

Desarrollo de sitios web con PHP y MySQL

Formularios



Formularios

- Los formularios permiten al usuario introducir información y, por lo tanto, interactuar con el sitio.

Un formulario HTML se define entre las etiquetas **<form>** y **</form>**.

```
<form [action="url_de_procesamiento"][method="GET"|"POST"][  
id="identificador_formulario"][target="destino"]>
```

```
...  
</form>
```

Acceso a formularios desde PHP

- Acceso a los diferentes tipos de elementos de entrada de formulario
 - Elementos de tipo INPUT
 - TEXT
 - RADIO
 - CHECKBOX
 - BUTTON
 - FILE
 - HIDDEN
 - PASSWORD
 - SUBMIT
 - Elemento SELECT
 - Simple / múltiple
 - Elemento TEXTAREA

Método GET y POST

Son métodos de protocolo HTTP para el intercambio de la información entre el cliente y el servidor.

GET es el método mas usado actualmente.

Este método le “pide” al servidor web que le devuelva al cliente la información identificada en la petición URL.

Lo mas común es que las peticiones URI se refieran a un documento HTML o a una imagen, aunque también puede ser una consulta a una base de datos.

Mientras el método **GET** lo utilizamos para recuperar información, el método **POST** se usa para enviar información a un servidor web.

Diferencia POST y GET

La principal diferencia radica en cómo codifican la información :

El GET envía las variables dentro de la URL de la pagina

Para entender como se envían las variables con GET es necesario entender

Partes URL

http://

www.ejempl.org

/ficheros/

Ejemplo1.html

Protocolo

Dominio

Directorio

Fichero



Además al final de esta podemos concatenar el **símbolo ?** , que se utiliza para indicar que se van a definir a continuación variables con el valor que toman (variable=valor).

Las distintas variables se separan con **&**.

```
/path/to/page.php?keyword=bell&length=3
```

POST

En el método POST la información va codificada en el cuerpo de la petición HTTP y, por tanto viaja oculta. Por eso hay que usar este método cuando queramos ingresar datos en un formulario, ya sea para la autenticación de un usuario y/o actualización en una base de datos.

FORMULARIOS

El lenguaje PHP nos proporciona una manera sencilla de manejar formularios, permitiéndonos de esta manera procesar la información que el usuario ha introducido.

Al diseñar un formulario debemos indicar la página PHP que procesará el formulario, así como en método por el que se le pasará la información a la página.

NOCIONES BÁSICAS

Los formularios html van encerrados entre etiquetas **<FORM> </FORM>**.

Esta etiqueta admite diversos atributos, entre ellos los mas importantes:

ACTION que indica la dirección del script que va a procesar la información y

METHOD que indica la forma de envío de los datos, "post" y "get"

el atributo **method**

especifica el método usado para enviar la información.

Este atributo puede tener dos valores:

- **get**

con este método los datos del formulario se agregan al URI utilizando un signo de interrogación "?" como separador y detrás vendrán las variables concatenadas con & con el formato nombreVariable=valor.

- **post**

con este método los datos se incluyen en el cuerpo del formulario y se envían utilizando el protocolo HTTP.

Dentro de nuestro formulario podemos usar diversos elementos html: INPUT o SELECT, cuyos valores son rellenos por el visitante de la página.

Al enviar el formulario al script indicado por el atributo ACTION, automáticamente se genera una variable cuyo nombre es el especificado con el atributo NAME y cuyo valor es el introducido por el usuario.

Parámetro	Descripción
Action	URL relativa o absoluta (Uniform Resource Locator) que procesará el formulario, en nuestro caso, un script PHP. Este atributo es obligatorio para cumplir con la estricta recomendación XHTML.
Method	Modo de transmisión al servidor de la información introducida en el formulario. GET (valor predeterminado): los datos del formulario se transmiten en la URL. POST: los datos del formulario se transmiten en el cuerpo de la consulta.
Id	Identificador del formulario. Si la página HTML contiene varios formularios, el identificador permite diferenciarlos. En nuestro caso, este identificador no tiene ningún valor porque no se recupera en el script de procesamiento del formulario. Por contra, se puede utilizar del lado del cliente, en JavaScript, por ejemplo.
Target	Destino (por ejemplo, otra ventana) en el que se abrirá la dirección URL de destino.

Estructura general de un formulario

<FORM ACTION = "dirección" METHOD = "POST|GET">

- distintos elementos para poder introducir los datos
- Botones de envío y borrado

</FORM>

<FORM ACTION="ft2.php" METHOD="POST">

Edad: <INPUT TYPE="text"
NAME="edad">

<INPUT TYPE="submit"
VALUE="aceptar">

</FORM>

ENVÍO Y RECEPCIÓN DE DATOS

Al diseñar un formulario debemos indicar la página PHP que procesará el formulario, así como en método por el que se le pasará la información a la página.

```
<html>
<head>
  <title>ejemplo de php</title>
</head>
<body>
<h1>ejemplo de procesamiento de formularios</h1>
introduzca su nombre:
<form action="procesa.php" method="get">
<input type="text" name="nombre"><br>
<input type="submit" value="enviar">
</form>
</body>
</html>
```

página PHP que
procesará el
formulario

método pasará la
información a la
página


Al pulsar el botón **Enviar** el contenido de cuadro de texto es enviado a la página que indicamos en el atributo ACTION de la etiqueta FORM.

Para acceder a las variables del formulario hay que usar el array de parámetros `$_POST[]` o `$_GET[]` dependiendo del método usado para enviar los parámetros.

(Fichero configuración php.ini)

En este ejemplo se ha creado una entrada en el **array `$_GET[]`** con el índice '**nombre**' y con el valor que haya introducido el navegador.

```
<html>
<head>
  <title>ejemplo de php</title>
</head>
<body>
<h1>ejemplo de procesamiento de formularios</h1>
el nombre que ha introducido es: <?php echo
$_GET['nombre'] ?>
<br>
</body>
</html>
```



Nos muestra el valor introducido
usuario que estará en la entrada que
se ha creado **array \$_GET[]** con el
índice '**nombre**'

Ejemplo de procesamiento de formularios

Introduzca su nombre:

Ejemplo de procesamiento de formularios

El nombre que ha introducido es: maria

GET.- consigue los datos a través de la URL

Ejemplos:

- 1.- Creamos nuestra página inicio.php, la cual va a tener dos cajas de texto y un botón.
- 2.- Creamos otro archivo .php llamado enviarget.php, el cual va a capturar los datos de nuestra inicio.php y mostrarlos.

