

Desarrollo Web con HTML

Módulo 3

Formularios en HTML

Introducción a formularios

Los formularios más sencillos se pueden crear utilizando solamente dos etiquetas: `<form>` e `<input>`.

Un ejemplo claro, es el formulario a continuación:

Formulario muy sencillo

Nombre usuario

Enviar

Formularios

La mayoría de formularios utilizan solo los atributos **action** y **method**.

Action

El atributo **action** indica la **URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios**. Esta aplicación también se encarga de **generar la respuesta que muestra el navegador**.

Para lograr un formulario simple, debemos hacer lo siguiente en nuestro HTML:

```
<form action="codigo.php">
<input type="text" name="nombre" value=""><br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

Method

El atributo **method** establece **la forma en la que se envían los datos del formulario al servidor**. Hace referencia al **método HTTP**, por lo que no es algo propio de HTML.

Valores de method: get y post

Los dos valores que se utilizan en los formularios son **get** y **post**. Casi todos los formularios incluyen el atributo **method="get"** o el atributo **method="post"**.



¿Qué método elegir?

Si no sabes qué método elegir para un formulario, existe una **regla general**, que dice:

- El método **get** se debe utilizar en los formularios que **no modifican la información**. (por ejemplo, en un formulario de búsqueda).

El ejemplo más común de formulario con método **get** es el de los buscadores. Si realizas una búsqueda con tu buscador favorito, verás que las palabras que introduces en tu búsqueda aparecen como parte de la URL de la página de resultados.

- Por su parte, el método **post** se debería utilizar **cuando el formulario modifica la información original** (insertar, modificar o borrar alguna información).

Nombre:

Edad:

Enviar datos

Uso del método *post*

Atributo *Enctype*

Cuando elegimos **post** para trabajar y por caso queremos que nuestro formulario nos posibilite adjuntar archivos, debemos especificar este atributo llamado **enctype**.

Los **valores posibles** son:

- Valor por defecto:

```
application/x-www-form-urlencoded
```

- Valor a utilizar si vamos a adjuntar un archivo:

```
multipart/form-data
```

- Otro valor posible es el siguiente:

```
text/plain
```

No se recomienda trabajar con `text/plain` dado que **envía data sin ser codificada**.

Campo de archivo

Utilización de *File*

Los formularios también permiten **adjuntar archivos** para subirlos al servidor.

Para adjuntar un archivo, además de utilizar el valor **multipart/form-data**, debemos trabajar con **file** como vemos en el ejemplo:

```
<form action="codigo.php" method="post" enctype="multipart/  
form-data">  
  <label>Adjuntar archivo <input type="file"></label>  
</form>
```



Adjuntar archivo Elegir archivo No se ha seleccionado ningún archivo

Aunque desde el punto de vista de HTML y del navegador no existe ninguna limitación sobre el número, tipo o tamaño total de los archivos que se pueden adjuntar, **todos los servidores añaden restricciones por motivos de seguridad.**

Atributo `multiple`

Si a su vez a este campo le agregamos el atributo **`multiple`**, como vemos en la imagen de la derecha, nos permitirá adjuntar varios archivos al mismo tiempo. **El atributo `multiple` no necesita un valor porque es un atributo booleano.**

```
<label>Adjuntar archivo <input multiple type="file"></label>
```

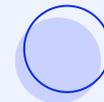
Archivo adjunto

Examinar

¿Qué es un atributo booleano?

En el trabajo con archivos, muchas veces los atributos no poseen valores, por ejemplo, en el caso de **multiple**.

Esto es así porque los atributos se los considera *booleanos* es decir, **el valor es igual a nombre del atributo**. Por ello, para evitar escribir dos veces la misma palabra, podemos omitir entonces el valor. Veamos un ejemplo:



```
<form action="codigo.php" method="post" enctype="multipart/form-data">  
  <label>Adjuntar archivo <input multiple="multiple" type="file"></label>  
</form>
```

Campo de texto

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un **cuadro de texto vacío en el que el usuario puede escribir cualquier texto.**

Para realizarlo debemos trabajar en nuestro código, como se muestra a la derecha:

```
Nombre <br>  
<input type="text" value="nombre">
```

Nombre:

Atributos y campos

- El elemento **label** sirve para ayudar al usuario a acceder a nuestro formulario.
- El atributo **placeholder** nos permite **guiar al usuario** con referencia lo que tiene que colocar dentro de nuestro campo.
- Los atributos **minlength** y **maxlength** permiten **indicar las cantidades mínima y máxima de caracteres** que admite el campo, respectivamente.

