

MANEJO DE FORMULARIOS

Programando con PHP

Manejo de formularios

Introducción

Introducción

- Un formulario es un conjunto de elementos que permiten obtener información del usuario.
- Tipos de elementos de un formulario:
 - Campos de texto: text
 - Conjunto selección circular: radio
 - Casillas de verificación: checkbox
 - Botones: submit – reset
 - Campos de contraseña: password

Introducción

- Es muy importante recordar en este punto que cada elemento de un formulario tenía como mínimo un atributo:
 - **NAME**: indicaba el nombre del elemento.
- Además, podía tener más atributos:
 - **VALUE**: valor que contiene el elemento

Introducción

- En la propia etiqueta de `<form>` era necesario indicar:
 - ▣ **Action**: programa que manejará el formulario.
 - ▣ **Method**: forma en que se enviarán los datos indicados por el usuario al servidor.
 - ▣ **Enctype**: método de encriptación utilizado.
 - ▣ **Accept-charset**: juego de caracteres que acepta el servidor.

Elementos de un formulario

Elementos

- Entre las etiquetas `<form>` y `</form>`...
- Podemos introducir un área de texto con las etiquetas `<textarea>` y `</textarea>`.
- Textarea acepta los siguientes atributos:
 - Name: nombre del área de texto
 - Rows: número de líneas que se podrán visualizar
 - Cols: número de caracteres por línea.

Elementos

- El elemento más utilizado es `<input> </input>` que engloba a la mayoría de los elementos de entrada de datos.
- Atributos de `<input>`
 - Name: nombre del elemento dentro del formulario
 - Value: valor inicial del elemento
 - Checked: opción preseleccionada
 - Type: tipo de elemento:

Text	Password	Checkbox	Radio	Submit
Reset	File	Hidden	Image	Button

Posibles tipos para INPUT

- Si elegimos poner en un formulario un elemento input, tenemos varios tipos para elegir:
- Tipos básicos:

Text	Password	Checkbox	Radio	Submit
Reset	File	Hidden	Image	Button

- Tipos extra

Number	Range	Date	Time	Search
Datalist	Tel	Email	url	Color

Elementos

- El elemento **text** permite introducir en el formulario un campo de texto.
- Tiene los siguiente parámetros:
 - **name='nombre'** Asigna un nombre único del elemento
 - **size='n'** Tamaño de la caja de texto que aparece en pantalla.
 - **maxlength='n'** Número máximo de caracteres.
 - **value=' texto '** Texto que aparecerá inicialmente.
 - **disabled** Desactiva la caja de texto, por tanto el usuario no podrá insertar ningún texto en dicho campo.
 - **readonly** El texto no puede ser modificado por el usuario.
 - **tabindex='n'** Asigna el orden de tabulador que tiene el campo respecto a los demás elementos que componen el formulario..

Elementos

- El elemento **password** permite introducir en el formulario un campo de texto pero el texto será sustituido por caracteres especiales.
- Tiene los mismos parámetros que text.
- El elemento **hidden** inserta en el formulario un campo escondido, por lo que no se ve en pantalla.
- Sus atributos son:
 - **name= 'nombre'** Asigna un nombre al campo oculto.
 - **value= 'valor'**. Asigna un valor campo.

Elementos

- El elemento **Radio** es un campo multivalor excluyente, permite escoger una y sólo una opción de un conjunto de valores.
- Posee lo siguientes parámetros:
 - **name='nombre'**. Asigna un nombre al conjunto. Este identificador debe ser el mismo para todos los elementos radio de un grupo.
 - **value='valor'**. Asigna un valor a la variable de cada casilla que componen el elemento radio.
 - **checked**. Selecciona por defecto uno de las opciones del grupo
 - **disabled**. Desactiva el radio botón, por lo que el usuario no podrá marcarlo.
 - **tabindex='n'**. Especifica el orden de tabulador que tendrá el campo respecto todos los elementos que componen el formulario.