inicio.php

```
1 <form id="form1" name="form1" method="get" action="enviarget.php">
2 <p>Nombre:<br />
3 <label>
4   <input name="nombre" type="text" id="nombre" />
5 </label>
6 <br />
7 Telefono:<br />
8 <input name="telefono" type="text" id="telefono" />
9 <br />
10 Email:<br />
11 <input name="email" type="text" id="email" />
12 <br />
13 Mensaje:<br />
14 <textarea name="mensaje" id="mensaje"></textarea>
15 <br />
16 <label>
17   <input type="submit" name="Submit" value="Enviar Formulario" />
18 </label>
19 </p>
20 </form>
21
```


Nombre:

maria

Telefono:

98777777

Email:

jjjj@gmail.com

Mensaje:

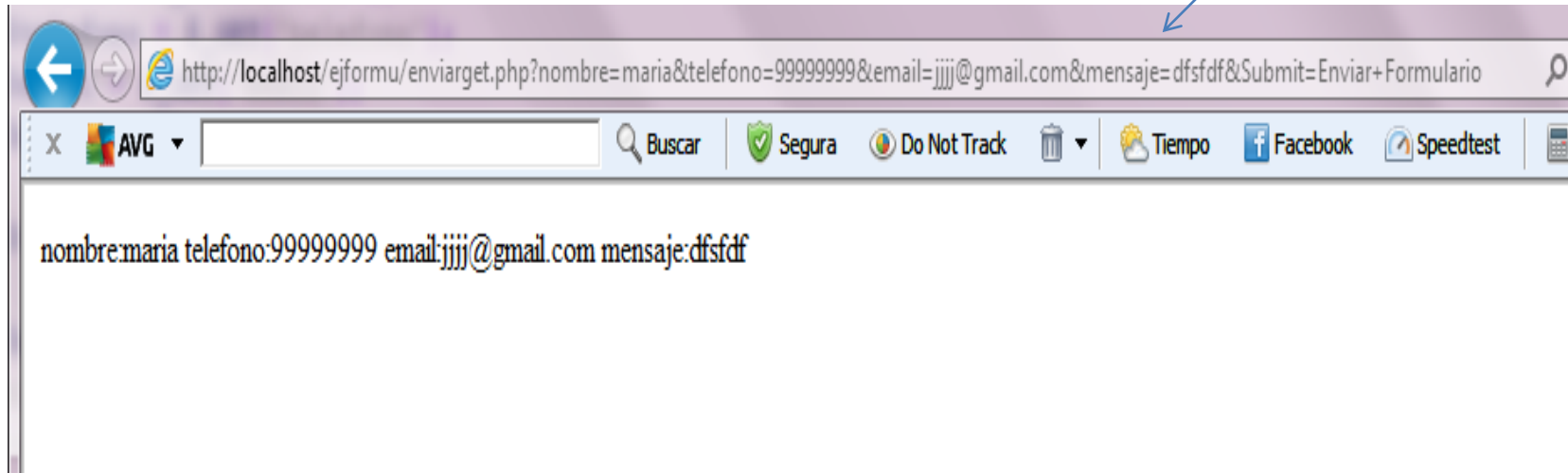
kkkkkkkk

Enviar Formulario

enviarget.php

```
1  <?php
2  $nombre = $_GET['nombre'];
3  $telefono = $_GET['telefono'];
4  $email = $_GET['email'];
5  $mensaje = $_GET['mensaje'];
6
7  $thank="gracias.html";
8
9  $mostrar = "
10 nombre:". $nombre ."
11 telefono:". $telefono ."
12 email:". $email ."
13 mensaje:". $mensaje ." ";
14 echo $mostrar;
15
16 ?>
17
```

Mirar URI



Al dar clic en el botón envía podemos darnos cuenta que la URL muestra los datos que se ingresamos en las cajas de texto.

- **POST.-** Oculta los valores que pasamos de una página a otra, esto lo hace más seguro.

Ejemplo:

Para este ejemplo cambiamos en nuestra inicio.php en la propiedad ***method*** del formulario el ***get*** por ***post***.

- En nuestro código de **enviar.php** cambiamos la forma de capturar los datos de ***\$_GET*** a ***\$_POST***.

```

1 <form id="form1" name="form1" method="post" action="enviar.php">
2 <p>Nombre:<br />
3 <label>
4 <input name="nombre" type="text" id="nombre" />
5 </label>
6 <br />
7 Telefono:<br />
8 <input name="telefono" type="text" id="telefono" />
9 <br />
10 Email:<br />
11 <input name="email" type="text" id="email" />
12 <br />
13 Mensaje:<br />
14 <textarea name="mensaje" id="mensaje"></textarea>
15 <br />
16 <label>
17 <input type="submit" name="Submit" value="Enviar Formulario" />
18 </label>
19 </p>
20 </form>
21

```

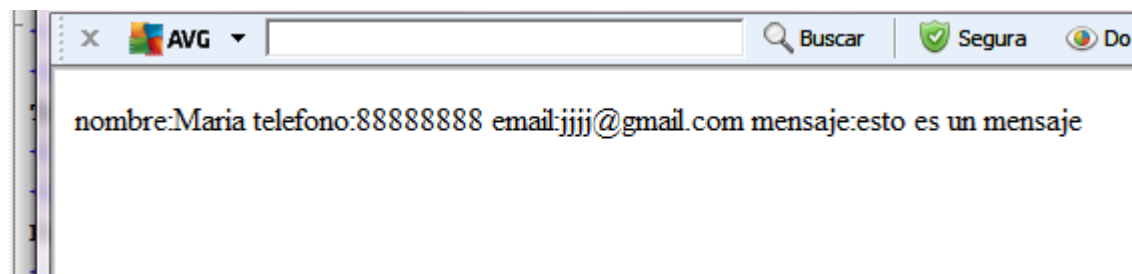
Nombre:

Telefono:

Email:

Mensaje:


```
1 <?php
2 $nombre = $_POST['nombre'];
3 $telefono = $_POST['telefono'];
4 $email = $_POST['email'];
5 $mensaje = $_POST['mensaje'];
6
7 $thank="gracias.html";
8
9 $mostrar = "
10 nombre:". $nombre. "
11 telefono:". $telefono. "
12 email:". $email. "
13 mensaje:". $mensaje. " ";
14 echo $mostrarmensaje;
15
16 ?>
```



Al dar clic en el botón Enviar nuestra URL no mostrara los datos enviados a enviar.php, ya que la captación de los valores se la hace de forma oculta

