UNIDAD 5. HTML 5

# HTML 5

#### Introducción

Las nuevas tendencias de la Web 2.0. invitan a que los usuarios tengan mucho más peso específico a la hora interactuar con los elementos informáticos que, antaño, solo estaban al alcance de los profesionales del tema. Pero, cuando el usuario es el centro de atención, se necesitan interfaces o entornos que le ofrezcan toda “la comodidad” posible a la hora de actuar con ellos, es decir, deben ser sencillos, muy intuitivos y prácticos además de cuidar los aspectos estéticos necesarios para potenciar su atractivo.

Un ejemplo claro de aplicación de lo expuesto en el párrafo anterior son los formularios HTML ya que permiten recoger información dentro de una página web rellenando campos o pulsando botones además de dar la posibilidad de enviarla a una dirección de correo o a un script dinámico escrito en php, asp o cgi.

El estándar HTML5 añade características nuevas para la realización de formularios mejorando sustancialmente los de las anteriores versiones HTML y presenta la novedad de la validación de datos por parte del navegador o del lado cliente o lo que simplifica en gran manera la creación y uso de formularios.

Los formularios en HTML5 no han cambiado mucho ya que mantiene la misma estructura pero se han agregado nuevos elementos, tipos de campo y atributos que les dan mayor potencia y mucha más sencillez a la hora de trabajar con ellos.

Un ejemplo que constata todo lo anterior es el elemento input el cual ha sido simplificado en su sintaxis pero ampliado en funcionalidades.

*La especificación oficial del consorcio W3C sobre formularios se encuentra en la dirección:*

[***http://www.w3.org/TR/html5/forms.html***](http://www.w3.org/TR/html5/forms.html)



En la siguiente tabla tenemos los elementos HTML5 relacionados con los formularios

|  |  |
| --- | --- |
| **Elemento** | **Descripción** |
| [<form>](http://www.w3schools.com/tags/tag_form.asp) | Define un formulario HTML |
| [<input>](http://www.w3schools.com/tags/tag_input.asp) | Define controles para el formulario |
| [<textarea>](http://www.w3schools.com/tags/tag_textarea.asp) | Define un área de texto |
| [<label>](http://www.w3schools.com/tags/tag_label.asp) | Define una etiqueta para un elemento <input> |
| [<fieldset>](http://www.w3schools.com/tags/tag_fieldset.asp) | Grupos de elementos relacionados en un formulario |
| [<legend>](http://www.w3schools.com/tags/tag_legend.asp) | Define un título para un elemento <fieldset> |
| [<select>](http://www.w3schools.com/tags/tag_select.asp) | Define una lista desplegable |

|  |  |
| --- | --- |
| [<optgroup>](http://www.w3schools.com/tags/tag_optgroup.asp) | Define un grupo de opciones relacionadas en un lista desplegable |
| [<option>](http://www.w3schools.com/tags/tag_option.asp) | Define una opción en una lista desplegable |
| [<button>](http://www.w3schools.com/tags/tag_button.asp) | Define un botón |
| [<datalist>](http://www.w3schools.com/tags/tag_datalist.asp) | Especifica una lista de opciones predefinidas para controles <input> |
| [<keygen>](http://www.w3schools.com/tags/tag_keygen.asp) | Define un campo para generar pares de claves para sistemas de gestión de certificados |
| [<output>](http://www.w3schools.com/tags/tag_output.asp) | Define el resultado de un cálculo |

  Nuevo en HTML5

# REGLAS BÁSICAS

La recogida y envío de datos en un formulario se hace a través de elementos o controles como botones, listas desplegables, etc.

El *nombre de elemento* de cualquier componente de un formulario viene indicado por el atributo **name**.

Cada elemento tiene dos posibles valores: uno inicial y otro actual, que siempre son cadenas de caracteres. El valor inicial se asigna con el atributo **value** y el valor actual, que inicialmente tiene el valor inicial, es aquel que el usuario introduce bien escribiendo, marcando casillas o pulsando botones.

Cuando la información de un formulario se envía a un fichero indicado en el atributo **action** se hace a través de un tipo especial de botón denominado “submit”. Dependiendo de la forma de envío, especificada en el atributo **method**, la información de un formulario puede ir en la línea de llamada al archivo, o sea, en la barra de navegación (method get) o no ser mostrada (method post).

#### Tipos de controles

* + - * **Botón**

Control que al ser pulsado o activado desencadena una acción. Los tipos de botones son:

* + - * + *Botón de envío (submit button)*: Cuando se pulsa o activa se envía la información del formulario.
        + *Botón de reset (reset buttom)*: Cuando se pulsa o activa este botón se asignan los valores iniciales a todos los elementos del formulario.
        + *Botón normal*: Cuando se pulsa o activa normalmente puede ejecutar uno o más scripts a través del atributo **event**.

#### - Casillas de verificación (checkbox)

Controles que permiten seleccionar cero o más opciones de un número limitado de ellas.

Una casilla de verificación está “seleccionada” cuando se activa el atributo **checked**

correspondiente a la casilla.

También hay que tener en cuenta que, cuando se envía el formulario, solamente se manda la información de las casillas de validación seleccionadas y que varias casillas pueden tener el mismo nombre de control.

#### Botones de opción (radio button)

Son controles que funcionan como las casillas de verificación a excepción de que, cuando se les asigna el mismo nombre de control, sólo puede estar uno de ellos activado.

#### Menús

Son controles que ofrecen valores entre los que se puede elegir.

#### Entradas de texto (text input)

Son controles que permiten introducir textos. Existen dos tipos:

* + - * + *De una sola línea*: A través del elemento **input** se crea este tipo de control.
        + *De varias líneas*: A través del elemento **textarea** se crea este tipo de control.

En ambos casos el valor actual del control es el texto introducido.

#### Selección de ficheros (file select)

Son controles que permiten escoger archivos de forma que la información de dichos ficheros pueda ser enviada con el formulario.

#### Controles ocultos (hidden controls)

Son controles que no se ven pero cuyos valores actuales pueden ser enviados con el formulario. Normalmente el usuario que introduce datos en el formulario desconoce la existencia de estos controles y se suelen usar para pasar información adicional al script o programa que va a procesar la información del formulario.

