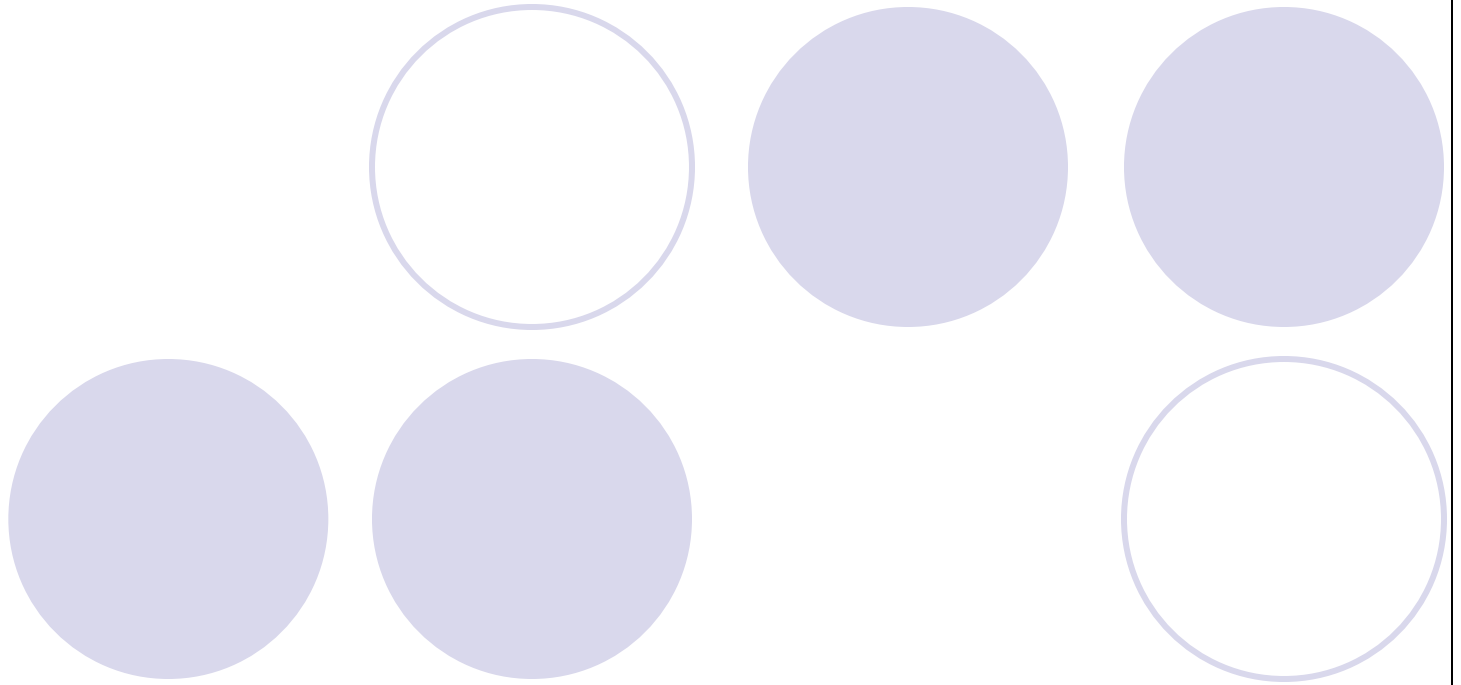


# LENGUAJES DE MARCAS



# FORMULARIOS

# Formularios WEB

- ❑ Los **formularios web** son uno de los principales puntos de interacción entre un usuario y un sitio web o aplicación .
- ❑ Su objetivo puede ser para el Servidor o para el cliente:
  - Se envían a un Servidor Web para su procesamiento y almacenamiento.
  - Se usan del lado del Cliente para provocar alguna actualización inmediata del interfaz (Con ayuda de javaScript).
- ❑ El código HTML de un **formulario web** está compuesto por uno o más **controles de formulario** (también llamados **widgets**) que son los que vamos estudiar.
- ❑ La etiqueta por la que comienza un formulario y donde cobran sentido los widgets es la etiqueta **<form>**