Elementos

- El elemento **Checkbox** define las casillas de verificación que tendrá nuestro formulario.
- Este tipo de elementos pueden ser activados o desactivados por el usuario al pulsar sobre la caja.
- Este elemento no es auto excluyente.
- Tiene como parámetros:
 - **name='nombre'**. Asigna un nombre a cada elemento del grupo.
 - **value='valor'**. Asigna un valor a cada casilla de verificación que componen el checkbox.
 - **checked**. Selecciona por defecto una o más de las casillas del grupo.
 - **disabled**. Desactiva la casilla de verificación, por lo que el usuario no podrá seleccionarla.
 - **tabindex='n'**. Especifica el orden de tabulador que tendrá el campo respecto todos los elementos que componen el formulario.

Elementos

- El elemento **file** permite enviar un archivo adjunto al formulario.
- Añade una caja de texto y un botón con el que se abre un explorador para adjuntar el archivo
- Algunos de sus atributos son:
 - **Name='nombre'**: Asigna un nombre al elemento.
 - **Size='n'**: anchura de la caja de texto

Elementos

- Los elementos **submit**, **button** y **reset** insertan en el formulario botones.
 - Submit: envía el formulario para que sea manejado por el programa indicado en 'action'
 - Reset: resetea el formulario dejándolo con los valores por defecto
 - Button: No hace nada. Su uso más frecuente es para la programación de eventos desde el lado del cliente.
- Sus parametros son:
 - **name='nombre'**. Asigna un nombre al botón,
 - **value='valor'**. Define el texto que veremos en el botón
 - **disabled**. Desactiva el botón, cuando se pulsa no se produce ninguna acción..
 - **tabindex = 'n'**. Especifica el orden de tabulador que tendrá el campo respecto todos los elementos que incluye el formulario.

Listas desplegables

- ❑ La etiqueta **<select>** crea una lista desplegable con varias opciones.
- ❑ Atributos de select:
 - ❑ **Name='nombre'**: nombre de la lista desplegable
 - ❑ **Size='n'**: Número de filas visibles en la lista
 - ❑ **Disable**: Desactiva la lista y no se podrá elegir ninguna opción
- ❑ Entre **<select>** y **</select>** aparecerán las etiquetas **<option>** y **</option>** que marcan cada opción de la lista.
 - ❑ **Selected**: Especifica la opción predeterminada
 - ❑ **Value='valor'**: Especifica el valor del elemento elegido

Listas desplegables

```
<form name = 'formulario1'>
  <select name='ciudades'>
    <option value=0 selected> selecciona tu ciudad de nacimiento </option>
    <option value='gra'> Granada </option>
    <option value='sev'> Sevilla </option>
    <option value='mal'> Malaga </option>
    <option value='alm'> Almería </option>
    <option value='cor'> Córdoba </option>
    <option value='jae'> Jaén </option>
    <option value='hue'> Huelva </option>
    <option value='cad'> Cádiz </option>
  </select>
</form>
```

selecciona tu ciudad de nacimiento ▼

selecciona tu ciudad de nacimiento ▼

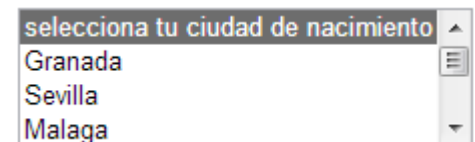
selecciona tu ciudad de nacimiento

- Granada
- Sevilla
- Malaga
- Almería
- Córdoba
- Jaén
- Huelva
- Cádiz

Listas desplegables

- Utilizando el atributo “size” podemos indicar el número de líneas que queremos visualizar desde el principio:

```
<form name = 'formulario1'>
  <select name='ciudades' size=3>
    <option value=0 selected> selecciona tu ciudad de nacimiento </option>
    <option value='gra'> Granada </option>
    <option value='sev'> Sevilla </option>
    <option value='mal'> Malaga </option>
    <option value='alm'> Almería </option>
    <option value='cor'> Córdoba </option>
    <option value='jae'> Jaén </option>
    <option value='hue'> Huelva </option>
    <option value='cad'> Cádiz </option>
  </select>
</form>
```