#### Controles de tipo objeto (object controls)

Son controles que pueden insertar objetos genéricos en los formularios de modo que los valores asociados se envían junto con la información del formulario.

# ELEMENTO <FORM>

Un formulario no es ni más ni menos que un conjunto de campos a través de los cuales el usuario introduce información que luego puede ser usada. Estos campos están representados por diferentes elementos de formulario como son los botones, casillas de chequeo, etc., que se adaptan a la forma de meter los datos y que suelen estar entre la etiqueta <form> y </form> del elemento **<form>**.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <form> | | | | | | |
| Descripción | Contiene los elementos de un formulario | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | accept-charset, action, autocomplete, enctype, method, name novalidate, target | | | | | | |
| Diferencia entre html  4.01 y html5 | autocomplete y novalidate son atributos nuevos HTML5 y accept ha sido suprimido | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| accept-charset | Mapa de caracteres | Especifica un conjunto de mapas de caracteres que será usados para el envío del formulario. En HTML4.01 este conjunto estaba separado por comas. |
| action | URL | Especifica la dirección a la que se envían los datos del formulario. |
| autocomplete | on off | Especifica si el formulario debe ser completado (on) o no (off). |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| enctype | application/x-www-form- urlencoded  multipart/form-data text/plain | Especifica la codificación de envío del formulario (solo para método post) |
| method | get post | Especifica el método de envío de los datos del formulario |
| name | Texto | Especifica el nombre del formulario |
| novalidate | novalidate | Especifica si se valida o no un formulario al enviarlo |
| target | \_blank  \_self  \_parent  \_top | Específica dónde se va a visualizar el archivo que recibe los datos. |

 Nuevo en HTML5

La estructura más básica de un formulario es:

|  |
| --- |
| <form>  .  Elementos de formulario  .  </form> |

Donde “Elementos de formulario” son controles que permiten recopilar información.

La forma anterior de usar un formulario no es muy práctica ya que normalmente los formularios sirven para que el usuario introduzca información y que ésta sea enviada a un script o fichero para ser procesada por lo que el elemento **<form>** debe tener algún atributo que permita ese envío. Así **<form>** se suele usar con los atributos method, action y enctype, siempre que method tenga valor ”post”, como mínimo.

|  |
| --- |
| <form action=”URL”>  .  Elementos de formulario  .  </form> |

En la estructura anterior, el atributo method vale “get” (valor por defecto sino se pone) por lo que no se usa el atributo enctype.

|  |
| --- |
| <form action=”URL” method=”post” enctype=”forma de codificación”>  .  Elementos de formulario  .  </form> |

En esta estructura si se usa el atributo enctype ya que method vale “post”.

En los dos ejemplos anteriores “URL”, o sea, el valor del atributo action es el lugar a donde vamos a enviar la información del formulario que puede ser o bien un script o programa php, javascript, etc., o una dirección de correo.

Cuando se envían los datos de un formulario a un script, éste debe estar construido para poder recogerlos correctamente y procesarlos. En cambio cuando se mandan a una dirección de correo se abre un cliente de correo y se envía un mensaje a la dirección especificada con el contenido del formulario.

*El desuso de los clientes de correo hace que en los formularios apenas se emplee esta forma de envío.*



Otra cuestión importante a tener en cuenta es saber qué manera de envío usar, es decir, cuando utilizar el atributo method con el valor “get” o “post”.

- **get**

* Añade los datos del formulario a la URL del atributo action poniendo los nombres de los elementos más sus respectivos valores.
* La longitud de la URL con los datos anteriores es limitada (alrededor de 3000 caracteres).
* No se debe usar esta forma de envío cuando se manden datos confidenciales ya que son visibles en la URL.
* Útil para añadir a los marcadores de los navegadores la URL destino más los datos del formulario.
* Es mejor usar este valor para el envío de datos que no necesitan seguridad como las peticiones a los buscadores de información como Google.

#### Ejemplo

<!DOCTYPE html>

<head>

<title>Formulario</title>

<meta charset="UTF-8">

</head>

<body>

<form action="actualizar.php" method="get">

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

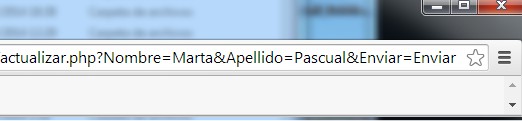
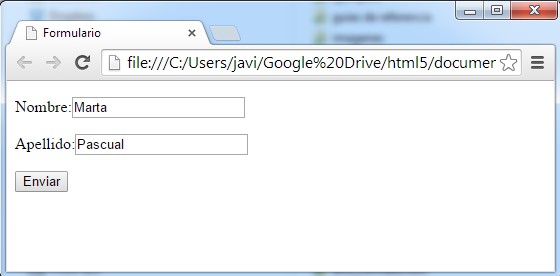
<p>Apellido:<input type="text" name="Apellido" /></p>

<p><input type="submit" name="Enviar"></p>

</form>

</body>

</html>



En este ejemplo se define un formulario que envía datos al archivo “actualizar.php” (atributo action de <form>) que se encuentra en el mismo directorio que el documento html en el que está incluido dicho formulario, que usa method=”get”, que contiene dos cuadros de texto (elementos <input> con el atributo type=”text” y name=”nombre del elemento”) con nombres “Nombre” y “Apellido” y, también, un botón para enviar los datos al archivo indicado en “action”.

Como vemos en la primera imagen aparece el código html, en la segunda como se vería en un navegador (permite al usuario introducir un nombre y un apellido y enviarlo al documento actualizar.php al pulsar el botón “enviar”) y, en la última imagen, lo que aparece en la barra de direcciones del navegador después de pulsar el botón de enviar.

Al emplear method=”get” se ve que al valor de “action” se añade un **?** y luego el nombre de los elementos igualados a su valor separados por un **&** y que, estos valores, son perfectamente visibles a la hora de cargar el archivo referido en “action” en el navegador.

- **Post**

o Inserta los datos del formulario dentro del cuerpo del envío por lo que no se añaden a la URL y, por lo tanto, no se ven.