- **id**: lo usamos para identificar el formulario, ya que podemos llegar a tener muchos formularios en una misma página y queremos que se identifique bien cada formulario. Aquí elegimos el nombre que queremos.

```
<form action="contacto_script.php"  
method="post" id="contacto"></form>
```



Recuperar datos de un formulario

Fichero uno.php

```
<html>
<body>
<form action="dos.php" method="post">
  Su nombre: <input type="text" name="nombre">
  Su edad: <input type="text" name="edad">
  <input type="submit" value="aceptar">
</form>
</body>
</html>
```

Fichero dos.php

```
<html>
<body>
<?php
$nombre=(isset($_POST['nombre']))?htmlspecialchars($_POST['nombre']):'';
$edad=(isset($_POST['edad']))?(int)$_POST['edad']:'';
print ("Hola: $nombre")
print ("su edad es: $edad");
?>
</body>
</html>
```

Recuperar datos de un formulario

- Todos los campos del formulario se almacenan en una matriz asociativa `$_POST` o `$_GET` según el valor de `method`

Ejemplo: `$_POST['nombre']`

- La clave de la matriz es igual al nombre del campo en el formulario (atributo `name` de la etiqueta `<input>`, `<select>` o `<textarea>`) y el valor igual al valor introducido en el campo.

Ejemplo: `<input type="text" name="nombre">`

- Esta información también está disponible en la matriz asociativa `$_REQUEST` que agrupa el contenido de las matrices `$_GET` y `$_POST`

La función isset()

- Esta función nos permite comprobar si una variable se ha definido y en ese caso devuelve un True.
- La sintaxis es **isset(\$variable)**

Ejemplo: `if (isset($_POST['nombre'])) { ... }`

- Muy útil para comprobar si se han rellenado los campos de un formulario

La función empty()

- **No permite a saber si una variable está vacía.**
"Una variable se considera vacía si no existe o si su valor es igual a FALSE. empty() no genera una advertencia si la variable no existe
- **La sintaxis es empty(\$var)**
Ejemplo if(empty(\$_POST['nombre'])){...}
- **Muy útil para saber si una variable está definida y no es NULL.**

isset / empty

isset y empty evalúan una variable de manera **diferente**.

isset si una variable es null,

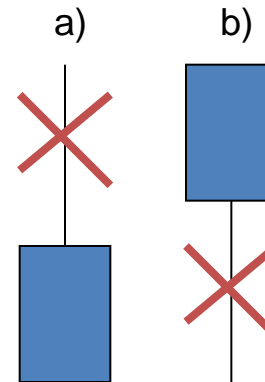
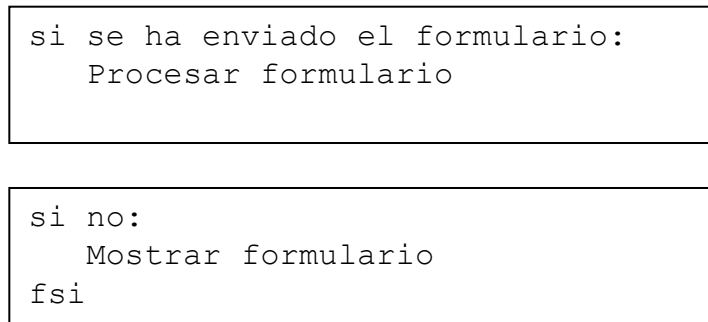
empty si está vacía pudiendo ser NULL

```
$var = 0;  
  
if (isset($var))  
    echo 'La variable está definida';  
  
if (empty($var))  
    echo 'La variable está definida pero está vacía';
```

- Al cargarse la página usamos **empty()** para comprobar si se ha enviado el formulario, y en tal caso mediante **isset()** comprobamos si hemos recibido todos los datos requeridos.
- El **signo de exclamación** es usado en PHP como **operador de negación**, por lo tanto con el código **if(! empty(\$_POST))** estamos comprobando "si \$_POST NO está vacío".

EL FORMULARIO DE PHP

- Esquema de funcionamiento:



- La 1ª vez que se carga la página se muestra el formulario (a)
- La 2ª vez se procesa el formulario (b)

Para saber si se ha enviado el formulario se acude a la variable correspondiente al botón de envío. Si este botón aparece de la siguiente forma en el formulario HTML:

```
<INPUT TYPE="SUBMIT" NAME="enviar" VALUE="procesar">
```

entonces la condición anterior se transforma en:

```
if (isset($enviar))  
o bien
```

```
if ($enviar == "procesar")
```

- Ejemplo (tanto el código PHP como el código HTML estarán en el mismo archivo .php):

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Ejemplo de en una única página tener el Formulario y su tratamiento</title>
<meta charset="utf-8">
</head>
<body>
<?php
if(!isset($_POST["enviar"])){//Si todavía no he pulsado enviar visualizo el formulario
?>
    <form action="q1.php" method="POST">
        Introduzca su nombre:
        <input type="text" name="nombre" id="nombre">
        <input type="submit" name="enviar" value="Enviar">
    </form>
<?php
}
else{//Si ya he pulsado Enviar hago el tratamiento
    echo "Hola, ".$_POST['nombre'];
}
?>
</body>
</html>
```

Introduzca su nombre:

Enviar

Hola, Juan

VALIDACIÓN

- Ejemplo

<https://diego.com.es/formularios-en-php>

ELEMENTOS DE ENTRADA EN UN FORMULARIO WEB.

Distintos elementos `<input>`

Para insertar un elemento de entrada es necesario incluir la etiqueta `<input>` entre las etiquetas `<form>` y `</form>` del formulario.

El atributo **name** indica el nombre que se desea dar al elemento de entrada, mediante el cual será

Evaluable

El atributo **type** indica el tipo de elemento de entrada.

Vamos a ver varios de los diferentes tipos de elementos de entrada, y el resto de atributos que pueden definirse para cada uno de ellos.

Acceso a los diferentes tipos de elementos de entrada de formulario

-Elementos de tipo INPUT

- TEXT
- RADIO
- CHECKBOX
- RESET
- BUTTON
- FILE
- HIDDEN
- PASSWORD
- SUBMIT
- IMAGE

-Elemento SELECT

- Simple / múltiple

-Elemento TEXTAREA

Elemento	Descripción
input type="text"	Caja de texto
input type="password"	Caja de texto donde se muestran asteriscos en lugar de los caracteres escritos
input type="checkbox"	Cajas seleccionables que permite escoger múltiples opciones
input type="radio"	Cajas seleccionables en grupos que sólo permiten escoger una opción
input type="submit"	Botón para enviar el formulario
input type="file"	Cajas de texto y botón que permite subir archivos
input type="hidden"	Elemento escondido. Especialmente útil para tokens de seguridad
option	Una opción posible dentro de un elemento element
select	Lista de opciones de elementos option
textarea	Texto mixtilínea

CAMPO DE TEXTO

Para insertar un campo de texto, el atributo type debe tener el valor text.

El atributos:

size .- número de caracteres que podrán ser visualizados en el campo de texto, determina el ancho de la caja.

maxlength número de caracteres que podrán ser insertados en el campo de texto.

value el valor inicial del campo de texto.

Por ejemplo, para insertar un campo de texto:

```
<input name="campo" type="text" value="Campo de texto" size="20" maxlength="15">
```



```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Ejemplo de en una única página tener el Formulario y su tratamiento</title>
<meta charset="utf-8">
</head>
<body>
<?php
if(!isset($_POST["enviar"])){//Si todavía no he pulsado enviar visualizo el formulario
?>
<form action="Formtexto.php" method="POST">
Introduzca su nombre:
<input type="text" name="nombre" id="nombre">
<input type="submit" name="enviar" value="Enviar">
</form>
<?php
}
else{//Si ya he pulsado Enviar hago el tratamiento
echo "Hola, ".$_POST['nombre'];
}
?>
</body>
</html>

```



Hola, Ana

Introduzca su nombre:

TEXT

Introduzca la cadena a buscar:

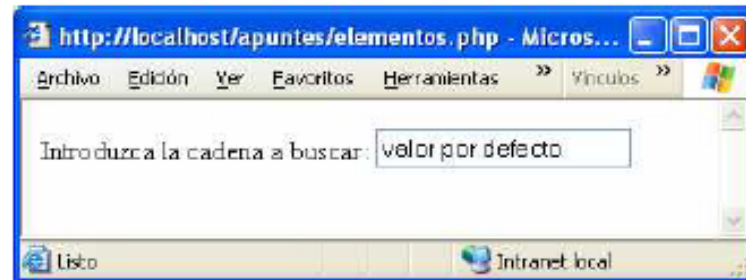
```
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">
```

```
<?PHP
```

```
    $cadena = $_REQUEST['cadena'];
```

```
    print ($cadena);
```

```
?>
```



He puesto \$_REQUEST porque no se qué método se ha empleado.

Ejemplo de cómo podemos hacerlo utilizando 2 páginas distintas


```
<html>
<head>
<title>Formulario de entrada del dato</title> </head>
  <body> <form method="post" action="pagina2.php">
Ingrese su nombre:
  <input type="text" name="nombre">
  <br>
<input type="submit" value="confirmar">
</form>
</body>
</html>
```

Ingrese su nombre:

confirmar

Pequeño programa en PHP que procese los datos ingresados en el formulario:

```
<html>
<head>
<title>Captura de datos del form</title> </head>
<body>
<?php
echo "El nombre ingresado es:";
echo $_REQUEST['nombre'];
?>
</body>
</html>
```

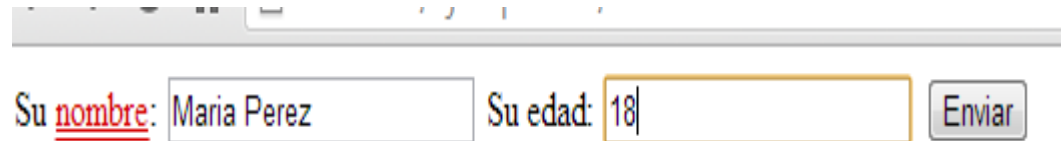


Como no sabemos el
método empleado

Para acceder al dato en PHP se cuenta con un vector llamado **\$_REQUEST** indicando como subíndice el nombre del cuadro de texto que definimos en el formulario (dicho nombre es sensible a mayúsculas y minúsculas)

- Un formulario HTML sencillo

```
<html>
<head>
<title>Formulario de entrada del dato</title> </head>
  <body>
    <form action="accion.php" method="POST">
      Su nombre: <input type="text" name="nombre" />
      Su edad: <input type="text" name="edad" /> <input type="submit">
    </form>
  </body>
</html>
```



A screenshot of a web browser window displaying the rendered HTML form. The form consists of two text input fields and a submit button. The first input field is labeled 'Su nombre:' and contains the text 'Maria Perez'. The second input field is labeled 'Su edad:' and contains the number '18'. To the right of the second input field is a button labeled 'Enviar'.

- **Procesamiento de información de nuestro formulario HTML**

```
<html>
Hola

<?php
$nombre=$_POST["nombre"];
echo $nombre;?>
tienes

<?php
$edad=$_POST["edad"];
echo $edad;?>
años

</html>
```

```
<?php
$nombre=$_POST["nombre"];
echo "Hola $nombre tienes ";
$edad=$_POST["edad"];
echo $edad. "años";?>
```

← → ↺ ↻ localhost/e

Hola Maria Perez tienes 18 años

```
<html>
<head>
<title>Formulario dos en uno</title>
</head>
<body>
<?php
    if (!(isset($_POST['enviar'])))
    { //No se ha enviado el formulario
        ?>

        <form action="edad.php" method="POST">
        Su nombre: <input type="text" name="nombre">
        Su edad: <input type="text" name="edad" >
        <input type="submit">
        </form>
    <?php
    }else
    { //Si se ha enviado el formulario
        $nombre=$_POST["nombre"];
        echo "Hola $nombre tienes ";
        $edad=$_POST["edad"];
        echo $edad. "años";
    }
    ?>

</body>
</html>
```


BOTÓN DE OPCIÓN

Para insertar un botón de opción, el atributo `type` debe tener el valor `radio`.

- **value** indica el valor asociado al botón de opción. Es necesario poner este atributo, aunque el usuario no pueda ver su valor. Es el valor a enviar.
- **checked** indica que el botón aparecerá activado inicialmente. Este atributo no toma valores.

Los botones de opción se utilizan cuando se desea que una variable del formulario pueda tomar un solo valor de entre varios posibles. Para ello, se insertan varios botones de opción con el mismo nombre (que indica la variable) y con distintos valores. Solamente uno de estos botones podrá estar activado, el que esté activado cuando se envía el formulario, su valor será el que tendrá la variable.

Por ejemplo, para insertar los botones de opción:

```
<input name="prefiere" type="radio" value="estudiar"  
checked>
```

```
<input name="prefiere" type="radio" value="trabajar">
```

Cuando se envíe el formulario, si el primer botón está activado la variable `prefiere` será igual a `estudiar`, si es el segundo el activado, la variable `prefiere` valdrá `trabajar`. Observa que lo que ponemos como valor no aparece escrito en la página es un dato interno.

RADIO

Sexo:

<INPUT TYPE="radio" NAME="sexo" VALUE="M" CHECKED>Mujer

<INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre

<?PHP

\$sexo = \$_REQUEST['sexo'];

print (\$sexo);

?>



```
<html>
  <head>
    <title>Problema</title>
  </head>
  <body>
<form action="pagina2.php"
  method="post">
  Ingrese primer valor:
  <input type="text" name="valor1">
  <br>
  Ingrese segundo valor:
  <input type="text" name="valor2">
  <br>
  <input type="radio" name="radio1" value="suma">sumar
  <br>
  <input type="radio" name="radio1" value="resta">restar
  <br>
  <input type="submit" name="operar">
  </form>
</body>
</html>
```

Ingrese primer valor: 10

Ingrese segundo valor: 4

☒ sumar

☐ restar

Enviar consulta

```
<html>
<head>
<title>Problema</title>
</head>
<body>
```

La suma es:14

```
<?php
    if ($_REQUEST['radiol']=='suma")
    {
        $suma=$_REQUEST['valor1'] + $_REQUEST['valor2'];
        echo "La suma es:".$suma;
    }
    else
    {
        if ($_REQUEST['radiol']=="resta")
        {
            $resta=$_REQUEST['valor1'] - $_REQUEST['valor2'];
            echo "La resta es:".$resta;
        }
    }
?>
</body>
</html>
```

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Formulario y tratamiento para Botones de Radio</title>
<meta charset="utf-8">
</head>
<body>
<?php
if(!isset($_POST["enviar"])){//Si todavía no he pulsado enviar visualizo el formulario
?>
<form action="Radio1.php" method="POST">
Sexo:
<label>
<input type="radio" name="sexo" id="radio1" value="Hombre" CHECKED>
Hombre
</label>
<input type="radio" name="sexo" id="radio2" value="Mujer">
<label for="radio2">Mujer</label>
<input type="submit" name="enviar" value="Enviar">
</form>
<?php
}
else{//Si ya he pulsado Enviar hago el tratamiento
echo "La persona es un/a ".$_POST['sexo'];
}
?>
</body>
</html>

```

Sexo: ☒ Hombre ☐ Mujer

← → ↻ 📄 localhost:8080

La persona es un/a Hombre

CASILLA DE VERIFICACIÓN: CHECKBOX

Para insertar una casilla de verificación, el atributo type debe tener el valor checkbox.

Atributos:

- **value** indica el valor asociado a la casilla de verificación. Es necesario poner este atributo, aunque el usuario no pueda ver su valor. Es el valor a enviar.
- **checked** indica que la casilla aparecerá activada inicialmente. Este atributo no toma valores.

- Por ejemplo, para insertar la casilla:

```
<input name="casilla" type="checkbox"  
value="acepto" checked>
```

CHECKBOX

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín

<?PHP

```
$extras = $_REQUEST['extras'];
```

```
foreach ($extras as $extra)
```

```
    print ("{$extra}<BR>\n");
```

?>



```
<head>
  <title>Problema</title>
</head>
<body>
<form action="pagina3.php"
  method="post">
  Ingrese primer valor:
  <input type="text" name="valor1">
  <br>
  Ingrese segundo valor:
  <input type="text" name="valor2">
  <br>
  <input type="checkbox" name="check1">sumar
  <br>
  <input type="checkbox" name="check2">restar
  <br>
  <input type="submit" name="operar">
  </form>
</body>
</html>
```

Ingrese primer valor: 13

Ingrese segundo valor: 2

☐ sumar

☒ restar

Enviar consulta

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Formulario y tratamiento para Casillas de verificación</title>
<meta charset="utf-8">
</head>
<body>
<?php
if(!isset($_POST["enviar"])){//Si todavía no he pulsado Enviar visualizo el formulario
?>
<form action="CheckBox1.php" method="POST">
<label><input type="checkbox" name="extras[]" value="garaje">Garaje</label>
<label><input type="checkbox" name="extras[]" value="piscina">Piscina</label>
<label><input type="checkbox" name="extras[]" value="jardin">Jardín</label>
<input type="submit" name="enviar" value="Enviar">
</form>
<?php
}
else{//Si ya he pulsado Enviar hago el tratamiento
$extras=$_POST['extras'];
echo "Los extras elegidos son: <br />";
foreach($extras as $extra){
echo $extra."<br />";
}
}
?>
</body>
</html>

```

☒ Garaje
 ☐ Piscina
 ☒ Jardín

Los extras elegidos son:
 garaje
 jardin

Para saber si existe una determinada componente en un vector se emplea la función `isset`, si retorna `true` significa que existe y por lo tanto el checkbox está seleccionado.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
    if (isset($_REQUEST['check1']))
    {
        $suma=$_REQUEST['valor1'] + $_REQUEST['valor2'];
        echo "La suma es:".$suma."<br>";
    }
    if (isset($_REQUEST['check2']))
    {
        $resta=$_REQUEST['valor1'] - $_REQUEST['valor2'];
        echo "La resta es:".$resta;
    }
?>
</body>
</html>
```

La resta es:11

Ejemplo

```
<!DOCTYPE>
<html>
<body>
  Selecciona la fruta que te gusta:
  <form action="procesarchek.php" method="post">
    <form action="checkbox.php" method="post">
      <input type="checkbox" name="fruta[]" value="pera"> pera<br>
      <input type="checkbox" name="fruta[]" value="naranja"> naranja<br>
      <input type="checkbox" name="fruta[]" value="platano"> platano<br>
      <br><input type="submit" value="selecciona">
    </form>
  </form>
</body>
</html>
```

Aplicaciones - HTML y CSS: Ejemplo

Selecciona la fruta que te gusta:

☒ pera

☐ naranja

☒ platano

```
<?php
echo "He seleccionado : <br>";
foreach($_POST['fruta'] as $fruta){
    echo $fruta."<br>";
}
?>
```

```
<?php
echo"<br>";
echo "Lo muestro de otra forma <br>";
echo "He seleccionado : <br>";
if(! (empty($_POST['fruta']))) {
    foreach($_POST['fruta'] as $fruta){
        echo $fruta."<br>";
    }
}
?>
```

