

## UT2. LENGUAJES DE MARCAS DE VISUALIZACIÓN.

### FORMULARIOS

1.Formularios.....	2
2.<INPUT> .....	5
3.Introducción por medio de texto (una línea).....	6
4.Introducción por medio de texto (múltiples líneas).....	8
5.Introducción por medio de menús.....	8
6.Formulario de confirmación (checkbox).....	10
7.Botones de radio.....	10
8.Botones de envío y de borrado.....	11
9. <fieldset> . . . . </fieldset> y <legend> . . . . </legend>.....	12
10. La etiqueta <label> del HTML:.....	13
11.Button.....	14
12.Otros Atributos de interés.....	14
13.INPUT TYPES.....	14

## 1. Formularios

Un formulario es un elemento que permite recoger datos introducidos por el usuario. Los formularios se utilizan para conocer las opiniones, dudas, y otra serie de datos sobre los usuarios, para introducir pedidos a través de la red, tienen multitud de aplicaciones.

Un formulario está formado, entre otras cosas, por etiquetas, campos de texto, menús desplegables, y botones.

Los formularios son un método en HTML para que el usuario de una página WEB pueda enviar información a un Sitio Web, mediante una serie de controles, cuadros de texto y dialogo.

La estructura de un formulario es de la siguiente forma:

```
<form ..... atributos.....>
.....
controles y textos del formulario
.....
</form>
```

Lo normal es que la etiqueta `<form>` lleve al menos un atributo que determine la acción a realizar por el formulario, pero si solo deseamos incluir botones no es necesario nada más.

Los principales atributos de la etiqueta `<form>` son 5:

Código	Descripción
<b>action</b>	Programa o URL que va a tratar los datos del formulario
<b>method</b>	Cómo se van a enviar los datos: GET   POST
<b>enctype</b>	Qué tipo de datos contiene (texto plano, ficheros, etc.). El valor por defecto es "application/x-www-form-urlencoded". Con el elemento INPUT, type="file" debe usarse el valor "multipart/form-data".
<b>target</b>	_blank _self _parent _top
<b>name</b>	nombre del formulario

**El atributo action:**

[https://www.w3schools.com/tags/att\\_form\\_action.asp](https://www.w3schools.com/tags/att_form_action.asp)

Es la URL del recurso que en el servidor se encargará, mediante un script CGI, PHP, etc. o mediante el protocolo de correo SMTP, de gestionar el contenido del formulario que hemos enviado; si se omite el prefijo de la URL, el servidor entenderá que el script está ubicado en la misma dirección que la página del formulario.

El programa del servidor será el encargado de tratar la información que le hemos enviado en el formulario (guardar en base de datos, agruparla en tablas, realizar estadísticas, enviar una página de confirmación, etc.) y devolver la respuesta al navegador en el formato que este entiende (HTML).

```
<form action="usuarios.php">  
. . . . .  
</form>
```

```
<form action="http://www.iesellago.net/tratamientos/usuarios.php">  
. . . . .  
</form>
```

Se puede usar un cliente de correo electrónico (con SMTP) a la que se enviarán los contenidos del formulario.

**Ejemplos:**

```
<form action="mailto:antoniomolina@retemail.es">  
. . . . .  
</form>
```

otra forma ampliada de la URL del correo es enviando el “asunto” del correo:

```
<form action="mailto:josele@retemail.es?subject=formulario">  
. . . . .  
</form>
```

**El atributo method:**

[https://www.w3schools.com/tags/att\\_form\\_method.asp](https://www.w3schools.com/tags/att_form_method.asp)

Es el método usado para enviar los contenidos del formulario al servidor que gestionará la información. Existen dos métodos: “get” y “post”, dependiendo de la funcionalidad de nuestro servidor. En la actualidad se usa casi exclusivamente el método “post” (mayor seguridad).

- **get:** con el método HTTP "get", el conjunto de datos del formulario se agrega al URI especificada por el atributo action (con un signo de interrogación ("?") como separador) y este nuevo URI se envía al agente procesador.
- **post:** con el método HTTP "post", el conjunto de datos del formulario se incluye en el cuerpo del formulario y se envía al agente procesador de forma no visible al usuario.

El método por defecto para enviar datos es GET.

Cuando el método usado es GET, los datos del formulario que se envían están visibles en el campo de dirección de la página. Suele utilizarse en buscadores.

#### Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

Se debe usar siempre POST si los datos que contiene el formulario incluyen información personal. El método POST no muestra los datos enviados en el campo de dirección del formulario

```
http://www.arrakis.es/datos?miformu="Datos"&Edad=25&Casado="si"& . . .
```

#### Notes on POST:

- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

#### **El atributo enctype:**

[http://www.w3schools.com/tags/att\\_form\\_enctype.asp](http://www.w3schools.com/tags/att_form_enctype.asp)

Especifica la codificación para los contenidos del formulario. Este atributo sólo se aplica cuando se utiliza el método "post".