* No hay limitación de caracteres a la hora de enviar datos.
* Los envíos con este valor no se añaden a los marcadores de los navegadores

#### Ejemplo;

<!DOCTYPE html>

<head>

|  |
| --- |
| <title>Formulario</title>  <meta charset="UTF-8">  </head>  <body>  <form action="actualizar.php" method="post">  <p>Nombre:<input type="text" name="Nombre" /></p>  <p>Apellido:<input type="text" name="Apellido" /></p>  <p><input type="submit" name="Enviar"></p>  </form>  </body>  </html> |
|  |
|  |

Basándonos en el ejemplo que hemos usado explicando el valor **get** de “method” vemos que, al usar, el valor **post**, no se ve la información en la barra de direcciones al cargar el archivo actualizar.php.

#### Ejemplos de <form>

|  |
| --- |
| <form action=”recogerdatos.php” method=”post”  enctype=”application/x-www-form-urlencoded”> |

|  |
| --- |
| .  Elementos de formulario  .  </form> |

En este ejemplo se envían los datos al archivo “recogerdatos.php” que se encuentra en el mismo directorio que la página que contiene el formulario con la codificación “application/x-www-form-urlencoded” sin añadir ningún dato a la URL de “action” (“recogerdatos.php”) debido que a “method” tiene el valor post.

|  |
| --- |
| <form action=”../recogerdatos.php” method=”get”>  .  Elementos de formulario  .  </form> |

En este ejemplo el archivo receptor del formulario está un directorio por encima del documento html que lo contiene (action=”../recogerdatos.php”) y, al usar el valor get del atributo method, se verán los valores de los elementos del formulario en la barra de direcciones del navegador al cargar dicho archivo receptor.

|  |
| --- |
| <form action=”[http://www.mentor.mec.es/cgi-bin/validar.php”](http://www.mentor.mec.es/cgi-bin/validar.php)  method=”post” enctype=”application/x-www-form-urlencoded”>  .  Elementos de formulario  .  </form> |

La única diferencia con el ejemplo anterior es que en éste el archivo al que se envía el formulario está en la estructura del servidor [http://www.mentor.mec.es.](http://www.mentor.mec.es/)

|  |
| --- |
| <form action=”<mailto:uno@uno.es>” method=”post” enctype=”text/plain”>  . |

|  |
| --- |
| Elementos de formulario  .  </form> |

En este ejemplo al tener “action” el valor de una dirección de correo, se abre el cliente de correo que esté configurado en el sistema (Outlook express, gmail, etc.) y se envía un mensaje a la dirección de correo [uno@uno.es](mailto:uno@uno.es) con el contenido del formulario (nombres de elementos igualados a sus valores). Se utiliza method=”post” para no mostrar datos y se envían estos datos como texto plano.

|  |
| --- |
| <form>  .  Elementos de formulario  .  </form> |

En este ejemplo al no tener asignado ningún valor inicial al atributo action significa que la URL de destino es el propio documento .html con method “get”. Normalmente se emplea esta forma cuando se envían los datos a un destino externo a través de script en vez de por el propio elemento <form>

|  |
| --- |
| <form>  .  Elementos de formulario  .  </form> |

En este ejemplo se ve que no hay definido un valor para “action” por lo que el fichero destinatario del formulario es el propio documento html, no hay definido un valor para method por lo que se usa su valor por defecto que es get.

|  |
| --- |
| <form action=”#”>  . |

|  |
| --- |
| Elementos de formulario  .  </form> |

En este ejemplo el valor “**#**” indica que el destino del formulario es una parte del propio documento html marcada por un ancla o enlace interno y con valor get para el atributo “method”.

#### Elemento <input>

Es el elemento más importante de los formularios ya que, a través de su atributo “type”, determina qué tipo de entrada o control es el que se le ofrece al usuario para que introduzca datos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <input> | | | | | | |
| Descripción | Elemento para poner un control en un formulario | | | | | | |
| Elemento | En línea y etiqueta vacía | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | accept, alt, autocomplete, autofocus, checked, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, step, type, value y width | | | | | | |
| Diferencia entre html  4.01 y html5 | El atributo align ha sido suprimido en HTML5 | | | | | | |

Los atributos comunes a todos los controles que se pueden crear con el elemento <input> están recogidos en la siguiente tabla

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |

|  |  |  |
| --- | --- | --- |
| autocomplete | on off | Especifica si un elemento <input> debe ser completado (on) o no (off) |
| autofocus | autofocus | Especifica si un elemento <input> será el foco de la página cuando esta se carga |
| disabled | disabled | Especifica que un elemento <input> está desactivado |
| form | *Identificador formulario* | Especifica al formulario o formularios que pertenece el elemento <input> |
| max | *número fecha* | Especifica el máximo valor del elemento  <input> |
| maxlenght | *número* | Especifica el máximo número de caracteres permitidos en un elemento <input> |
| min | *número fecha* | Especifica el mínimo valor del elemento  <input> |
| multiple | multiple | Especifica si el usuario puede introducir más de un valor |
| name | *Texto* | Especifica el nombre del elemento <input> |

|  |  |  |
| --- | --- | --- |
| pattern | *Expresión* | Especifica una expresión contra la que se chequea el valor del elemento <input> |
| placeholder | *Texto* | Especifica un breve descripción del valor del elemento <input> |
| readonly | readonly | Especifica sin un campo de entrada es de solo lectura |
| required | required | Especifica que un campo de entrada debe ser rellanado obligatoriamente antes de enviar el formulario |
| size | *Número* | Especifica el ancho, en caracteres, de un elemento <input> |
| step | *Número* | Especifica el número de intervalos para un elemento <input> |
| type | button checkbox color date datetime  datetime-local email  file hidden image month number password radio range  reset | Especifica el tipo de elemento <input> |

|  |  |  |
| --- | --- | --- |
|  | search submit tel  text time url week |  |
| value | *Texto* | Especifica el valor del elemento <input> |

 Nuevo en HTML5

*Los atributos* ***name*** *y* ***value*** *son los más importante de todos ya que sin ellos no se puede ni identificar un control ni enviar su valor.*

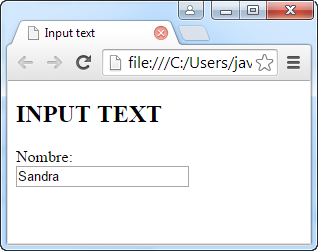


#### Controles básicos <input>

* + - * **Cuadro de texto**

Muestra un cuadro de texto para que el usuario escriba un contenido. Se inserta en un formulario con el valor “text” del atributo type.