# `<form> ... </form>`

- ❑ La etiqueta `<form>` es una etiqueta que envuelve todo el formulario
- ❑ No es habitual ni suele tener mucho sentido poner widgets fuera de estas etiquetas.
- ❑ Normalmente es recomendable, casi obligatorio, que en un 1 html haya solo 1 FORM.
- ❑ No es imposible tener varios pero debemos saber que un botón submit solo envía 1 formulario, un botón no podría enviar todos los formularios, solo se podría hacer individualmente (Salvo que usemos javascript para hacerlo).
- ❑ El destino de un formulario se suele realizar con un botón submit pero también se puede lanzar con javascript.

# ATRIBUTOS PRINCIPALES DEL <FORM>

**<form action="destino.html" method="post" enctype="multipart/form-data">**

- ❑ **ACTION** (requerido): Este atributo se indica la ruta (normalmente relativa) del fichero destino donde se enviarán todos los datos y procesará el formulario. Se puede enviar a 2 sitios:
  - Generalmente se envía a un fichero del lado del Servidor.
  - También se puede enviar a un correo electrónico
- ❑ **METHOD**: Es la forma que utilizará el navegador para comunicarse con el servidor Web y enviarle la información del formulario. Puede tomar 2 valores:
  - POST: Se envían todos los datos del formulario en la cabecera HTTP enviada. No es visible por el usuario en la URL por lo que es más segura.
  - GET: Se envían todos los datos del formulario por la query string. Añadiendo esta con el carácter «?» después de la ruta al fichero indicado en el ACTION. Cada parámetro del formulario se separa con un carácter &.
  - Destino.html?nombre=Pablo&apellido=Garramone&telefono=611...
- ❑ **ENCTYPE**: Este atributo indica el formato en que los datos son enviados al servidor. Por defecto es application/x-www-form-urlencoded pero si queremos poder subir fichero al servidor (uso del input "file") es **obligatorio** que sea:  
**enctype=multipart/form-data**

# Controles o widgets

- ❑ Dentro de la etiqueta FORM van una serie de "controles" o widgets que son los que conforman el formulario para interactuar con el usuario y que vaya rellenándolos para ser enviados al servidor o a un correo.
- ❑ Los más importantes son:
  - INPUT (Cajas de texto, checkbox, radios, hidden, botones...)
  - SELECT (Desplegables)
  - TEXTAREA (Cajas de texto extenso)
  - BUTTON (Otra forma de hacer botones alternativa al input)
- ❑ Todas ellas (excepto los botones) suelen tener un texto que indica qué hay que rellenar en ellos, es recomendable que dicho texto también esté entre etiquetas. En este caso se utiliza la etiqueta LABEL.
- ❑ **Ejemplo:**      Nombre

*En este caso "Nombre" debería estar entre etiquetas LABEL y la cajita de texto es el INPUT.*

# <label>...</label>

- ❑ Nos permite definir una etiqueta para un control de un formulario HTML.
- ❑ Esta asociación mejora la usabilidad del formulario.
- ❑ Cuando un usuario hace clic en una etiqueta, el foco pasa automáticamente al control de formulario asociado o si es un checkbox o radio, si pulsamos sobre el texto del label automáticamente se selecciona sin necesidad de tener que pulsar sobre ellos directamente.
- ❑ Hay dos formas de realizar el LABEL y asociarlo a su INPUT:

## 1. Encapsulando el INPUT dentro del LABEL.

```
<label>NOMBRE :  
  <input type="text" name="nombre">  
</label>
```

## 2. Dejándolos separados asociando ambos a través del atributo FOR en el LABEL y poniendo el mismo valor en un ID en el INPUT.

```
<label for="nombre">NOMBRE :</label>  
<input type="text" id="nombre" name="nombre">
```