selecciona tu ciudad de nacimiento	
Granada	
Sevilla	
Malaga	

Manejo de formularios

Envío de formularios al servidor

Envío de formularios

- El atributo 'method' especifica la forma en que el protocolo HTTP enviará los datos del formulario al servidor.
- Puede tener dos valores:
 - Get
 - El conjunto de datos se agrega a la URL especificada en 'action'
 - Se utiliza el carácter '?' como separador.
 - Post
 - Los datos se incluyen en el cuerpo del mensaje enviado.

Envío de formularios

- El atributo 'action' indica quién se va a hacer cargo de los datos que se recogen en el formulario
- Una vez enviado el formulario, se activa el programa que gestiona dicho formulario:
 - Se envía un e-mail si
action = 'mailto:jjj@llkjo.com'
 - Se ejecuta PHP si
action = 'documento.php'

Envío de formularios

- El conjunto de datos que envía un formulario (al pulsar el botón 'submit') es un array asociativo:
 - Posición: Nombre del elemento del formulario
 - Contenido: Valor de dicho elemento.

Ejemplo

Formulario

```
<form name = 'form' method = 'get' action = 'prueba.php'>
  <input name = 'nombre' type='text'></td>
  <input name = 'edad' type='text'></td>
  <input name='enviar' type = 'submit'>
</form>
```

Array enviado

Nombre	Edad	Enviar
'Juan'	35	'Enviar'

Envío de formularios

- ¿Cómo se accede al array generado por un formulario?
- Dependiendo del método de envío que hayamos utilizado se generan dos arrays distintos:

ARRAY	CONTENIDO
<code>_POST</code>	Array con las variables pasadas a través del método POST
<code>_GET</code>	Array con las variables pasadas a través del método GET

- Puntualmente también se podría usar un array genérico:

ARRAY	CONTENIDO
<code>_REQUEST</code>	Array con las variables pasadas a través del método POST y GET

Manejo de formularios

Ejemplo de procesamiento con GET

```
<form name = 'contacto' method = 'get' action = 'formulario1.php'>
  Nombre
  <input name = 'nombre' type='text'>
  Apellidos
  <input name = 'apellidos' type='text'>
  Correo electrónico
  <input name = 'email' type='text'>
  ¿Qué apartado te ha gustado más?</td>
  <select name="seccion">
    <option value = '0' selected>--- Elige seccion ---</option>
    <option value='presentacion'>Presentación</option>
    <option value='intro'>Introducción</option>
    <option value='texto'>Formato de texto</option>
    <option value='enlaces'>Hiperenlaces</option>
    <option value='imagenes'>Imagenes</option>
  </select>
  ¿Desea suscribirse a nuestro servicio de noticias?
  Si <input name='suscripcion' type = 'radio' value = 'si' checked>
  No <input name='suscripcion' type = 'radio' value = 'no'>
  Escriba aquí su consulta
  <textarea name = 'consulta' rows='10' cols='30'>Consulta...</textarea>

  <input name='enviar' type = 'submit'>
  <input name='cancelar' type = 'reset'>
</form>
```

Ejemplo de manejo (Documento php)

```
<?php
```

```
$nombre = $_GET["nombre"];  
$apellidos = $_GET["apellidos"];  
$mail = $_GET["email"];  
$apartado = $_GET["seccion"];  
$suscribirse = $_GET["suscripcion"];  
$consul = $_GET["consulta"];
```

Ejemplo de manejo (Documento php)

```
if ($nombre)
{
    echo "Hola $nombre";
}
if ($apellidos)
{
    echo " $apellidos.<br>";
}
```