<input type=”text” name=”nombre del control” value=”valor inicial”>



<form action="">

<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input text </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT TEXT </h2>

</body>

</html>

Nombre: <br>

<input type="text" name="Nombre">

<br>

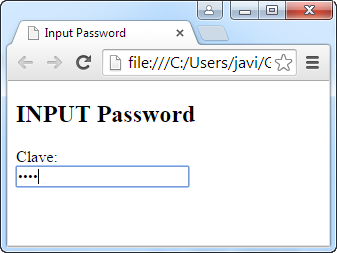
</form>

En este ejemplo el texto se introduce tal cual en el cuadro de texto.

#### Cuadro de contraseña

Muestra un cuadro de texto en el que la información que introduce el usuario se muestra como un conjunto de asteriscos o círculos. Se inserta en un formulario con el valor “password” del atributo type.

<input type=”password” name=”nombre del control” value=”valor inicial”>



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input Password </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT Password </h2>

<form action=""> Clave: <br>

<input type="password" name="Clave">

<br>

</form>

</body>

</html>

En este ejemplo el texto que se introduce en el cuadro de contraseña se convierte en “\*” pero solo a efectos de visualización ya que su valor es el que se escribe sin encriptar.

#### Casilla de verificación

Muestra varias opciones de las que se puede elegir una, varias o ninguna de ellas. Se inserta en un formulario con el valor “checkbox” del atributo type.

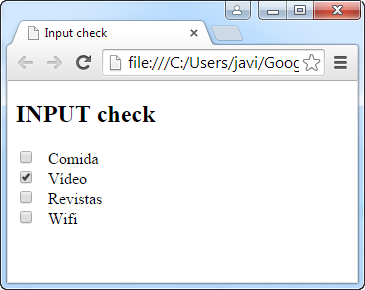
<input type=”checkbox” name=”nombre control1” value=”valor inicial1”>

<input type=”checkbox” name=”nombre control2” value=”valor inicial2”>

<input type=”checkbox” name=”nombre control3” value=”valor inicial3”>

Si se quiere dejar una opción seleccionada se le añade el atributo “checked” sin ningún valor.

<input type=”checkbox” name=”nom” value=”valor” checked>



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input check </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT check </h2>

<form action="">

&nbsp; Comida<br>

<input type="checkbox" name="video"

value="si" checked> &nbsp; Vídeo <br>

<input type="checkbox" name="revistas"

<input type="checkbox" name="comida"

value="si" >

value="si">

&nbsp; Revistas<br>

<input type="checkbox" name="wifi"

value="si"> &nbsp; Wifi<br>

</form>

</body>

</html>

En este ejemplo los elementos marcados se envían o devuelven con el valor “si”, es decir, si se marca la casilla “Vídeo”, por ejemplo, entonces al elemento “video” se le asigna el valor “si”. Si se deja alguna casilla sin marcar entonces el elemento correspondiente no se envía ni devuelve ningún valor. También la casilla “Vídeo” estaría marcada por defecto al llevar el atributo “checked”.

#### Botón de opción

Muestra varias opciones de las que sólo se puede escoger una de ellas. Se inserta en un formulario con el valor “radio” del atributo type y deben tener el mismo valor en el atributo name.

<input type=”radio” name=”nombre” value=”valor1” checked”>

<input type=”radio” name=”nombre” value=”valor2”>

<input type=”radio” name=”nombre” value=”valor3”>



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input Radio </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT Radio </h2>

<form action="">

<input type="radio" name="fumador"

value="si" >

&nbsp; Fumador<br>

<input type="radio" name="fumador"

value="no" checked> &nbsp; No fumador<br>

</form>

</body>

</html>

En este ejemplo el control se llama de la misma manera (fumador) y se envía con el valor “si” o “no” dependiendo de la casilla que haya sido activada (hay que tener en cuenta que sólo puede haber una casilla activa). La casilla correspondiente a “no fumador” es que está marcada por defecto al llevar el parámetro “checked”.

#### Botón de envío

Muestra un botón que, cuando se pulsa, envía el formulario al archivo indicado en la URL del atributo “action”. Se inserta en un formulario con el valor “submit” del atributo type.

<input type=”submit” name=”nombre” value=”enviar”>



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input Radio </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT Radio </h2>

<form action="archivo.html">

<input type="radio" name="fumador" value="si" >

&nbsp; Fumador<br>

<input type="radio" name="fumador" value="no" checked>

&nbsp; No fumador<br>

<input type=”submit” value=”enviar”>

</form>

</body>

</html>

En este ejemplo, al pulsar el botón de enviar, mandaría los datos del formulario al fichero “archivo.html” el elemento “fumador” con el valor “si” o “no” dependiendo de la opción marcada.

#### Botón de reset

Muestra un botón que, cuando se pulsa, les asigna el valor inicial a los elementos del formulario. Se inserta en un formulario con el valor “reset” del atributo type.

<input type=”reset” name=”nombre” value=”Reset”>



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Input Radio </title>

<meta charset="UTF-8">

</head>

<body>

<h2> INPUT Radio </h2>

<form action="archivo.html">

<input type="radio" name="fumador" value="si" >

&nbsp; Fumador<br>

<input type="radio" name="fumador" value="no" checked>

&nbsp; No fumador<br>

<input type=”submit” value=”enviar”>

<input type=”reset” value=”Borrar”>

</form>

</body>

</html>

En este ejemplo se agrega otro botón de tipo “reset” que borrará los datos que hubiera en el formulario y pondría los que están por defecto.

#### Botón

Muestra un botón que, cuando se produce algún evento sobre él, como ser pulsado, realiza alguna acción. Se inserta en un formulario con el valor “button” del atributo type.

<input type=”button” onclick=”alert(‘Pulsado’) value=”Pulsa el botón”> En este ejemplo se muestra el mensaje *Pulsado* con se pulsa el botón.