# <input>

- ❑ La etiqueta <input> indica a los usuarios que han de introducir datos en el formulario para transmitirlos al servidor Web.
- ❑ Hay muchos tipos específicos de esta etiqueta (Texto, contraseñas, hidden, subida de ficheros, checkbox, radios, números, email, url, fechas...) es necesario elegir el tipo de etiqueta input (mediante el atributo **TYPE**) más adecuado y colocarlo en el lugar oportuno.
- ❑ Es un etiqueta vacía
- ❑ En todos los <input> el atributo **NAME** es obligatorio si se desea poder acceder al mismo de forma sencilla. Como hemos visto si asociamos la etiqueta LABEL con el INPUT el atributo **ID** también será necesario.
- ❑ En conclusión para realizar un INPUT deberemos añadir necesariamente y como mínimo el TYPE, NAME y en ocasiones ID. Habrán más atributos posibles dependiendo del TYPE elegido, que los veremos a continuación.

# <input>. Atributes Comunes

- Todos los input tienen una serie de atributos comunes y otros particulares, en esta transparencia veremos los comunes más usuales:
- ❑ **READONLY:** Atributo booleano. el usuario no puede modificar el valor de entrada, pero este se envía con el resto de los datos del formulario como un valor más.
- ❑ **DISABLED:** El valor de entrada no se puede modificar y nunca se envía con el resto de los datos del formulario, es como si no existiera pero se puede ver.
- ❑ **PATTERN:** Se utiliza para propósitos de validación. Usa expresiones regulares para personalizar reglas de validación.
- ❑ **AUTOFOCUS:** Atributo booleano. Indica que un control de formulario debería recibir el foco tan pronto como la página sea cargada. Sólo un campo de formulario puede tener este atributo en una página dada.
- ❑ **FORM:** Permite asociar al formulario un control que esté fuera de las etiquetas FORM. Para ello debemos poner como valor de este atributo el ID del formulario.



# INPUT. TYPE TEXT

- ❑ Es el control o widget de formulario más básico y el más usado, y permite al usuario introducir cualquier tipo de dato, normalmente un texto.
- ❑ Su aspecto será un rectángulo horizontal donde por defecto se podrá escribir sobre él cuando reciba el foco del formulario.
- ❑ Es el valor por defecto que tiene TYPE, es decir si ponemos un INPUT sin el atributo TYPE, o el type que indicas no es reconocido o no existe, el navegador entenderá que queremos ponerle TEXT.
- Código HTML: `<input type="text" name="nombre">`
- Visualización en el Navegador:

# INPUT. TYPE TEXT. Atributos

- ❑ **PLACEHOLDER:** Se trata de un texto que aparece dentro de la caja de entrada de texto y que se usa para describir brevemente el propósito de la caja de texto. Pero el navegador no lo considera como valor, lo sigue considerando como vacío, y en cuando rellenamos un valor se elimina.

```
<label>NOMBRE:  
  <input type="text" name="nombre" placeholder="Introduce tu nombre. Ej: Pablo Garramone">  
</label>
```

NOMBRE:

- ❑ **SIZE:** El tamaño visible de caracteres que se pueden ver en la caja de texto.
- ❑ **MAXLENGTH:** Máximos de caracteres que caben en el campo.
- ❑ **MINLENGTH:** Mínimo de caracteres que debe rellenar el usuario en el campo.
- ❑ **VALUE:** Le da valor al control. En un formulario vacío para rellenar no se debe poner este atributo, ya que es el usuario el que le dará valor rellenando el mismo. Pero hay casos que sí se utiliza, en formularios que son rellenados por una base de datos o cuando queramos que el control tenga ciertos valores por defecto ya inicialmente. No se debe confundir con PLACEHOLDER, el VALUE sí se envía con el botón submit.

# INPUT. TYPE PASSWORD

- ❑ Es el mismo control que un input text con la diferencia que el navegador no muestra los caracteres y en su lugar pone puntos o asteriscos (depende del navegador).
- ❑ Es importante especificar que esto no aporta ninguna seguridad ni encriptación a los datos, se envían en texto plano, la única seguridad que aporta es que una persona, que este junto al usuario mientras escribe en esa caja de texto, no podrá ver el contenido.