Los posibles valores son:

- Por defecto: "application/x-www-form-urlencoded". La información es codificada de tal forma que los espacios se convierten a + y los caracteres especiales a valores ASCII en hexadecimal.
- Para información textual: text/plain". Los espacios se convierten a símbolos +, pero los caracteres especiales no se codifican.
- Para información binaria: "multipart/form-data" . Obligatorio cuando tenemos un control de tipo <input type=file...

Ejemplo:

```
<form action="datos.php" method="post" enctype="multipart/form-data">  
. . . . .  
</form>
```

### **El atributo name**

Es el nombre que identifica el formulario. Es la manera de hacer y tener referencia de un elemento dentro de la web.

### **El atributo target**

[http://www.w3schools.com/tags/att\\_form\\_target.asp](http://www.w3schools.com/tags/att_form_target.asp)

Indica dónde se va a visualizar la información devuelta por el servidor.

El valor por defecto es "\_self" que indica que la información se mostrará en la ventana actual.

Para hacer que la información aparezca en una pestaña nueva el valor es "\_blank".

Otros valores son "\_parent" y "\_top" que indican el nombre de un iframe.

## **2. <input>**

Los elementos de formulario como botones y cuadros de texto también se denominan "campos de formulario" y "controles de formulario". La mayoría de controles se crean con la etiqueta <input>,

por lo que su definición formal y su lista de atributos es muy extensa. La introducción de los datos se consigue por medio de la etiqueta:

```
<INPUT TYPE="xxx" NAME="yyy" VALUE="zzz" >
```

El atributo **type** indica el tipo de control que se incluye en el formulario **type="text | password | checkbox | radio | submit | reset | file | hidden | image | button"**

El atributo **name** identifica el campo del formulario. De hecho, si un campo no incluye el atributo **name**, sus datos no se envían al servidor. El valor que se indica en el atributo **name** es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.

El atributo **value** se emplea para establecer el valor inicial del cuadro de texto. Si se crea un formulario para insertar datos, los cuadros de texto deberían estar vacíos. Por lo tanto, o no se añade el atributo **value** o se incluye con un valor vacío **value=""**. Si por el contrario se crea un formulario para modificar datos, lo lógico es que se muestren inicialmente los datos guardados en el sistema. En este caso, el atributo **value** incluirá el valor que se desea mostrar:

```
<input type="text" name="nombre" value="Juan Pérez" >
```

### 3. Introducción por medio de texto (una línea)

En este caso es **xxx=text**, es decir, **input type="text"**. El atributo **VALUE** no procede en este caso.

Vamos a poner un ejemplo:

```
<form action="/action_page.php" method="get">
    First      name: <input type="text" name="fname"><br>
    Last      name: <input type="text" name="lname"><br>
    <input type="submit" value="Submit">
</form>
```

La longitud de este campo de formulario es por defecto de 20 caracteres. Se puede variar incluyendo en la etiqueta el atributo **size="número"**.

Su uso es imprescindible en muchos formularios, en los que algunos campos como la dirección deben mostrar más caracteres de lo normal (`<input size="100" ...`) y otros campos como el código postal deben mostrar menos caracteres de lo normal (`<input size="5" ...`).

Por otra parte, sea cual sea la longitud del formulario, si no se indica nada, el usuario puede introducir el número de caracteres que quiera. Se puede limitar esto, incluyendo en la etiqueta el atributo `MAXLENGTH="número"`.

El uso de `MAXLENGTH` es imprescindible para campos como el código postal, el número de la Seguridad Social y cualquier otro dato con formato predefinido y limitado.

El atributo `READONLY` permite que el usuario pueda ver los contenidos del cuadro de texto pero no pueda modificarlos:

```
<form action="/action_page.php">
  Email: <input type="text" name="email"><br>
  Country: <input type="text" name="country" value="Norway" readonly><br>
  <input type="submit" value="Submit">
</form>
```

También se puede hacer que el texto introducido no sea reconocible, es decir que todos los caracteres se representen por asteriscos. Basta con cambiar en la etiqueta `INPUT TYPE="text"` por `INPUT TYPE="password"`.