#### Controles nuevos HTML5 <input>

En la siguiente tabla tenemos los nuevos controles de formulario que se pueden crear con el elemento <input> en HTML5.

Los navegadores antiguos y algunos más modernos no soportan estos nuevos controles por lo que los interpretarán como un simple cuadro de texto.

También el aspecto de estos controles depende mucho del navegador variando de unos a otros.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Control** | **type** | **Explicación/ejemplo/navegadores que lo soportan** | | | | | | |
| Número | number | Contiene un valor numérico | | | | | | |
| <input type=”number” name=”número” value=”1”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | 9.0 | - | - |
| Fecha | date | Contiene una fecha | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | <input type=”date” name=”fecha” value=”2014-11-24”> | | | | | | |
|  |  |  | | | | | | |
|  |  | | | | |  |
| - | - | - | - | - |
| Color | color | Contiene la identificación de un color | | | | | | |
| <input type=”color” name=”color” value=”#00ff00”>  <input type=”color” name=”color” value=”red”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | - | - | - |
| Rango | range | Contiene un valor numérico comprendido entre un rango | | | | | | |
| <input type=”range” name=”rango” min=”1” max=”10”  value=”7”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | 9.0 | - | - |
| Mes y año | month | Contiene un mes y un año | | | | | | |
| <input type=”month” name=”mes” value=”2014-03”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | - | - | - |
| Semana y año | week | Contiene un número de semana y un año | | | | | | |
| <input type=”week” name=”semanayaño” value=”2014-W13”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | - | - | - | - | - |  |
| Hora | time | Contiene una hora | | | | | | |
| <input type=”time” name=”hora” value=”22:55”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | - | - | - |
| Fecha y hora | datetime | Contiene una fecha y una hora con zona horaria | | | | | | |
| <input type=”datetime” name=”fechayhora”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | - | - | - |
| Fecha y hora local | datetime- local | Contiene una fecha y una hora sin zona horaria | | | | | | |
| <input type=”datetime-local” name=”local”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | - | - | - |
| Correo | email | Contiene una dirección de correo | | | | | | |
| <input type=”email” name=”correo” value=”[uno@uno.es](mailto:uno@uno.es)”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | 9.0 | - | - |
| Búqueda | search | Contiene un término para campos de búsqueda | | | | | | |
| <input type=”search” name=”buscar” value=”gatos”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | . | - | - |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Teléfono | tel | Contiene un número de teléfono | | | | | | |
| <input type=”tel” name=”teléfono”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | . | - | 8.0 |
| Dirección Internet | url | Contiene una dirección de Internet | | | | | | |
| <input type=”url” name=”dirección” value=”[http://google.es](http://google.es/)”> | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
| - | - | 9.0 | - | - |

**Nota:** El símbolo “**-“**significa cualquier versión del navegador y si hay un número indica la versión a partir de la cual se reconoce el control.

#### Ejemplos completos

#### Otros elementos de formulario

Aunque el elemento <input> permite crear una gran variedad de controles de formulario, existen otros tales como listas desplegables, áreas de texto, generadores de claves, etc., que usan otros elementos diferentes.

#### - <select>

Elemento que crea una lista desplegable cuyas opciones vienen definidas por elementos

<option> que van dentro de las etiquetas de apertura y cierre de <select>.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <select> | | | | | | |
| Descripción | Elemento que crea una lista desplegable | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | autofocus, disabled, form, multiple, name, required y size | | | | | | |
| Diferencia entre html | autofocus, form y required son atributos HTML5 (\*) | | | | | | |

|  |  |
| --- | --- |
| 4.01 y html5 |  |

(\*) Estos atributos están explicados en el apartado del elemento <input>

#### <option>

Elemento que forma parte de un elemento <select> o <datalist> y define sus posibles opciones y qué valores iniciales tendrán.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <option> | | | | | | |
| Descripción | Define las ocpiones de un elemento <select> o <datalist> | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | disabled, label, selected y value | | | | | | |
| Diferencia entre html  4.01 y html5 | Ninguna | | | | | | |

#### Ejemplo de <select> y <option>

<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Estilos en tabla</title>

<meta charset="UTF-8">

<style type="text/css">

.fondoverde

{background-color:#ccffcc;}

</style>

</head>

<body class="fondoverde">

<h2> Select </h2>

<form>

<select name="origen">

<option value="Soria" selected>Soria</option>

<option value="Zaragoza">Zaragoza</option>

<option value="Burgos">Burgos</option>

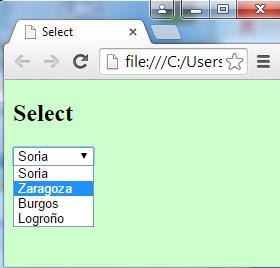
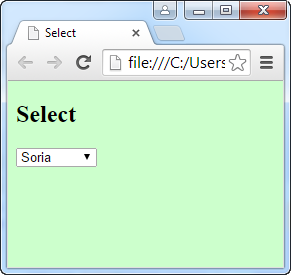
<option value="Logroño">Logroño</option>

</select>

</form>

</body>

</html>



En este ejemplo tenemos una serie de ciudades de destino de las cuales “Soria” está seleccionada y sería su valor por defecto. En cuanto se pulsa con el ratón sobre la flecha de la lista, ésta se despliega y deja elegir otro valor como sería en la segunda imagen “Zaragoza”.