- Código HTML: 

```
<label>PASSWORD:
    <input type="password" name="pass" >
</label>
```

- Visualización en el Navegador: PASSWORD:

# INPUT. TYPE NUMBER

- ❑ También es igual que un type text. Con html5 apareció este interesante type que nos permite poner una caja de texto que solo admite números.
- ❑ Si el usuario pone otra cosa que no sea número cuando debemos al botón submit no dejará seguir.
- ❑ Al control se le añade automáticamente 2 flechitas que por defecto aumentarán o reducirán el número de 1 en 1.

- Código HTML:

```
<label>EDAD:
  <input type="number" name="edad" >
</label>
```
- Visualización en el Navegador: EDAD:

# INPUT. TYPE NUMBER. Atributos

- ❑ MIN: Mínimo número que se puede indicar.
- ❑ MAX: Máximo número que se puede indicar.
- ❑ STEP: Salto que hace el número cada vez que le damos a una flechita del control. Por defecto irá de 1 en 1.

# INPUT. TYPE FECHA

- ❑ **DATE.** Otro control de html5 , type que nos permite poner una fecha y tiene adjunto un calendario para facilitar poner la fecha. Para ello usaremos el type DATE
- ❑ **DATETIME-LOCAL.** Si además queremos poner la hora entonces en vez de poner type="date" usaremos type="datetime-local".

- Código HTML:

```
<p>
  <label>
    Fecha y hora local:
    <input type="datetime-local" name="fechayhora">
  </label>
</p>
<p>
  <label>
    Fecha de Nacimiento:
    <input type="date" name="fechaNac">
  </label>
</p>
```

- Visualización en el Navegador:

Fecha y hora local:

Fecha de Nacimiento:



A calendar widget for date selection. At the top, it shows the month and year 'octubre de 2022' with left and right navigation arrows. Below this is a grid of days. The days of the week are abbreviated: 'lun', 'mar', 'mié', 'jue', 'vie', 'sáb', 'dom'. The dates are arranged in rows. The date '13' is highlighted with a grey background. Other dates are in red. The grid shows dates from 26 to 31 for the current month, and 1 to 6 for the next month.

lun	mar	mié	jue	vie	sáb	dom
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

# INPUT. TYPE FECHA. Atributos

- ❑ **MIN:** Mínima fecha que se puede indicar.
- ❑ **MAX:** Máximo fecha que se puede indicar.
- ❑ **STEP:** Salto que hace que solo se puedan elegir días con el número indicado, por defecto es 1. Ejemplo con step="5":

Fecha de Nacimiento:



# INPUT. TYPE RANGE

- ❑ **RANGE.** Otro control de html5, un control deslizante, este type permite al usuario seleccionar un valor a partir de una serie de valores o rango. Normalmente, se muestra como una barra deslizable (slider)
- ❑ Es muy recomendable utilizar los atributos **MIN, MAX y STEP**, que establecen los valores mínimo, máximo y de incremento, respectivamente.
- ❑ El valor por defecto del range es el valor medio del rango indicado.
- ❑ Su diferencia con un NUMBER es que la idea es poner un número aproximado y no importar concretar. Sin embargo, se puede usar javascript para poder visualizar el número de la barra deslizante.

<p>Escoge un precio máximo:</p>

```
<input type="range" name="barra" min="50000" max="500000" step="100"
      oninput="numero.value = this.value">
```

```
<input type="number" name="numero" min="50000" max="500000" step="100"
      value="250000" oninput="barra.value = this.value">
```

Escoge un precio máximo:





# INPUT. TYPE RADIO

- ❑ Es el control de opción, nos permite elegir entre varias opciones 1 de ellas. Por lo que es exclusivo, si se elige una no se puede elegir ninguna más.
- ❑ Para conseguir esa exclusividad hay que cumplir 1 requisito. TODOS los radio del mismo grupo a elegir deben tener el mismo valor en su atributo NAME.
- ❑ Si queremos poder elegir varios, se debe usar el checkbox no el radio.