He seleccionado :
pera
platano

Lo muestro de otra forma
He seleccionado :
pera
platano

BUTTON

```
<INPUT TYPE="button" NAME="actualizar" VALUE="Actualizar  
datos">
```

```
<?PHP
```

```
    $actualizar = $_REQUEST['actualizar'];
```

```
    if ($actualizar)
```

```
        print ("Se han actualizado los datos");
```

```
?>
```

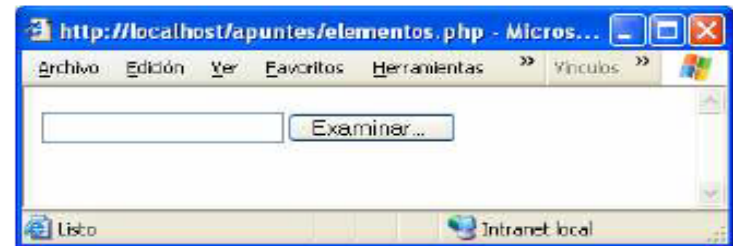


FILE

Para que el usuario envíe un fichero suyo a través del formulario tenemos el tipo file, este tipo hay que utilizarlo con mucho cuidado ya que si permitimos que cualquier internauta nos envíe cualquier tipo de fichero nos podríamos encontrar con sorpresas no siempre agradables.

FILE

```
<FORM ACTION="procesa.php" METHOD="post"  
  ENCTYPE="multipart/form-data">  
  <INPUT TYPE="file" NAME="fichero">  
</FORM>
```



Vamos a ver como se envían ficheros desde un formulario y como se reciben

file permite el envío de archivos

Para recibir datos cuenta con dos matrices específicas `$HTTP_POST_FILES` y `$_FILES` que cumplen la misma función.

Aunque es recomendable usar **`$_FILES`** .

PHP no reconoce ni puede procesar ficheros que hayan sido enviados mediante GET . **SIEMPRE usar el POST**

Las matrices que gestionan los ficheros son asociativas de dos índices:

Filas-> contiene el nombre que se ha dado en el formulario a cada campo del fichero

Dentro de cada fila encontramos otro con las propiedades del fichero, tales como nombre, tamaño, tipo, etc

Las propiedades que definen un archivo enviado a un servidor

PROPIEDAD	CONTENIDO
error	Indica si se ha producido un error en el envio del fichero . Si todo ha ido correctamente esta propiedad almacena el valor 0, si se ha producido un error el valor es1.
name	El nombre del archivo, tal como lo tiene almacenado el usuario en el ordenador
Tmp_name	Es un nombre temporal que usa php para la gestión del archivo, hasta que lo almacena en el disco del servidor
type	Almacena el tipo de fichero que se ha enviado. Puede ser una imagen, audio ,video, etc
size	Elpesodelarchivo en bytes.

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<form action="enviarFichero.php" method="post" name="f_prof" id="f_prof" enctype="multipart/form-data">
```

```
  Fichero:
```

```
  <input type="file" name="fichero" id="fichero">
```

```
  <input type="submit" value="ENVIAR" name="ok" id="ok">
```

```
</form>
```

```
</body>
```

```
</html>
```

Fichero: F:\IAW 2010-2011\mater

Examinar...

ENVIAR

```
<?php
```

```
foreach ($_FILES["fichero"] as $clave => $valor){  
    echo ("Propiedad: $clave ---- Valor: $valor<br>");  
}
```

```
?>
```

Propiedad: name ---- Valor: imagen.jpg

Propiedad: type ---- Valor: image/jpeg

Propiedad: tmp_name ---- Valor: C:\wamp\tmp\php1DD.tmp

Propiedad: error ---- Valor: 0

Propiedad: size ---- Valor: 19871

Limitar tamaño archivo

Si queremos limitar el tamaño de los archivos que subimos al servidor. Utilizaremos el campo `MAX_FILE_SIZE`, valor a establecer como limite y que estará en bytes

```
<html>

<head>
</head>
<body>
  <form action="enviarFichero.php" method="post" name="f_prof" id="f_prof" enctype="multipart/form-data">
    Fichero:
    <input type="hidden" name="MAX_FILE_SIZE" value="10240">
    <input type="file" name="fichero" id="fichero">
    <input type="submit" value="ENVIAR" name="ok" id="ok">
  </form>
</body>
</html>
```

Propiedad: name ---- Valor: video.MOV
Propiedad: type ---- Valor:
Propiedad: tmp_name ---- Valor:
Propiedad: error ---- Valor: 2
Propiedad: size ---- Valor: 0

El campo oculto tiene que estar siempre antes que el campo de archivo

Si el usuario intenta mandar un archivo que supere el tamaño máximo permitido , dicho archivo no será aceptado por PHP.

Las propiedades type y tmp_name no tendrán contenido ,la propiedad size será 0 y error será 2.

Este sistema no es el mejor, ya que cualquier persona con unos mínimos conocimientos de HTML Injection se lo saltara.

Siempre que se pueda llevar a cabo las comprobaciones en el servidor.

En este caso podemos hacerlo mediante la comprobación en el servidor de la propiedad size

```
<?php
```

```
foreach ($_FILES["fichero"] as $clave => $valor){  
    echo ("Propiedad: $clave ---- Valor: $valor<br>");  
}
```

```
?>
```

```
Propiedad: name ---- Valor: video.MOV  
Propiedad: type ---- Valor:  
Propiedad: tmp_name ---- Valor:  
Propiedad: error ---- Valor: 2  
Propiedad: size ---- Valor: 0
```

HIDDEN

El tipo hidden que sirve para definir un campo oculto, campo que no ve el usuario, en este caso habría que incluir el atributo **value** para que el formulario pase ese valor al programa que recoge los datos del formulario.

hidden

```
<?PHP
```

```
    print("<INPUT TYPE='hidden' NAME='username' VALUE='$usuario'>\n");
```

```
?>
```

```
<?PHP
```

```
    $username = $_REQUEST['username'];
```

```
    print ($username);
```

```
?>
```



CAMPO DE CONTRASEÑA: PASSWORD

Para insertar un campo de contraseña, el atributo type debe tener el valor password.

El resto de atributos son los mismos que para un campo de texto normal. La única diferencia es que todas las letras escritas en el campo de contraseña serán visualizadas como asteriscos.

Por ejemplo, para insertar un campo de contraseña:

```
<input name="contra" type="password"  
value="contraseña" size="20" maxlength="15">
```

password

Contraseña: <INPUT TYPE="password" NAME="clave">