#### <textarea>

Elemento que define un cuadro de texto de n-columnas por n-filas

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <textarea> | | | | | | |
| Descripción | Define un cuadro de texto | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | autofocus, cols, disabled, form, maxlengt, name, placeholder, required, rows y wrap | | | | | | |
| Diferencia entre html  4.01 y html5 | autofocus, form, maxlength, placeholder, required, wrap son atributos HTML5 | | | | | | |

#### Ejemplo:

<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Textarea</title>

<meta charset="UTF-8">

<style type="text/css">

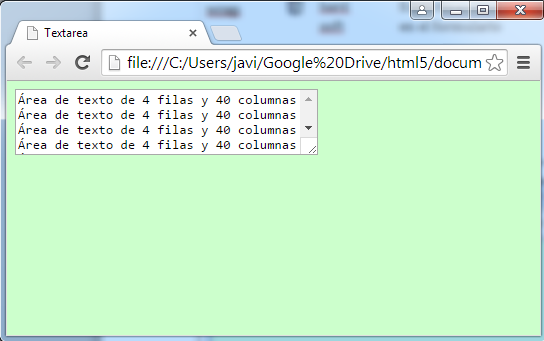
.fondoverde

{background-color:#ccffcc;

}

</style>

</head>



<body class="fondoverde">

<textarea rows="4" cols="40">

Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas Área de texto de 4 filas y 40 columnas

</textarea>

</body>

</html>

Como se ve en el ejemplo, el elemento “textarea” ha generado un área de texto donde sólo se ven 4 filas (para acceder a las demás habría que hacer scrolling) y 40 columnas.

#### <button>

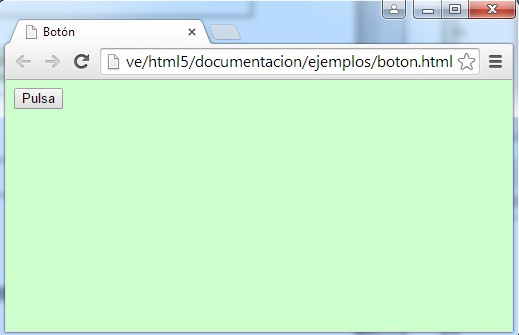
Define un botón sobre el cual se puede pulsar con el ratón.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <button> | | | | | | |
| Descripción | Define un botón para ser pulsado con el ratón | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | autofocus, disabled, form, fromaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value | | | | | | |
| Diferencia entre html  4.01 y html5 | autofocus, form, formaction, formmethod, formnovalidate, formtarget son atributos HTML5 | | | | | | |



*Dentro de un elemento <button> se puede poner texto o imágenes a diferencia de los botones creados con <input> que no lo permiten.*

#### Ejemplo:



<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Botón</title>

<meta charset="UTF-8">

<style type="text/css">

.fondoverde

{background-color:#ccffcc;

}

</style>

</head>

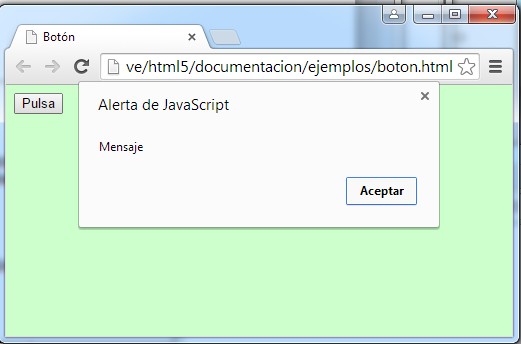
<body class="fondoverde">

<button type="button" onclick="alert('Mensaje')">Pulsa

</button>

</body>

</html>



En este ejemplo hemos creado un botón que, cuando se pulsa, saca un mensaje con el texto “Mensaje”. Como se ve el botón responde a lo que se denomina un “evento” que no es ni más ni menos que una acción que, en este caso, ha sido generada por la pulsación del botón con el ratón. Al producirse dicho “evento” abre la ventana con el mensaje.

#### <datalist>

Especifica una lista desplegable de opciones predefinidas mediante el elemento <option> para un elemento <input> con atributo list.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <datalist> | | | | | | |
| Descripción | Especifica una lista de opciones predefinidas | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| **20.0** | **4.0** | **10.0** | **9.0** | **Safari** |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | Ninguno | | | | | | |
| Diferencia  entre html  4.01 y html5 | Elemento nuevo HTML5 | | | | | | |

#### Ejemplo:

<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Datalist</title>

<meta charset="UTF-8">



<style type="text/css">

.fondoverde

{background-color:#ccffcc; }

</style>

</head>

<body class="fondoverde">

<h2> Datalist</h2>

<form>

<input list="estrellas" value="Arturo">

<datalist id="estrellas">

<option value="Mizar">

<option value="Rigel">

<option value="Sirio">

<option value="Aldebarán">

</datalist>

</form>

</body>

</html>

En este ejemplo se ve una lista desplegable que, cuando se pasa con el ratón por encima el cuadro de diálogo, aparece un icono  que permite desplegar la lista de valores posibles de un elemento <input> (en este caso son nombres de estrellas).



*El atributo* ***id*** *del elemento <datalist> debe coincidir con el valor del atributo* ***list*** *del elemento <input>*

#### <keygen>

Genera una clave privada, que es cargada localmente, y otra pública que es enviada a un servidor.

Se emplea siempre dentro de un formulario para el envío seguro de datos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <keygen> | | | | | | |
| Descripción | Genera un para de claves para el envío seguro de datos | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| **10** | **4.0** | **IExplorer** | **11.0** | **5.1** |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | autofocus, challenge, disabled, form, keytype, name | | | | | | |
| Diferencia entre html  4.01 y html5 | Elemento nuevo HTML5 | | | | | | |

#### Ejemplo:

<!DOCTYPE html>

<html lang="es-es">

<head>

<title>Keygen</title>

<meta charset="UTF-8">

<style type="text/css">

.fondoverde

{background-color:#ccffcc; }

</style>

</head>

<body class="fondoverde">

<h2> Keygen</h2>

<form action="validacion.php"> Nombre:

<input type="text" name="user">

<br><br>

Encriptación: <keygen name="seguro">

<br><br>

<input type="submit">

</form>

</body>

</html>



En este ejemplo se enviaría en la variable “user” del formulario el valor de “Sandra” con la clave de encriptación en el variable “seguro”. Como se ve la encriptación está en una lista desplegable con dos posibles valores: 2048 (Grado elevado) o 1024 (Grado medio). Estas variables serían recogidas por el programa “validacion.php”.

# EJEMPLO DE CONSTRUCCIÓN DE UN FORMULARIO CON ESTILOS

Los formularios no dejan de ser conjuntos de elementos HTML5 cuya información, introducida por el usuario, se envía a otros programas para ser procesada, por lo que la aplicación de estilos es prácticamente igual a que hemos visto en el tema de CSS, es decir, se crea la página html con el formulario (se deciden los tipos de elementos que forman parte del formulario y se les van aplicando estilos.

