano, solo se requiere su presencia nada más.

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

En el momento en que se encuentra un error de validación se ejecuta la acción por defecto del navegador, normalmente aparece un pequeño globo emergente (*tooltip*) conteniendo un texto de advertencia. Añadiendo el atributo `title` en la etiqueta `<input>`, se puede extender la información mostrada.

`<input title="Se necesita un nombre" type="text" name="nombre" required/>`

Nombre:



Nombre:



El atributo pattern

Pattern busca un patrón/expresión regular dentro de un campo de texto y determina si este lo cumple o no. Por tanto, su función es validar la información introducida por los usuarios.

Utilizando el atributo **pattern**, conjuntamente con **required** se logra que se verifique no solo la presencia de un valor, sino que este valor debe contener un formato, una longitud o un tipo de dato específico. Esto último se logra definiendo un patrón con expresiones regulares.

`<input type="text" pattern="^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+>`

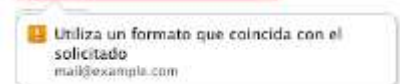
Con el patrón anterior se valida un formato correcto de correo electrónico (mail@example.com), pero en la práctica esto resulta un tanto absurdo, ya que definiendo un tipo email en la etiqueta `<input>`, el navegador por sí mismo hace la validación del tipo en cuestión.

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

Supongamos que se necesita que el usuario ingrese una dirección válida de IPV4, lo hacemos de la siguiente manera:

`<input title="0.0.0.0" type="text" name="ipv4">`

Email:



Email:



Cuatro bloques de uno a tres dígitos es la notación punto-decimal de una dirección válida de IPV4. El uso de *required* y *pattern* asegura que la entrada debe ser rellenada siguiendo la expresión regular.

```
<input type="text" id="referencia" name="referencia" required pattern="[A-Z]{3}[0-9]{4}" title="El formato debe coincidir con 3 letras mayúsculas y 4 números."/>
```

```
<form >
<label for="twitter">Usuario de Twitter: </label>
<input type="text" pattern="^[A-Za-z0-9_]{1,15}$" name="twitter_account" id="twitter" />
<input type="submit" value="Enviar" />
</form>
```

Si cumple con la expresión regular indicada “^[A-Za-z0-9_]{1,15}\$” (admite caracteres alfanuméricos y guion bajo hasta un máximo de 15), el formulario será enviado normalmente, sino, el navegador mostrará una advertencia y evitará que el formulario se envíe.



```
<input type="text" name="twitter_username"
      pattern="^@?(\\w){1,15}$"
      title="Este no parece un usuario de twitter válido">
```

```
<label for="tel">Teléfono (9 dígitos comenzando por 9, 8, 7 o 6)</label>
<input type="text" pattern="^[9|8|7|6]\\d{8}$">
```

```
<input type="text" name="twitter_username"
      pattern="^@?(\\w){1,15}$"
      title="Este no parece un usuario de twitter válido"
      required>
```



Conclusión

Los formularios dieron un salto cualitativo con HTML5, los nuevos tipos de input y la validación por el navegador. Hablamos mucho de experiencia de usuario y de cómo mejorar las expectativas de los visitantes en nuestras aplicaciones web, sobre cómo mejorar la velocidad de carga hasta detalles concretos como el número máximo de caracteres por línea para tener textos más legibles, ubicación de elementos en el layout, etc. Pero si hay un ítem en la web que concentra todos los sentidos del usuario ese es sin duda los formularios, es por eso que en esta clase finalizamos con este tema.

¡Nos vemos en la próxima clase, no te la pierdas!



Ejercitación

Codifica los siguientes formularios, aplica los estilos aprendidos cuando sean necesarios,

Ejercicio 1

Por favor, rellena estos datos	
Nombre:	<input type="text"/>
Apellidos:	<input type="text"/>
Dirección:	<input type="text"/>
Código Postal:	<input type="text"/>
Marca tu equipo favorito:	<input checked="" type="radio"/> Real Madrid <input type="radio"/> Atlético de Madrid
Marca tu música favorita:	<input checked="" type="checkbox"/> Rock <input checked="" type="checkbox"/> Pop <input type="checkbox"/> Heavy <input type="checkbox"/> Tecno
Elige un escritor:	<input type="text" value="Cervantes"/>
¿Algún comentario?	<div><div></div></div>
<div><div>enviar</div><div>borrar</div></div>	

Ejercicio 2:

Formulario de alta

Nombre y apellidos *

Nombre

Primer apellido

Segundo apellido

Dirección *

Calle, número, piso, puerta

Código postal

Municipio

Provincia

País

Email

Teléfono *

Fijo

Móvil

Darme de alta →

Ejercicio 3

Nombre:

Apellidos:

E-Mail:

Población:

Provincia:

Edad:

- ☐ 0-20
- ☒ 20-40
- ☐ 40-60
- ☐ 60-100

¿Cómo nos conociste?

- ☐ A través de un amigo.
- ☐ A través de un buscador.
- ☐ Navegando por la red.
- ☐ Otros

Opinión sobre nuestra página web

Escriba aquí su opinión...

Tiene alguna sugerencia...

Escriba aquí sus sugerencias...