Para el último ejemplo:

```
<INPUT TYPE="text" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

ó

```
<INPUT TYPE="password" NAME="Apellido" SIZE="10" MAXLENGTH="12" >
```

Los diseñadores de aplicaciones deberían tener en cuenta que este mecanismo ofrece en realidad un nivel muy bajo de seguridad. Aunque la contraseña se oculte a las personas que puedan estar mirando, se transmite al servidor como texto sin enmascarar, y puede ser leído por cualquiera que tenga acceso de bajo nivel a la red.

## 4. Introducción por medio de texto (múltiples líneas)

Cuando el texto a introducir puede alcanzar una gran longitud, por ejemplo un comentario, es conveniente utilizar un formulario de texto de múltiples líneas.

Esto se consigue con la etiqueta de inicio:

```
<TEXTAREA NAME="yyy" ROWS="número" COLS="número">
```

Como puedes ver:

- no se utiliza INPUT TYPE
- **ROWS** representa el número de filas, y **COLS** el de columnas
- Se cierra **</TEXTAREA>**

Ejemplo:

```
<textarea rows="4" cols="50">
At w3schools.com you will learn how to make a website. We offer
free tutorials in all web development technologies.
</textarea>
```

## 5. Introducción por medio de menús

Si queremos que el usuario, en vez de introducir un texto, como hemos visto en los casos anteriores, escoja entre varias opciones que le presentamos nosotros, haremos uso de un formulario en forma de menú.

Se consigue con la etiqueta de inicio **<SELECT NAME="yyy">** y la de cierre **</SELECT>**.

Las distintas opciones a escoger se consiguen con la etiqueta **<OPTION>**.

Ejemplo: Pedimos al usuario que elija su color preferido:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
¿Cuál es tu Animal preferido?
<SELECT NAME="AnimalPreferido">
  <OPTION value="Gato">Gato</option>
  <OPTION value=" Perro" selected> Perro </option>
  <OPTION value=" León"> León </option>
  <OPTION value="Aguila"> Aguila </option>
</SELECT >
```



```
</FORM>
```

El atributo **value** de cada opción es obligatorio, ya que es el dato que se envía al servidor cuando el usuario envía el formulario. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo **selected** a la opción deseada.

En el ejemplo anterior, sólo es visible en el formulario una opción. Si queremos que sean visibles múltiples opciones a la vez, añadimos en la etiqueta el atributo **SIZE="número"**, donde especificamos el número de opciones visibles.

```
<select size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

El atributo **multiple** permite la selección de varias opciones.

```
<SELECT NAME="AnimalPreferido" MULTIPLE SIZE="2">
<SELECT NAME="AnimalPreferido" MULTIPLE >
```

**Disabled** deshabilita el menú sin permitir su selección y su valor no se envía al servidor junto con el resto de datos.

```
<SELECT NAME="AnimalPreferido" disabled>
```

Por otra parte, las listas desplegables permiten agrupar sus opciones de forma que el usuario pueda encontrar fácilmente las opciones cuando la lista es muy larga:

```
<select>
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

La etiqueta **<optgroup>** permite agrupar opciones relacionadas dentro de una lista desplegable.

El único atributo que suele utilizarse con la etiqueta **<optgroup>** es **label**, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación, de forma que el usuario pueda localizar más fácilmente la opción deseada.

## 6. Formulario de confirmación (checkbox)

Si queremos que el usuario confirme una opción determinada, podemos hacer uso de un formulario de confirmación, o checkbox, que se consigue con la etiqueta:

```
<INPUT TYPE="checkbox" NAME="yyy" >
```

Ejemplo:

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a
    bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

Si no se añade un texto al lado de la etiqueta **<input >** del checkbox, el usuario sólo ve un pequeño cuadrado sin ninguna información relativa a la finalidad de ese checkbox.

Si queremos que el formulario aparezca inicialmente como marcado (el usuario no necesitará hacerlo), basta con añadir el atributo **CHECKED** dentro de la etiqueta. En el ejemplo anterior si sustituimos la etiqueta equivalente por:

```
<INPUT TYPE="checkbox" NAME="Lista" checked>
```

## 7. Botones de radio

Cuando queremos que el usuario elija una única opción entre varias, podemos hacer uso de los botones de radio, que se consiguen con la etiqueta:

```
<INPUT TYPE="radio" NAME="yyy" VALUE="zzz" >
```

Donde **yyy** es el nombre que le ponemos a la variable que se trata de elegir, y **zzz** es el nombre de cada una de las opciones en concreto.

Ejemplo:

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Obsérvese el atributo opcional **CHECKED** que se ha añadido en la primera etiqueta. Esa será la opción que aparece marcada por defecto.