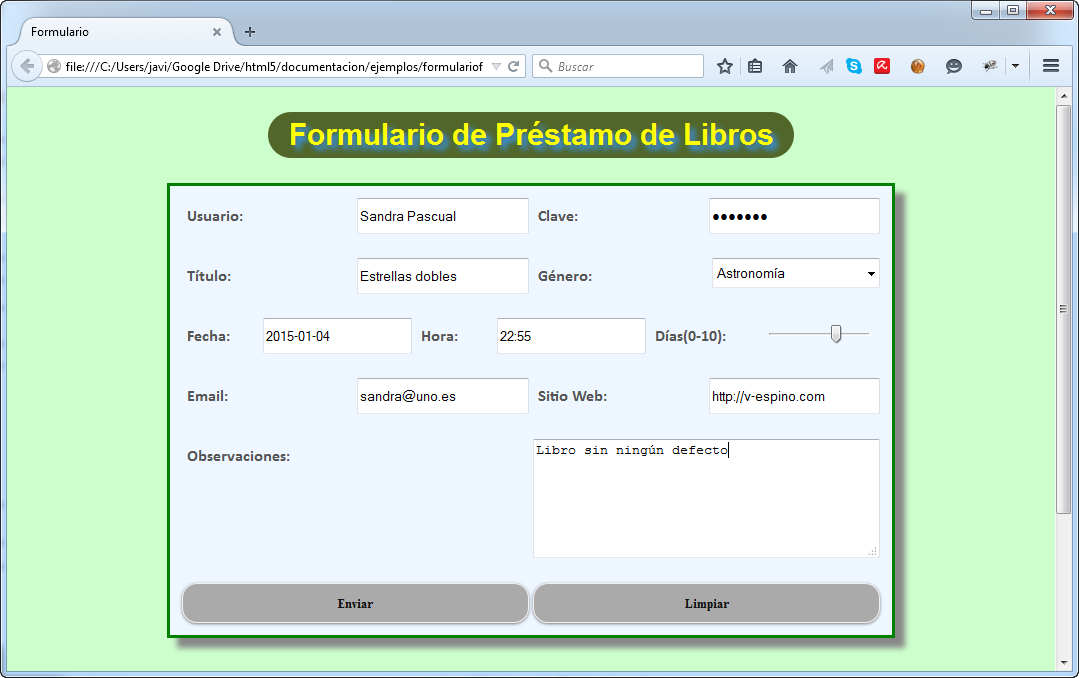
Un posible ejemplo de formulario podría ser el siguiente:

Queremos gestionar los préstamos de libros de una biblioteca pública a través de una página HTML5. Como necesitamos recolectar datos, aunque, en principio, no se envían a ningún sitio, vamos a incluir un formulario en nuestra página con los siguientes campos:

* *Usuario*: Nombre de la persona a la que se presta el libro.
* *Clave*: Clave de acceso que tiene la persona anterior.
* *Título:* Título del libro que se presta.
* *Género:* Temática del libro anterior que solo puede ser: astronomía, literatura, matemáticas, historia o física.
* *Fecha*: Fecha de préstamo.
* *Hora:* Hora de préstamo.
* *Días*: Días del préstamo que tiene como rango de 1 a 10 con el valor 7 por defecto.
* *Email:* Dirección de correo electrónico de la persona a la que se presta el libro.
* *Sitio web*: Página web, si la tuviera, de la persona a la que se presta el libro.
* *Observaciones:* Consideraciones a tener en cuenta sobre el préstamo que será un área de texto de 40 columnas y 6 filas.
* *Enviar:* Botón para enviar el formulario.
* *Limpiar*: Botón para limpiar los datos del formulario.

También le añadiremos un título a la página y haremos que el formulario quede centrado en la página sin ocupar todo el ancho.

Una posible distribución sería esta:

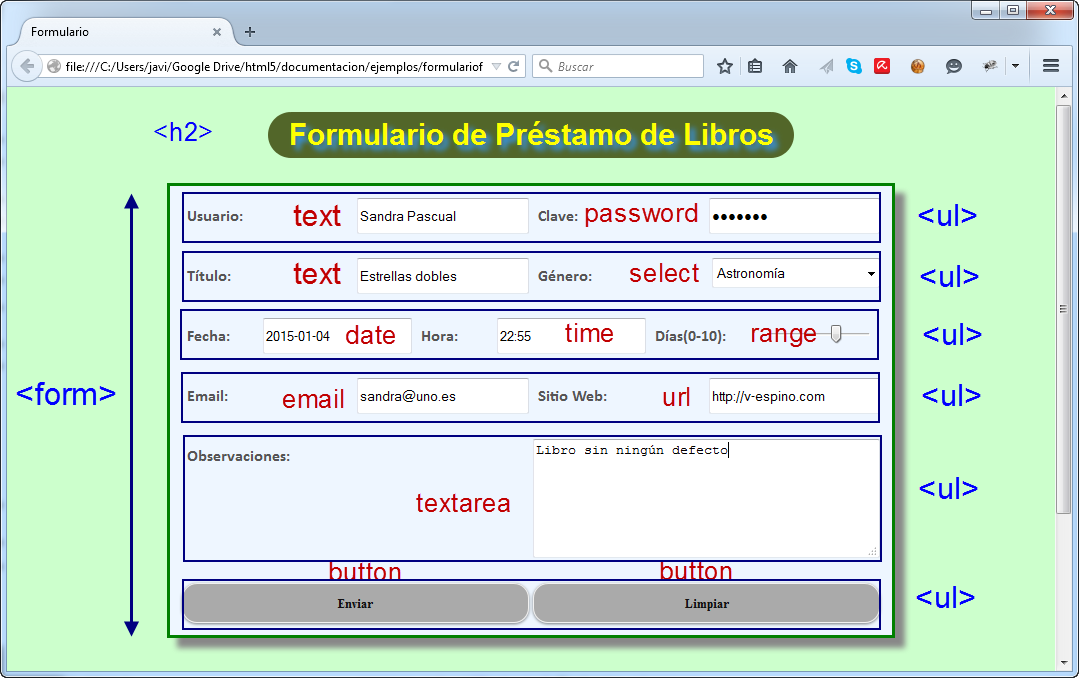


Ahora tenemos que decidir qué tipos de elementos usamos dentro del formulario y cuáles requeridos u obligatorios (atributo “required”). Podrían ser estos:

|  |  |
| --- | --- |
| **CAMPO** | **ELEMENTO Y TIPO** |
| Usuario | <input type=”text” required> |
| Clave | <input type=”password” required> |
| Título | <input type=”text” required>> |
| Género | <select> |
| Fecha | <input type=”date” required> |
| Hora | <input type=”time”> |
| Días | <input type=”range” required> |
| Email | <input type=”email” required> |
| Sitio Web | <input type=”url”> |
| Observaciones | <textarea> |
| Enviar | <button type=”submit”> |
| Limpiar | <button type=”reset”> |