Ejercicio 4

Formulario sencillo

Usuario

Introduce Contraseña

Repite Contraseña

Deportes practicados: ☒ Futbol ☐ Baloncesto ☐ Ciclismo

Ejercicio 5

Información sobre el producto

Datos básicos	
Nombre	<input type="text"/>
Descripción	<input type="text"/>
Foto	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
<input type="checkbox"/> Añadir contador de visitas	
Datos económicos	
Precio	<input type="text"/> € Impuestos <input type="button" value="4%"/>
Promoción	
<input checked="" type="radio"/> Ninguno	
<input type="radio"/> Transporte gratuito	
<input type="radio"/> Descuento 5%	

Ejercicio 6

Ejemplo de formulario: Matriculación en el ASIR					
Nombre:	<input type="text"/>	Apellidos:	<input type="text"/>	Sexo:	<input type="radio"/> M <input type="radio"/> H
Fecha de nacimiento:	dia <input type="text"/>	mes <input type="text"/>	año <input type="text"/>		
Escoge los módulos en los que te matriculas:					
<input type="checkbox"/> Fundamentos de Hardware	<input type="checkbox"/> Servicios de Red e Internet				
<input type="checkbox"/> Gestión de Bases de Datos	<input type="checkbox"/> Administración de sistemas Operativos				
<input type="checkbox"/> Implantación de Sistemas Operativos	<input type="checkbox"/> Implantación de Aplicaciones Web				
<input type="checkbox"/> Planificación y Administración de Redes	<input type="checkbox"/> Administración de Sistemas Gestores de Bases de Datos				
<input type="checkbox"/> Lenguajes de Marcas y Sistemas de Gestión de Información	<input type="checkbox"/> Seguridad y Alta Disponibilidad				
<input type="checkbox"/> Formación y Orientación Laboral	<input type="checkbox"/> Empresa e Iniciativa Emprendedora				
<input type="checkbox"/> Proyecto de Administración de Sistemas Informáticos en Red	<input type="checkbox"/> Formación en Centros de Trabajo				
Estudios previos:					
<input type="text"/>					
<input type="button" value="Enviar"/> <input type="button" value="Restablecer"/>					
Pulsa el botón de "enviar" para formalizar la matricula o el botón de "restablecer" para limpiar el formulario.					

Ejercicio 7

iDESWEB
Formulario de registro
Los campos marcados con un asterisco son obligatorios

¿Quién eres?

Nombre * Apellidos *
Correo electrónico *
Fecha de nacimiento dd/mm/aaaa

¿De dónde eres?

Ciudad
Codigo postal
País

¿Cómo quieres iniciar sesión?

Nombre de usuario * (letras del alfabeto inglés y números, mínimo 4 caracteres)
Contraseña * (letras del alfabeto inglés, números, guión "-", subrayado "_", punto ".", mínimo 8 caracteres)
Vuelve a escribir la contraseña *

Condiciones de registro

☐ Deseo recibir ofertas de iDESWEB ☐ Una vez al día ☐ Una vez a la semana ☐ Una vez al mes
☐ Acepto el acuerdo de servicios, la declaración de privacidad y la declaración de uso de cookies

Ejercicio 8

Application form

Full Name

Email

Gender

☒ Male ☐ Female

Nationality

Skills

☒ Asp.Net ☒ jQuery
☐ MVC ☒ Sql Server

Post applying for

☐ Trainee ☒ Software Engineer
☐ Team Leader ☐ Project manager

Upload Resume:

banner.png

Paste your resume

☒ Notify me the status via Email
☒ I agree to terms & conditions

Ejercicio 9

Formulario de registro

¿QUIÉN ERES?

Nombre

Apellidos

Correo electrónico

Fecha de nacimiento
Día Mes Año

¿DE DÓNDE ERES?

Ciudad

Código postal

País

¿CÓMO QUIERES INICIAR SESIÓN?

Nombre de usuario

Contraseña

Vuelve a escribir la contraseña

CONDICIONES DE REGISTRO

Deseo recibir ofertas de IDESWEB
☐ Una vez al día ☐ Una vez a la semana ☐ Una vez al mes

☐ Acepto el acuerdo de servicios, la declaración de privacidad y la declaración de uso de cookies

Ejercicio 10

MIS DATOS:

1. Nombre:

2. Correo electrónico:

3. Teléfono:

Dirección

1. Dirección:

2. Código postal:

3. Ciudad:

Forma de pago

1. Tarjetas de Credito

- ☐ VISA
- ☐ American Express
- ☐ Master Card

2. Número tarjeta

3. Código seguridad

4. Nombre tarjeta



Autoevaluación

Contesta las siguientes preguntas:

1. ¿Qué diferencia existe entre los Radio buttons y las casillas de verificación?
2. ¿Para qué sirve agrupar controles de un formulario?
3. ¿Cuál es la función del atributo Autocomplete?
4. ¿Cuál es el propósito del tag <LABEL>?
5. ¿Cuál es la función de la etiqueta <OPTION>?
6. ¿Qué pasa si se insertan varios botones de opción con el mismo nombre?
7. ¿Que indica el atributo disabled?
8. ¿Cuál es la función del atributo hidden?
9. ¿Cuál es la función del atributo pattern?
10. ¿Cuál es la diferencia entre POST y GET?