- Código HTML:

```
<p>
  <label>Sexo:</label>
  <label>
    Hombre<input type="radio" name="sexo" value="H">
  </label>
  <label>
    Mujer<input type="radio" name="sexo" value="M">
  </label>
</p>
```

- Visualización en el Navegador:

Sexo: Hombre ☐ Mujer ☐

# INPUT. TYPE RADIO. Atributos

- ❑ **VALUE:** Es necesario el atributo VALUE porque es lo que se enviará al servidor cuando quede seleccionado.
- ❑ **CHECKED:** Atributo booleano no se pone valor, que deja marcado ese radio por defecto. Solo podemos marcar 1 de los radio con checked.

# INPUT. TYPE CHECKBOX

- ❑ Es el control de verificación, se suele usar de dos formas, o individualmente para verificar si queremos algo o no, o bien elegir entre varias opciones y poder elegir una o varias.
- ❑ En este caso el NAME si es 1 solo se le pone cualquier valor, y si tiene más de una opción hay varias opciones, mandarlo como array de todo lo elegido, por lo que se puede poner el mismo nombre (incluso a veces con corchetes) en este caso es necesario el atributo VALUE. O bien un name distinto a cada uno, en este caso se puede omitir el VALUE.
- ❑ Aunque pongas el mismo nombre no ocurre como el radio, un checkbox nunca es exclusivo.

- Código HTML:

- Visualización en el Navegador:

Deportes: Fútbol ☐ Baloncesto ☐ Judo ☐ Tenís ☐

```
<p>
  <label>Deportes:</label>
  <label>
    Fútbol<input type="checkbox" name="futbol" value="FUT">
  </label>
  <label>
    Baloncesto<input type="checkbox" name="baloncesto" value="BAL">
  </label>
  <label>
    Judo<input type="checkbox" name="judo" value="JUD">
  </label>
  <label>
    Tenís<input type="checkbox" name="tenia" value="TEN">
  </label>
</p>
```

# INPUT. TYPE CHECKBOX. Atributos

- ❑ **VALUE:** Es necesario el atributo VALUE porque es lo que se enviará al servidor cuando quede seleccionado.
- ❑ **CHECKED:** Atributo booleano no se pone valor, que deja marcado ese checkbox por defecto. Solo podemos marcar 1 de los radio con checked.

# INPUT. TYPE FILE

- ❑ Hay un último tipo de `<input>`: el tipo entrada de archivo. Los formularios pueden enviar archivos a un servidor.
- ❑ Esto no sirve de nada si no hay un servidor, la parte cliente se limita a recoger una fichero o imagen del PC local y cargarla en el formulario para su envío, pero su gestión y almacenamiento la hace el servidor.
- ❑ Tienes 2 atributos opcionales:
  1. **ACCEPT**: Para restringir los tipos de archivos que se aceptan
  2. **MULTIPLE**: Atributo booleano para poder subir más de 1 fichero

- Código HTML:

```
<p>  
  <label>  
    <input type="file" name="file" accept="image/*" multiple>  
  </label>  
</p>
```

- Visualización en el Navegador:

Examinar... No se han seleccionado archivos.

# INPUT. BOTONES

- ❑ Tenemos 2 formas de hacer botones con la etiqueta INPUT y con la etiqueta BUTTON.
- ❑ Con los INPUT tenemos 3 tipos.
  1. **Type SUBMIT:** Solo 1 por formulario. El botón más importante del formulario envía automáticamente todos los datos que haya en el formulario (dentro del FORM) al fichero de destino indicado en el ACTION de la etiqueta FORM. En otras palabras, se puede decir que el ACTION es el HREF del botón submit.
  2. **Type RESET:** Solo 1 por formulario. Botón especial que borra automáticamente todo el formulario entero de lo que haya rellenado el usuario.
  3. **Type BUTTON:** Puede haber todos los que quieras. Botón programable para que el desarrollador le de la función que desee por lo que se debe programar, sin JavaScript no tiene utilidad alguna.