Ejemplo de manejo (Documento php)

```
switch ($apartado)
{
    case "intro":
        echo "Sección preferida: Introducción.";
        break;
    case "presentacion":
        echo "Sección preferida: Presentación.";
        break;
    case "texto":
        echo "Sección preferida: Formato de texto.";
        break;
    case "enlaces":
        echo "Sección preferida: Enlaces.";
        break;
    case "imagenes":
        echo "Sección preferida: Imágenes.";
        break;
    default:
        echo "No tienes sección preferida.";
}
```

Ejemplo de manejo (Documento php)

```
if ($mail and $suscribirse=='si')
{
    echo "<br><br>Has sido suscrito a nuestro canal de noticias";
    echo "<br> Dirección de suscripción: $mail<br><br>";
}
if ($consul != "Consulta...")
{
    echo "<br>Tu consulta: $consul <br>será pasada a nuestros expertos";
    echo "y esperamos responderte a la mayor brevedad posible";
    if ($mail)
    {
        echo "<br> La respuesta la enviaremos a la dirección de correo: $mail";
    }
    else
    {
        echo "<br> Sería necesario una forma de ponernos en contacto con";
        echo " usted para poder enviarle la respuesta";
    }
}
```

Nombre

Apellidos

Correo electrónico

¿Qué apartado te ha gustado más?

¿Desea suscribirse a nuestro servicio de noticias?

Si ☒ No ☐

Escriba aquí su consulta

Consulta...

Enviar

Restablecer

Nombre

chari

Apellidos

vargas

Correo electrónico

eag.chari@gmail.com

¿Qué apartado te ha gustado más?

Introducción ▼

¿Desea suscribirse a nuestro servicio de noticias?

Si ☒ No ☐

Escriba aquí su consulta

¿Cómo se usan los enlaces?

Enviar

Restablecer

Ejemplo de manejo (Ejecución)

Hola chari vargas.

Sección preferida: Introducción.

Has sido suscrito a nuestro canal de noticias

Dirección de suscripción: eag.chari@gmail.com

Tu consulta: ¿Cómo se usan los enlaces?

será pasada a nuestros expertos y esperamos responderte a la mayor brevedad posible

La respuesta la enviaremos a la dirección de correo: eag.chari@gmail.com

Muchas gracias por visitar mi página

Ejercicios

Ejercicios

- Programar una página en HTML – PHP que a través de formularios pida al usuario su nombre y le muestre un saludo personalizado.
- Programar una página en HTML – PHP que a través de formularios pida al usuario dos números y muestre en una tabla el resultado de:
 - Sumarlos
 - Restarlos
 - Multiplicarlos
 - Dividirlos

ENVÍO DE ARCHIVOS

Manejo de Formularios

Envío de archivos

Utilización de `<input file>`

Ficheros adjuntos

- Si queremos que se pueda subir un archivo al servidor, debemos añadir obligatoriamente al formulario el tipo de encriptación:
 - `enctype="multipart/form-data"`
- Además, el método de envío no podrá ser GET, deberá ser **POST**
- Dentro del formulario utilizaremos el tipo file.
 - Añade un botón con el que se abre un explorador para añadir el archivo.

Ficheros adjuntos

- En el documento .php se genera un nuevo array (tipo matriz) llamado **_FILES** que contiene una línea por cada elemento tipo file del formulario
- **Cada fila** de la matriz tiene como nombre el **name** del elemento y las siguientes posiciones:
 - Name: nombre del archivo en el ordenador cliente
 - Type: tipo de archivo
 - Tmp_name: nombre temporal del archivo en el servidor
 - Error: true o false (1 o 0)
 - Size: tamaño en bytes del archivo subido

Ficheros adjuntos

- El campo type, contendrá los tipo MIME, no las extensiones:
 - Imagen en jpg: image/jpeg
 - Imagen en png: image/png
 - Imagen en gif: image/gif
 - Documentos pdf: application/pdf
 - Documentos Word 97 – 2003: application/msword
 - Documentos contexto plano (.txt): text/plain

Más información sobre los tipo MIME en:

https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

Ficheros adjuntos

- `move_uploaded_file($origen, $destino);`
 - Esta función mueve un archivo subido a una nueva ubicación.
 - `$origen`: nombre temporal del archivo subido
 - `$destino`: ruta completa, o relativa, de la nueva ubicación del archivo.