```
<?PHP
```

```
    $clave = $_REQUEST['clave'];
```

```
    print ($clave);
```

```
?>
```



BOTÓN

Para insertar un botón, el atributo type debe tener el valor **submit**, **reset** o **button**.

- Si el valor es **submit**, al pulsar sobre el botón se enviará el formulario.
- Si el valor es **reset**, al pulsar sobre el botón se restablecerá el formulario, borrándose todos los campos del formulario que hayan sido modificados y adquiriendo su valor inicial.
- Si el valor es **button**, al pulsar sobre el botón no se realizará ninguna acción por defecto.
- El atributo **value** indica el texto que mostrará el botón.

submit

```
<INPUT TYPE="submit" NAME="enviar" VALUE="Enviar datos">
```

```
<?PHP
```

```
    $enviar = $_REQUEST['enviar'];
```

```
    if ($enviar)
```

```
        print ("Se ha pulsado el botón de enviar");
```

```
?>
```



CAMPOS DE SELECCIÓN <SELECT>

Los campos de selección se utilizan para insertar menús y listas desplegables.

Para insertar uno de estos menús o listas es necesario insertar las etiquetas <select> y </select> en un formulario.

Atributos:

- **name** indica el nombre del menú o lista será el nombre de la variable que contendrá el valor seleccionado.
- **size** indica el número de elementos de la lista que pueden ser visualizados al mismo tiempo, determina el alto de la lista.

- **multiple** indica que el usuario podrá seleccionar varios elementos de la lista mismo tiempo, ayudándose de la tecla Ctrl. Este atributo no toma valores.
- **disabled** indica que la lista estará desactivada, por lo que el usuario no podrá seleccionar sus elementos. Este atributo tampoco toma valores.
- Cada uno de los elementos de la lista ha de insertarse entre las etiquetas `<option>` y `</option>`.
- **value** indica el valor a enviar si se selecciona el elemento. Si no se especifica este atributo, se enviará el texto de la opción, que se encuentra entre las etiquetas `<option>` y `</option>`.
- **selected** indica que el elemento aparecerá seleccionado. Este atributo no toma valores.

Select simple

Color:

```
<SELECT NAME="color">  
  <OPTION VALUE="rojo" SELECTED>Rojo  
  <OPTION VALUE="verde">Verde  
  <OPTION VALUE="azul">Azul  
</SELECT>
```

```
<?PHP  
  $color = $_REQUEST['color'];  
  print ($color);  
?>
```



```
<h3>PHP HTML Form select box Ejemplo</h3>
```

```
<form name="UserInformationForm" method="POST" action="select1.php">
    Introducizca su nombre :
    <input name="nombre" type="text" value=""><br/><br/>
    My Country :
    <select name="pais">
        <option value=""></option>
        <option value="Argentina" >Argentina</option>
        <option value="Bangladesh"> Bangladesh</option>
        <option value="Alemania" >Alemania</option>
    </select>
    <br/><br/>
    <input name="Submit" type="submit" value="Seleccionar">
</form>
```

PHP HTML Form select box Ejemplo

Introducizca su nombre :

My Country :

Seleccionar

```
<?php
    echo "Nombre : {$_POST['nombre']}</br>";
    echo "Pais : {$_POST['pais']}</br>";

    echo "<hr>";
?>
```

Nombre : pedro Picapiedra

Pais : Alemania

```
] <html>
[   <head>
-   <title>Problema</title>
[   </head>
-   <body>
[   <form action="pagina2.php"
-   method="post">
[   Ingrese primer valor:
-   <input type="text" name="valor1">
[   <br>
-   Ingrese segundo valor:
[   <input type="text" name="valor2">
-   <br>
[   <select name="operacion">
-   <option value="suma">sumar</option>
[   <option value="resta">restar</option>
-   </select>
[   <br>
-   <input type="submit" name="operar">
[   </form>
-   </body>
[   </html>
```

Ingrese primer valor:

Ingrese segundo valor:

▼

```
<html>
<head>
<title>Problema</title>
</head>
<body>
  <?php
    if ($_REQUEST['operacion']==suma)
    {
      $suma=$_REQUEST['valor1'] + $_REQUEST['valor2'];
      echo "La suma es:".$suma;
    }
    else
    {
      if ($_REQUEST['operacion']==resta)
      {
        $resta=$_REQUEST['valor1'] - $_REQUEST['valor2'];
        echo "La resta es:".$resta;
      }
    }
  ?>
</body>
</html>
```

La suma es:16

Select múltiple

Idiomas:

```
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
  <OPTION VALUE="ingles" SELECTED>Inglés
  <OPTION VALUE="frances">Francés
  <OPTION VALUE="aleman">Alemán
  <OPTION VALUE="holandes">Holandés
</SELECT>
```

```
<?PHP
```

```
  $idiomas = $_REQUEST['idiomas'];
  foreach ($idiomas as $idioma)
    print (" $idioma<BR>\n");
```

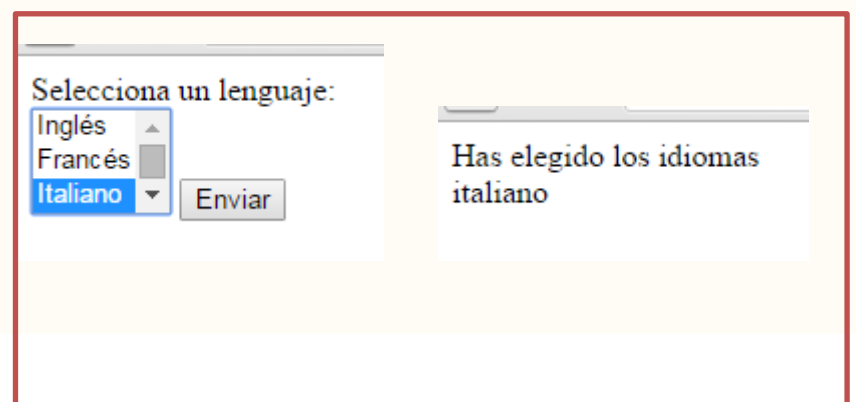
```
?>
```