# INPUT. BOTONES. Atributos

- ❑ Los atributos fundamentales son:
- ❑ **TYPE**: Los que hemos indicado. SUBMIT, RESET o BUTTON
- ❑ **VALUE**: Es el valor que dará nombre al botón
- ❑ **NAME**: Es interesante ponerlo para que el servidor pueda acceder y detectar de qué botón y/o pantalla hemos venido.

- Código HTML:

```
<p>  
  <input type="submit" name="enviar" value="ENVIAR">  
  <input type="reset" name="eliminar" value="BORRAR">  
  <input type="button" name="boton1" value="PROGRAMAME">  
</p>
```

- Visualización en el Navegador:



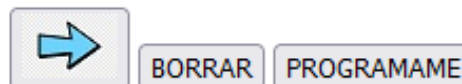
# BUTTON. BOTONES

- ❑ Existe otra forma de hacer botones usando la etiqueta BUTTON
- ❑ Es exactamente igual que los botones INPUT vistos anteriormente, con la diferencia que BUTTON no es una etiqueta atómica. En este caso no tiene el atributo VALUE y el nombre del botón se pone con el valor que pongamos de contenido. La ventaja con los botones INPUT es que estos botones puedes incorporar imágenes y no solo texto.

- Código HTML:

```
<p>  
  <button type="submit" name="enviar">  
      
  </button>  
  <button type="reset" name="eliminar">BORRAR</button>  
  <button type="button" name="boton1">PROGRAMAME</button>  
</p>
```

- Visualización en el Navegador:





# DISEÑO del FORMULARIO

- ❑ Mientras no veamos CSS tenemos ciertas estrategias de crear un formulario usando estructuras o etiquetas HTML.
- ❑ Hay que recordar que todas las etiquetas de un formulario (INPUT, TEXTAREA, SELECT...) son todas etiquetas en línea, por lo que si no hacemos nada todo se pondrá en la misma línea.
- ❑ Vamos a poner las más interesantes:
  1. Con etiquetas BR. Se trataría de separar cada uno de los label y controles del formulario a base de etiquetas BR.
  2. Con etiquetas P. Ir poniendo cada control con su label dentro de un P.
  3. Usando una estructura LISTA. Cada componente en un LI.
  4. Usando una estructura TABLA. Ir colocando los componente en las filas y celdas de la tabla

# Ejercicio 1

- ☐ Realiza este formulario, con las opciones por defecto y nombre y clave obligatoria.
- ☐ Clave 4 caracteres obligatoriamente.
- ☐ El botón submit debe ser una imagen

**Hoteles El Paraiso**

**Sistema de reserva de habitaciones:**

Introduzca su nombre:

Introduzca una clave para su identificación (4 letras):

Tipo de habitación

☐ Individual ☒ Doble

Opciones extra:

☒ Desayuno en cama ☐ Jacuzzi ☐ Climatizador

**¡ENVÍEME!**

# TEXTAREA

- ❑ Aquí vemos otra etiqueta del formulario que se sale de los INPUT. El textarea es lo mismo que un input text pero en este caso es para rellenar una importante cantidad de texto, no como el type text que es para una o unas pocas palabras.
- ❑ Podemos definir el tamaño de la cara con los atributos **ROWS** y **COLS**. Son atributos solamente visuales, para indicar filas y columnas visibles.
- ❑ No admite el atributo VALUE, si deseamos darle valor por defecto debemos colocarlo como contenido. Ojo, que espacios y tabuladores los coge como valor (no obstante, no se suele poner valor por defecto).