El atributo **name** se emplea para indicar los **radiobutton** que están relacionados. Por lo tanto, cuando varios **radiobutton** tienen el mismo valor en su atributo **name**, el navegador sabe que están relacionados y puede deseleccionar una opción del grupo de **radiobutton** cuando se seleccione otra opción

## 8. Botones de envío y de borrado

Hasta ahora, en todos los ejemplos que hemos visto, faltaba un elemento esencial en cualquier formulario, y es el botón de envío de los datos, que se consigue con la etiqueta:

```
<INPUT TYPE="submit" VALUE="zzz" >
```

En donde **zzz** es el texto que queremos que aparezca en el botón. Si no se establece el atributo **value**, el navegador muestra el texto predefinido **Enviar consulta**.

Otro botón interesante es el de borrado de los datos introducidos, muy conveniente en un formulario con muchos elementos. Es muy similar al de envío, pues se consigue con la etiqueta:

```
<INPUT TYPE="reset" VALUE="zzz" >
```

En donde **zzz** es el texto que queremos que aparezca en el botón.

Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original. Si el formulario no contenía originalmente ningún valor, el botón de

**reset** lo vuelve a mostrar vacío. si el formulario contenía información, el botón **reset** vuelve a mostrar la misma información original.

Como es habitual en los botones de formulario, el atributo **value** permite establecer el texto que muestra el botón. Si no se utiliza este atributo, el navegador muestra el texto predefinido del botón, que en este caso es Restablecer.

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

El atributo **VALUE** sólo puede contener texto, no imágenes.

Hasta ahora hemos visto uno a uno los diferentes elementos que se pueden utilizar. Pero no hay ningún inconveniente en usar, dentro del mismo formulario, **distintos tipos** de introducción de datos.

Otros tipos de botones: el atributo **button** define un botón cualquiera sin comportamiento predefinido.

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

## 9. **<fieldset> . . . . </fieldset>**      **y**      **<legend> . . . . </legend>**

Se pueden agrupar elementos de formularios en “cajas” que pueden llevar un título.

La etiqueta **<fieldset>** se utiliza para agrupar elementos y dibuja una línea alrededor de los elementos agrupados.

La etiqueta **<legend>** es el título del formulario

Con un ejemplo se comprende su funcionamiento:

```
<form>
  <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text"><br>
    Email: <input type="text"><br>
    Date of birth: <input type="text">
```

```
</fieldset>
</form>
```

## 10. La etiqueta <label> del HTML:

Todos los controles de los formularios, salvo los botones de radio, tienen una carencia muy importante: no disponen de la opción de establecer el título o texto que se muestra junto al control, de manera el nombre de cada campo se incluye en forma de texto normal, sin ninguna relación con el campo al que hace referencia.

En html existe la etiqueta <label> que se puede asociar con un campo de formulario para establecer su título. Su formato es: <label for="ID\_del\_Elemento"> . . . </label>

Un código como el que sigue:

```
Nombre <br>
<input type="text" name="nomb" value="">
Apellidos <br>
<input type="text" name="apel" value="">
DNI <br>
<input type="text" name="dni" value="" size="10" maxlength="9">
```

Estaría mucho mejor estructurado de esta forma:

```
<label for="nomb"> Nombre </label> <br>
<input type="text" id="nomb" name="nomb" value="">
<label for="apel"> Apellidos </label> <br>
<input type="text" id="apel" name="apel" value="">
<label for="dni"> NIF </labe> <br>
<input type="text" id="dni" name="dni" value="" size="10"
maxlength="9">
```

Evidentemente el uso de <label> no nos evita el tener que escribir sus datos en el sitio correcto de HTML, pero además de quedar reflejado en el código a quien pertenece ese texto, nos sirve para que al pulsar con el puntero del ratón sobre su texto, se nos posicione el mismo sobre el campo asociado correspondiente, esto es de gran utilidad en los controles radiobutton y checkbox.

## 11. Button

Button define un botón sobre el que se puede pulsar, dentro del botón se puede introducir contenido como texto e imágenes.

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
  <button name="enviar" type="submit" value="mandar">
    </button>
  <button name="resetear" type="reset" value="borrar">
    Resetear</button><br/>
  <button name="boton" type="button" value="boton">
    Simplemente un botón</button><br/>
  <button name="boton" type="button" value="deshabilitado" disabled>
    Botón deshabilitado</button><br/>
</FORM>
```

**src = "url"** - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario

## 12. Otros Atributos de interés

[http://www.w3schools.com/html/html\\_form\\_attributes.asp](http://www.w3schools.com/html/html_form_attributes.asp)

## 13. Input Types HTML5

HTML5 incorpora input types que no estaban en otras versiones y que son de interés.

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Puedes encontrarlos en la siguiente dirección, junto con los controles ya vistos:

[http://www.w3schools.com/html/html\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html_form_input_types.asp)