Importante: el directorio donde se vaya a colocar el archivo debe existir

Ficheros adjuntos

- ❑ `file_exists($path);`
 - ❑ Comprueba si un directorio existe o no.
 - ❑ Devuelve true o false
- ❑ `mkdir($path);`
 - ❑ Crea una carpeta en la ruta especificada.
- ❑ `scandir ($path)`
 - ❑ Permite escanear un directorio.
 - ❑ Devuelve una array posicional con todos los archivos y carpetas dentro del directorio indicado ordenados por orden alfabético

0	1	2	3	5
.	..	Foto1.jpg	Foto2.jpg	Fotos viejas

Ejemplo “Envío de archivos”

Formulario de envío

```
<html>
  <body>
    <form action="procesar.php" method="post" enctype="multipart/form-data">
      Archivo
      <input type="file" name="foto"/>
      <input type="submit" name="button" value="Enviar" />
    </form>
  </body>
</html>
```

`$_FILES['foto']`

name	type	tmp_name	error	size
perro.jpg	image/jpeg	c:\wamp\tmp\php3514.tmp	0	55370

Formulario de envío

```
<html>
  <body>
    <form action="procesar.php" method="post" enctype="multipart/form-data">
      Archivo 1
      <input type="file" name="archivo1">
      Archivo 2
      <input type="file" name="archivo2">
      <input type="submit" name="button">
    </form>
  </body>
</html>
```

\$_FILES

fila	name	type	tmp_name	error	size
archivo1	perro.jpg	image/jpeg	c:\xampp\tmp\php3514.tmp	0	55370
archivo2	Curriculum.pdf	application/pdf	c:\xampp\tmp\php3243.tmp	0	2048

Procesamiento del documento

```
<html>
<body>

<?php
    $name = $_FILES['foto']['name']; // nombre original del archivo
    $temp = $_FILES['foto']['tmp_name']; //nombre temporal del archivo en el servidor

    // Comprobamos si existe la carpeta donde queremos guardar la foto
    if(!file_exists("./img"))
    {
        mkdir("./img");
    }

    // Guardamos el archivo en el directorio del servidor
    $ruta = "img/$name";
    move_uploaded_file ($temp, "$ruta");
?>

" />
</body>
</html>
```

Ejercicios propuestos

- Realizar un documento en HTML – PHP que solicite al usuario una imagen y un nombre para la imagen (no tiene por qué coincidir con el nombre original). El documento deberá guardar la imagen con el nuevo nombre y mostrar por pantalla la información de la imagen subida por el usuario en una tabla como la siguiente:

Nombre original	Nuevo nombre	Tamaño en megas	Tipo de imagen
...

Varios formularios juntos

Procesamiento de html con varios formularios

Varios formularios

- En ocasiones nos va a interesar tener varios formularios en el mismo documento HTML
- Será necesario saber a ciencia cierta qué formulario ha rellenado el usuario
- Utilizaremos el nombre del botón “enviar” para saber qué formulario se ha enviado
- **isset(variable)**
 - Comprueba si la variable existe
 - Devuelve true o false

Manejo de varios formularios

```
<body>
FORMULARIO PARA HOMBRES
  <form action='ej_dos.php' method='post' enctype='multipart/form-data'>
    <b>Nombre: </b>
    <br>
    <input type='text' name='nombre_hombre'>
    <br>
    <b>Edad: </b>
    <br>
    <input type='text' name='edad_hombre'>
    <br>

    <input type='submit' name='enviar_hombre'>
  </form>
<hr>
FORMULARIO PARA MUJERES
  <form action='ej_dos.php' method='post' enctype='multipart/form-data'>
    <b>Nombre: </b>
    <br>
    <input type='text' name='nombre_mujer'>
    <br>
    <b>Edad: </b>
    <br>
    <input type='text' name='edad_mujer'>
    <br>

    <input type='submit' name='enviar_mujer'>
  </form>
```