- Código HTML:

```
<p>  
  <label>Opinión:<br>  
    <textarea name="opinion" cols="20" rows="5">Valor por defecto, no confundir con placeholder  
  </textarea>  
</label>  
</p>
```

- Visualización en el Navegador:

Opinión:

Valor por defecto, no  
confundir con  
placeholder

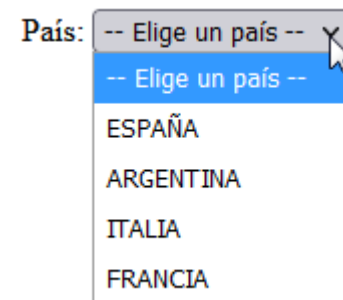
# SELECT. DESPLEGABLES

- ❑ Otro control que no es INPUT y que es uno de los más usados en cualquier formulario, el desplegable. Se utiliza la etiqueta SELECT, OPTION para cada opción del desplegable y OPTGROUP(el único opcional) para formar grupitos en las opciones.
- ❑ Es un control de selección, que puede actuar exclusivo como el RADIO o de múltiple opción como el CHECKBOX. Por lo que es un potente control.
- ❑ SELECT tiene varios atributos
  1. **NAME:** Como todo control, para acceder a él.
  2. **SIZE:** Por defecto 1. Indica la cantidad de opciones del desplegable que son visibles. A menos que sea un desplegable múltiple no se suele usar.
  3. **MULTIPLE:** Atributo booleano que indica que podemos elegir más de 1 opción, si lo usamos estamos haciendo que el desplegable actúe como un CHECKBOX en caso contrario actúa como un RADIO (su forma más común).

# OPTION. DESPLEGABLES

- ❑ Dentro de un SELECT solo puede haber OPTION, debe haber obligatoriamente al menos 1.
- ❑ Debemos diferenciar entre el VALUE y el CONTENIDO, lo que pongamos en el contenido es lo que se muestra en pantalla, el value es lo que se envía realmente como valor del control. En caso que en el option no se ponga value, el navegador usa como VALUES lo que haya en el CONTENIDO de ese OPTION. Por esta razón no es posible poner contenido vacío dentro de un OPTION.
- ❑ SELECTED: Atributo booleano que marca el valor por defecto que debe tener el desplegable (hace lo mismo que checked en el radio o checkbox)

```
<p>  
  <label>País:  
    <select name="pais">  
      <option value="">-- Elige un país --</option>  
      <option value="ESP">ESPAÑA</option>  
      <option value="ARG">ARGENTINA</option>  
      <option value="ITA">ITALIA</option>  
      <option value="FRA">FRANCIA</option>  
    </select>  
  </label>  
</p>
```



# OPTION. Opción vacía

- ❑ Si queremos dejar una opción vacía, tenemos varias opciones. El value siempre será "". Pero dentro del contenido debemos poner algo:
  1. Poner un texto para que el usuario indique algún valor, generalmente ponemos rayitas delante y detrás.
  2. Poner un espacio de html con su codificación &nbsp;
  3. Utilizando javascript
- ❑ Con cualquier de las opciones podemos usar también atributos booleanos en el option como **disabled** o **hidden**, para que la opción solo aparezca inicialmente. El problema es que cuando rellenemos el desplegable ya no podremos dejarlo otra vez vacío.

# OPTGROUP. Agrupaciones

- ❑ Es una mera cuestión estética para organizar mejor el desplegable.
- ❑ Utiliza el atributo label para indicar el nombre del grupo.
- ❑ Simplemente debemos ir agrupando OPTION dentro del OPTGROUP.

```
<p>  
  <label>País:  
    <select name="pais">  
      <option value="" hidden>&nbsp;</option>  
      <optgroup label="Europa">  
        <option value="ESP">ESPAÑA</option>  
        <option value="ITA">ITALIA</option>  
        <option value="FRA">FRANCIA</option>  
      </optgroup>  
      <optgroup label="América">  
        <option value="ARG">ARGENTINA</option>  
        <option value="USA">USA</option>  
      </optgroup>  
    </select>  
  </label>  
</p>
```