```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Formulario y tratamiento de un SELECT</title>
<meta charset="utf-8">
</head>
<body>
<?php
if(!isset($_POST["enviar"])){//Si todavía no he pulsado enviar visualizo ?>
Selecciona un lenguaje:
<form action="SelectIdiomas.php" method="POST">
<select multiple size="3" name="idiomas[]">
<option value="ingles">Inglés</option>
<option value="frances">Francés</option>
<option value="italiano">Italiano</option>
<option value="ruso">Ruso</option>
<option value="chino">Chino</option>
</select>
<input type="submit" name="enviar" value="Enviar">
</form>
<?php
}
else{
$idiomas=$_POST["idiomas"];
echo "Has elegido los idiomas<br />";
foreach($idiomas as $idioma){
echo $idioma."<br />";
}
}
?>
</body>
</html>

```



The screenshot shows the web application's output. On the left, a form titled "Selecciona un lenguaje:" contains a multiple-select dropdown menu with three visible options: "Inglés", "Francés", and "Italiano". The "Italiano" option is currently selected. To the right of the dropdown is a button labeled "Enviar". On the right side of the screenshot, there is a text area displaying the output of the PHP script: "Has elegido los idiomas italiano".

Si en un formulario web tienes que enviar alguna variable en la que sea posible almacenar más de un valor, como es el caso de las casillas de verificación o del select multiple en el ejemplo anterior (se pueden marcar varias a la vez), tendrás que ponerle corchetes al nombre de la variable para indicar que se trata de un array.

ÁREAS DE TEXTO <TEXTAREA>

Las áreas de texto permiten a los usuarios insertar varias líneas de texto. Por ello, suelen utilizarse para que se incluyan comentarios.

Para insertar un área de texto es necesario incluir las etiquetas `<textarea>` y `</textarea>` entre las etiquetas `<form>` y `</form>` del formulario.

Para que el área de texto tenga algún valor inicial, habrá que insertarlo entre las etiquetas `<textarea>` y `</textarea>`.

- **name** indica el nombre del área de texto. Es necesario dar nombres diferentes a cada uno de los elementos de un formulario, para poder identificarlos a la hora de procesarlos.
- **rows** indica el número de líneas que podrán ser visualizadas en el área de texto por lo que determina al alto del área de texto.

- **cols** indica el número de caracteres por línea que podrán ser visualizados en el área de texto por lo que determina al ancho del área de texto.

Gracias a las barras de desplazamiento, es posible insertar más líneas de las especificadas en el atributo rows, y más caracteres por línea de los especificados en el atributo cols.

Por ejemplo:

```
<textarea name="ejemploarea" cols="30"  
rows="3">Escribe el texto que quieras</textarea>
```

textarea

Comentario:

```
<TEXTAREA COLS="50" ROWS="4" NAME="comentario">
```

Este libro me parece ...

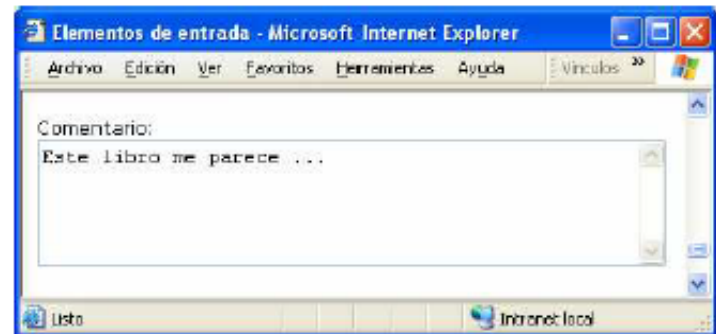
```
</TEXTAREA>
```

```
<?PHP
```

```
    $comentario = $_REQUEST['comentario'];
```

```
    print ($comentario);
```

```
?>
```



```
<html>
  <head>
    <title>Problema</title>
  </head>
  <body>
    <form action="pagina4.php" method="post">
      Ingrese nombre:<input type="text" name="nombre"><br>
      Ingrese su curriculum:<br>
      <textarea name="curriculum"></textarea>
      <br>
      <input type="submit" value="Confirmar">
    </form>
  </body>
</html>
```

Ingrese nombre:

Ingrese su curriculum:

ciclo de

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
    echo "El nombre ingresado:".$_REQUEST['nombre'];
    echo "<br>";
    echo "El curriculum:".$_REQUEST['curriculum'];
    ?>
</body>
</html>
```

El nombre ingresado:pepe
El curriculum:Estudiante de ciclo de gradosuperior

EJEMPLOS

- <http://www.mclibre.org/consultar/php/ejercicios/controles formularios/controles formularios.html>

DIRECTIVA REGISTER_GLOBALS

Si la directiva *register_globals* –en el fichero **php.ini**– está configurada con la opción **ON**, los valores transferidos desde el navegador –además de ser recogidos en las variables como – ***\$_GET['n1']*** a ***\$_POST['n1']***.son asignados a otras variables del tipo: ***\$n1***, ***\$n2***, ... cuyos nombres son el resultado de anteponer el símbolo **\$** a los nombres de las variables contenidas en la petición.

No aconsejable

Ejemplos

- http://www.mclibre.org/consultar/php/lecciones/php_controles.html

PROCESAMIENTO DE LA INFORMACIÓN
DEVUELTA POR UN FORMULARIO WEB.

EJERCICIO-PHP4 Formulario color

Crea dos documentos llamados respectivamente *color1.html* y *color2.php*.

En el primero de ellos incluye un formulario en el que -mediante un menú de opciones– se pueda elegir uno de entre cinco colores (especifica como *value* el código hexadecimal de cada uno de ellos).

Al enviar los datos, deberá cargarse la página *color2.php* con el color de fondo correspondiente a la opción seleccionada en el formulario

EJERCICIO-PHP-5

Crea dos documentos llamados respectivamente **formulario1.php** y **visor1.php**.

En el primero de ellos incluye un formulario que permita recoger datos personales y académicos de alumnos, utilizando todos los tipos de campos de formulario que conozcas.

Los datos insertados en ese formulario deberán ser visualizados después de su envío a través del documento visor1.php.

Utiliza los recursos estéticos –fondos, colores, tipografía, etcétera– que estimes oportunos para una correcta presentación

EJERCICIO-PHP-6

Con criterios similares en cuanto a estética y funcionalidad a los del ejercicio anterior, crea dos nuevos documentos con nombres **formulario2.php** y **visor2.php**. En este caso el formulario deberá poder recoger el nombre y apellidos de un alumno hipotético al que debemos formularle dos preguntas. La primera de ellas con cuatro posibles respuestas entre las que deba elegir como válida una de ellas. La segunda, también con cuatro respuestas, deberá permitir marcar las respuestas correctas que pueden ser: todas, ninguna, o algunas de ellas. El alumno debería poder insertar sus datos personales en el formulario y elegir las respuestas a las preguntas formuladas. Al pulsar en el *botón* de envío, los datos del alumno y las respuestas elegidas deben visualizarse a través del documento **visor2.php**.