```
<label>Nombre  
  <input type="text" minlength="5" maxlength="15">  
</label>
```

```
<label> Nombre <br> <input type="text" placeholder="Nombre Completo" maxlength="10"  
placeholder="Nombre Completo"> </label>
```

Campo de contraseña

La única diferencia entre este control y el cuadro de texto normal, es que **el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla.**

Los navegadores ocultan el texto, utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.

Cambiando el valor del atributo **type** por **password**, se transforma el cuadro de texto normal en un cuadro de contraseña. **Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.**

```
<input type="password" name="contraseña">
```

Contraseña:

Campos de *email* y de número

Campo de *email*

El campo de **email** me permite **validar un formato de correo electrónico correcto**. Es decir, empieza a aparecer la validación con **HTML5**.

```
<label>Email <input type="email" required  
placeholder="info@dominio.com" > </label>
```

Si a este campo le agregamos el atributo **multiple**, permitirá colocar varios correos separados por **coma**, por ejemplo.



```
<label>Email personal, Email corporativo  
<input type="email" multiple>  
</label>
```

Campo de número

El campo de **number** me permite **fixar el valor de número como único valor posible**. Podemos fijar un **rango** con el atributo **min** y **max**.

A su vez es importante entender que el **campo de número** debe usarse sólo para números cuya finalidad no sea por ejemplo números que forman parte de una tarjeta de crédito o un teléfono

Para esa finalidad deberían utilizarse **campos de texto o de teléfono** en su defecto.

Nota: en la actualidad, la mayoría de los navegadores no dejan siquiera escribir letras en los campos number.

```
<label>Edad <input type="number" required  
min="10" max="80" > </label>
```



Tu edad:

Enviar ⚠ Debes introducir un número.

Detailed description: The image shows a user interface for an age field. It consists of a label 'Tu edad:', a text input field containing the word 'veinte', and a light blue 'Enviar' button. A red error message box with a warning icon and the text 'Debes introducir un número.' is positioned below the input field, pointing to it. The entire form is enclosed in a rounded rectangle with a red border.

Menú desplegable

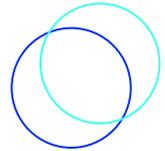
Para trabajar con un menú desplegable, se debe utilizar el elemento **select**, como se muestra en el ejemplo.

Si además queremos que una **opción aparezca seleccionada de manera predeterminada**, el atributo a utilizar es **selected**. El atributo `selected` es booleano.

```
<label>Países
  <select name="países">
    <option value="argentina">Argentina</option>
    <option value="brasil" selected>Brasil</option>
  </select>
</label>
```

Atributo *multiple*

El menú desplegable generalmente se utiliza para seleccionar una opción entre varias. Sin embargo, podemos transformar este menú en uno **múltiple**, que permita **seleccionar varias opciones**. Lo haremos de la siguiente manera:



```
<label>Deportes
  <select name="deportes" multiple>
    <option value="tenis">Tenis</option>
    <option value="futbol">Fútbol</option>
    <option value="natacion">Natación</option>
  </select>
</label>
```



Caja de comentarios

Elemento *textarea*

La **caja de comentarios** es una caja de texto multilínea que permitirá **hacer comentarios o sentencias de más de una línea**.

```
<textarea> </textarea>
```



Etiquetas *Fieldset* y *Legend*

La etiqueta `<fieldset>` agrupa campos del formulario y la etiqueta `<legend>` asigna un nombre a cada grupo.

En el ejemplo, se muestra el código HTML del formulario que hace uso de `<fieldset>` y `<legend>` para agrupar los campos del formulario.

En el siguiente slide, encontrarás el resultado de este bloque de código.

```
<form>
  <fieldset>
    <legend>Datos personales</legend>
    Nombre: <input type="text"><br>
    E-mail: <input type="text"><br>
    Edad: <input type="number">
  </fieldset>
</form>
```

En la pantalla del navegador, el código del ejemplo anterior se verá así:

Datos Personales

Nombre:

Email:

Edad:



Revisión

- Repasar los **elementos de formulario** vistos.
- Incorporar **variantes y diferentes atributos**.
- Visitar el siguiente [link](#) para obtener más información sobre formularios.



**¡Sigamos
trabajando!**