Manejo de varios formularios

```
<?php
    if(isset($_POST['enviar_hombre']))
    {
        $nombre = $_POST['nombre_hombre'];
        $edad = $_POST['edad_hombre'];
        echo "Hola $nombre, eres hombre y tienes $edad años";
    }
    elseif (isset($_POST['enviar_mujer']))
    {
        $nombre = $_POST['nombre_mujer'];
        $edad = $_POST['edad_mujer'];
        echo "Hola $nombre, eres mujer y tienes $edad años";
    }
?>
```

Formularios internos

Procesar un formulario en el mismo documento

Formularios internos

- En ocasiones necesitaremos procesar un formulario en la misma página en la que lo hemos escrito.
- Si en el atributo **action** de un formulario ponemos **#**, el formulario será tratado en el mismo documento.
- Habrá que comprobar cómo se ha ejecutado el documento, es decir, si viene o no de una ejecución de un formulario.
- La función **isset** comprueba si una variable cualquiera existe o no.

Formularios internos

```
<?php
if (isset($_POST['enviar']))
{
    procesamiento....
}
```

```
?>
<form method="post" action="#">
    nombre <input type="text" name="name"><br>
    elementos ...
        <input type="submit" name="enviar"><br>
</form>
```

Creando nuestro propio GET

Creando nuestro propio GET

- En ciertas situaciones necesitaremos adjuntar información a una URL de forma que el usuario no se dé cuenta de qué pasa.
- Para eso podemos utilizar la forma en que GET modifica la URL añadiendo los valores del array `$_GET` de la siguiente forma:
 - `url?posicion=valor&posicion=valor&posicion=valor`

Creando nuestro propio GET

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  Opcion 1:
  <a href='colores.php?c=verde'> color verde </a>
  Opcion 2:
  <a href='colores.php?c=rojo'> color rojo </a>
  Opcion 3:
  <a href='colores.php?c=amarillo'> color amarillo </a>
</body>
</html>
```

```
<?php
    $color = $_GET['c'];

    echo "has elegido el color $color";

?>
```

Información extra

Redireccionamiento

Uso de HEADER

- header (\$cadena)
 - La función header de php nos permite redireccionar de varias formas al usuario.
 - Una de las formas es utilizando la cabecera “Location”.
 - Con location redireccionamos al cliente a la página que queramos

```
header (“location: http://www.google.es”)
```

Uso de HEADER

- Es necesario tener en cuenta que:
 - ▣ Una vez que php lea la función header, dejará sin ejecutar todo lo que haya por debajo.
 - ▣ La función header no tiene retraso, con lo que si queremos mostrar un mensaje antes de hacer la redirección será mejor utilizar la cabecera “refresh”

```
Header("refresh:3;url= http://www.google.es");
```

Uso de META

- El problema de utilizar header es que lo que estamos modificando es la cabecera del documento HTML.
- Si hemos escrito algún carácter antes, header ya no surtirá efecto.
- En ese caso utilizaremos la etiqueta META de la siguiente forma:

```
<META HTTP-EQUIV="REFRESH"  
CONTENT="5;URL=http://www.google.es">
```

Combinando información

- Ahora, al redirigir al usuario podremos enviarnos un «código de error» con el que controlar qué ha pasado

```
Header("refresh:3;url=mipagina.php?e=1");
```

```
<META HTTP-EQUIV="REFRESH"  
CONTENT="5;URL=mipagina.php?e=1">
```

MANEJO DE FORMULARIOS

Programando con PHP