También vamos a utilizar listas desordenadas para organizar los grupos de elementos. Cada fila del formulario será un elemento <ul> compuesto de elementos <li> formados por

un elemento <label> y el correspondiente elemento del formulario. Fíjate en la imagen para hacerte una idea.



Teniendo en cuenta lo anterior el código html quedaría así:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="utf-8">

<title>Formulario</title>

</head>

<body>

<h2>Formulario de Préstamo de Libros</h2>

<form>

<ul>

<li>

<label>Usuario:</label>

<input type="text" required>

</li>

<li>

<label>Clave:</label>

<input type="password" required>

</li>

</ul>

<ul>

<li>

<label>Título:</label>

<input type="text" required>

</li>

<li>

<label>Género:</label>

<select>

<option value="Astronomía" selected>Astronomía</option>

<option value="Literatura">Literatura</option>

<option value="Matem&acute;ticas">Matemáticas </option>

<option value="Historia">Historia </option>

<option value="Física">Física </option>

</select>

</li>

</ul>

<ul>

<li>

<label>Fecha:</label>

<input type="date" required>

</li>

<li>

<label>Hora:</label>

<input type="time">

</li>

<li>

<label>Días(0-10): </label>

<input type="range" min="0" max="10" value="7" required>

</li>

</ul>

<ul>

<li>

<label>Email:</label>

<input type="email" required>

</li>

<li>

<label>Sitio Web:</label>

<input type="url">

</li>

</ul>

<ul>

<li>

<label>Observaciones:</label>

<textarea cols="40" rows="6"></textarea>

</li>

</ul>

<ul>

<li>

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

</li>

<li>

<button type="reset">Limpiar</button>

</li>

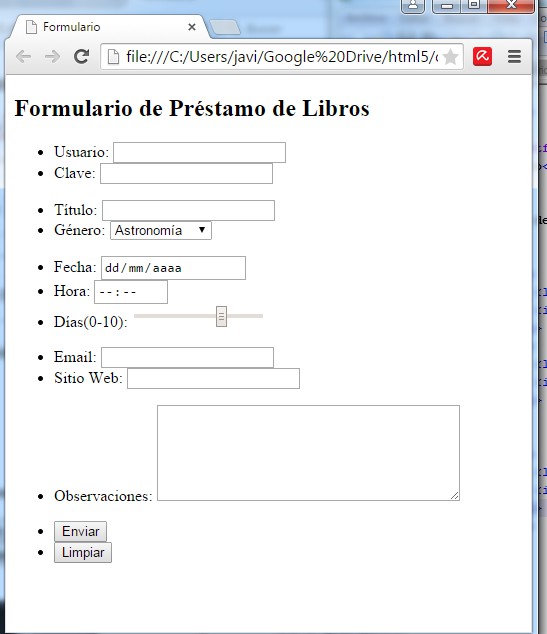
</ul>

</form>

</body>

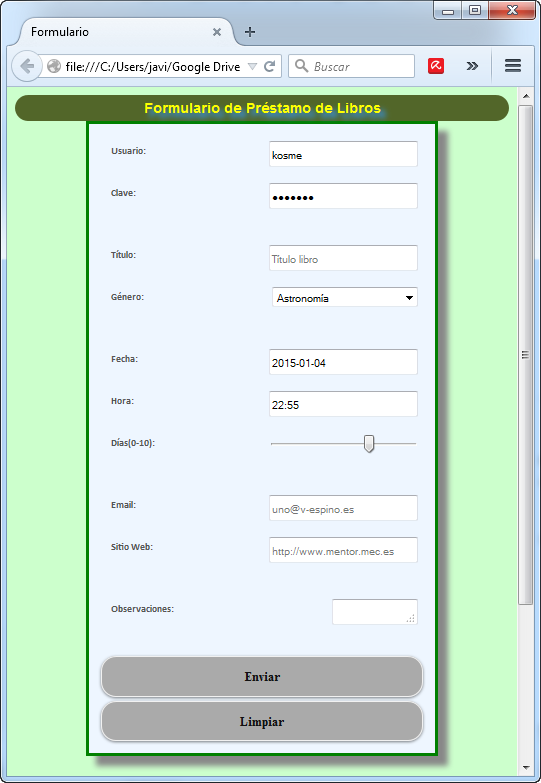
</html>

En el navegador se vería así:



A partir de aquí es cuando se empiezan a aplicar estilos

**Ejercicio 59:** Aplica estilos al ejemplo anterior utilizando contenedores y elementos flexibles para que, cuando cambies la resolución te quede de la siguiente manera:



Sugerencia: El elemento <form> y los elementos <ul> son contenedores flexibles y el tamaño de la letra es más pequeño al cambiar la resolución.

**Ejercicio 60:** Para que veas como se envía la información a través de un formulario, vamos a enviar la información a la página parámetros.html que, a través de script javascript, devuelve los valores del formulario y vamos a visualizar dicha página en un

marco flotante, <iframe>, que pondremos justo debajo del formulario. Así, cuando se pulse el botón de “Enviar”, la información se ve en el marco.

Usa la siguiente declaración del elemento <form>

<form id="formulario" class="centrar" action="parámetros.html" method="get" target="marco">

-------

</form>

Y esta para el elemento <iframe>

<iframe class="centrar" name="marco">

……

</iframe>