# <FIELDSET> ... </FIELDSET>

- ❑ Es una etiqueta estética, que sirve para agrupar partes y componentes del formulario que tengan alguna relación semántica o con algún criterio o relacionados de forma lógica.
- ❑ El navegador dibuja una caja alrededor del conjunto de campos para indicar que están relacionados.
- ❑ Se le puede añadir un pequeño texto que haga de título de la sección que el navegador lo coloca en la parte de arriba de la caja, usando la etiqueta LEGEND.

```
<fieldset>
<legend>Personal Data:</legend>
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname"><br><br>
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname"><br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>
<label for="birthday">Birthday:</label>
<input type="date" id="birthday" name="birthday"><br><br>
</fieldset>
<fieldset>
<legend>Actions:</legend>
<input type="reset" value="Delete">
<input type="submit" value="Submit">
</fieldset>
```

Personal Data:

First name:

Last name:

Email:

Birthday:

Actions:



# EJERCICIO 2

## Información sobre el producto

Realiza este formulario llevándolo a otro html que también tendrás que hacer, usa método get para ver por la query string todos los valores que llenas.

- Campos nombre (*mínimo 5 caracteres máximo 50*) y precio obligatorios.
- Usa atributos cols y rows

<b>Datos básicos</b>	
Nombre	<input type="text"/>
Descripción	<input type="text"/>
Foto	<input type="text"/> <input type="button" value="Examinar..."/> No se han seleccionado archivos.
<input type="checkbox"/> Añadir contador de visitas	
<b>Datos económicos</b>	
Precio	<input type="text"/> €
Impuestos	<input type="text" value="4%"/>
Promoción	<input checked="" type="radio"/> Ninguno <input type="radio"/> Transporte gratuito <input type="radio"/> Descuento 5%

# DATALIST y LIST. DESPLEGABLES

- ❑ HTML5 añadió otro control parecido al desplegable que también utiliza la etiqueta OPTION y acaba asociando el control a otro control type text normalmente para sugerir opciones al usuario.
- ❑ El elemento <datalist> es específico de formularios. Se usa para construir una lista de ítems que luego, con la ayuda del atributo **LIST**, será usada como sugerencia en un campo del formulario.

```
<datalist id="datosCursos">
  <option value="1DAM" label="Curso - 1º DAM">
  <option value="2DAM" label="Curso - 2º DAM">
  <option value="1DAW" label="Curso - 1º DAW">
  <option value="2DAW" label="Curso - 2º DAW">
</datalist>

<p>
  <label for="curso">Curso:</label>
  <input type="text" name="curso" id="curso" list="datosCursos">
</p>
```

Curso:

- Curso - 1º DAM
- Curso - 2º DAM
- Curso - 1º DAW
- Curso - 2º DAW

- ❑ Es importante observar que lo que se muestra como sugerencia es el valor del atributo label del elemento <option>, pero lo que se carga en el campo al seleccionar la sugerencia es el valor del atributo value.

# OTRAS ETIQUETAS HTML5

HTML5 introdujo otras etiquetas nuevas (quizá menos útiles) además de algunas de las ya mencionadas (más relevantes), que pueden resultar interesantes en alguna ocasión.

- ❑ **Type="email"**: Es lo mismo que un input text con la salvedad que nos obliga a tener un carácter @ en medio de 2 palabras.
- ❑ **Type="tel"**: Se supone que es para validar números de teléfono pero no realiza ninguna restricción, solo es interesante para ciertos dispositivos táctiles que al ver este tipo despliegan el teclado numérico.
- ❑ **Type="url"**: Valida que pongas una dirección absoluta con el protocolo incluido. (Ej: http:// o https://). Algunos dispositivos llegan a incluir estos caracteres automáticamente.
- ❑ **Type="color"**: Despliega una ventana con una paleta de colores para elegir uno de ellos.
- ❑ **Type="search"**: Es igual que un input text, la única diferencia es que algunos navegadores añaden un botoncito "x" al final de la caja para limpiar el campo. Y que tiene autocompletado de búsquedas anteriores.