

Formularios

- métodos para recolectar información que debe ser rellena por el usuario.
- Pueden servir para permitir la autenticación de un usuario en una página Web, recoger información que debemos insertar. Se actualiza en una base de datos, realizar búsquedas de información en el que debemos especificar un criterio de búsqueda, etc.

Formularios: GET y POST

- Son métodos del protocolo HTTP para el intercambio de información entre el cliente y el servidor.
 - El método *GET* envía las variables dentro de la URL de la página.
 - En el método *POST* la información va codificada en el cuerpo de la petición HTTP y por lo tanto viaja oculta.
 - Mientras el método *GET* lo utilizamos para recuperar información, el método *POST* se usa para enviar información a un servidor Web.

Formularios: GET

```
<html>

  <head>

    <title>Ejemplo Formularios</title>

  </head>

  <body>

    <h1>Ejemplo de procesamiento de formularios</h1>

    <FORM ACTION="ejemplo1.php" METHOD="GET">

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

      <BR>
      Introduzca sus apellidos:<INPUT TYPE="text"

        NAME="apellidos"><BR>
      <INPUT TYPE="submit" VALUE="Enviar">

    </FORM>

  </body>

</html>
```

Formularios: POST

```
<html>

  <head>

    <title>Ejemplo Formularios</title>

  </head>

  <body>

    <h1>Ejemplo de procesamiento de formularios</h1>

    <FORM ACTION="ejemplo2.php" METHOD="POST">

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

      <BR>
      Introduzca sus apellidos:<INPUT TYPE="text"

        NAME="apellidos"><BR>
      <INPUT TYPE="submit" VALUE="Enviar">

    </FORM>

  </body>

</html>
```

Formularios: acceso con GET

- **GET:**

- **PHP:**

- `echo $_GET['nombre'];`
 - `echo $_GET['apellidos'];`
 - `print_r($_GET);`

- **ASP:**

- `response.write(request.QueryString("nombre"))`
 - `response.write(request.QueryString("apellidos"))`

- **JSP (válido para GET y POST):**

- `out.print(request.getParameter("nombre"));`
 - `out.print(request.getParameter("apellidos"));`

Formularios: acceso con POST

- **POST:**

- **PHP:**

- `echo $_POST['nombre'];`
 - `echo $_POST['apellidos'];`
 - `print_r($_POST);`

- **ASP:**

- `response.write(request.Form("nombre"))`
 - `response.write(request.Form("apellidos"))`

- **JSP (válido para GET y POST):**

- `out.print(request.getParameter("nombre"));`
 - `out.print(request.getParameter("apellidos"));`

Formularios: campos vacíos

¿Qué sucede cuando el usuario no rellena algún **campo** de un formulario?

- Algunas veces se envía el campo como una cadena vacía.
- Algunas veces no se envía ni el nombre de campo.

La tabla siguiente ilustra el comportamiento de varios controles de formulario cuando no están rellenos o seleccionados:

Formularios: campos vacíos

Tipo control	¿Qué sucede cuando no está relleno o seleccionado?
text	Se envía el nombre de campo, junto con un valor vacío
password	Igual que text.
checkbox	No se envía nada.
radio	Igual que checkbox.
submit	No se envía nada si no se hace clic en el botón. Esto puede suceder si el usuario pulsa Intro para enviar un formulario. Sin embargo, si solamente hay un botón de envío en el formulario, la mayoría de los navegadores enviarán el nombre y valor del campo del botón.
reset	Igual que checkbox.
file	Igual que text.
hidden	Igual que text.
image	Igual que submit.
button	Igual que checkbox.
select	Imposible no seleccionar una opción, por eso siempre se envía un valor.
Cuadro de lista	Igual que checkbox.
multiselector	Igual que checkbox.
textarea	Igual que text.

Formularios: subir archivos

Usaremos un control de selector de archivo dentro del formulario:

```
<label for="archivoSeleccionado">Seleccione archivo: </label>
```

```
<input type="file" name="archivoSeleccinado" id="archivoSeleccionado" value="" />
```

Además,

- 1.un formulario que contiene un selector de archivo debe utilizar el ***method="post"***.
- 2.debe tener un atributo ***enctype="multipart/form-data"*** en su etiqueta `<form>`. El atributo asegura que los datos se codifiquen como datos MIME multiparte.

Nota: Un formulario puede tener tantos selectores de archivo como se desee.

Formularios: subir archivos

La tabla superglobal `$_FILES` contiene la siguiente información sobre cada archivo subido:

Elemento	Descripción
name	Nombre del archivo subido.
type	Tipo MIME del archivo subido. Por ejemplo, una imagen JPEG probablemente tendría un tipo MIME de image/jpeg.
size	Tamaño del archivo subido, en bytes.
tmp_name	Ruta completa del archivo temporal en el servidor que contiene el archivo subido. (Todos los archivos se almacena inicialmente como archivos temporales, hasta que los cambiemos a otra ubicación)
error	Error o código de estado asociado al archivo subido.

Formularios: subir archivos

El elemento ***error*** contiene un valor entero que corresponde con una constante incorporada (built-in) que explica el estado del archivo subido:

Constante	Valor	Significado
UPLOAD_ERR_OK	0	El archivo se ha subido con éxito.
UPLOAD_ERR_INI_SIZE	1	El tamaño del archivo supera el especificado en la directiva <i>upload_max_filesize</i> en php.ini
UPLOAD_ERR_FORM_SIZE	2	El tamaño del archivo supera el especificado en la directiva MAX_FILE_SIZE en el formulario.
UPLOAD_ERR_NO_FILE	4	No se ha subido ningún archivo.
UPLOAD_ERR_NO_TMP_DIR	6	PHP no tiene acceso a una carpeta temporal en el servidor para almacenar el archivo.
UPLOAD_ERR_CANT_WRITE	7	El archivo no se ha podido escribir en el disco del servidor por alguna razón.
UPLOAD_ERR_EXTENSION	8	El archivo subido se ha detenido por una de las extensiones PHP actualmente cargadas.