CONTROLES EN FORMULARIOS

Vamos a ver los distintos aspectos de los formularios con respecto al procesamiento de los datos.

Para que un control envíe información es necesario:

* que el control esté incluido en un formulario (<form>).
* que el formulario tenga establecido el atributo action, con la dirección absoluta o relativa del fichero php que procesará la información
* que el control tenga establecido el atributo name  
  Nota: el atributo name puede contener cualquier carácter (números, acentos, guiones, etc), pero si contiene espacios, PHP sustituye los espacios por guiones bajos (\_) al procesar los datos enviados.
* que el formulario contenga un botón de tipo submit

El siguiente ejemplo muestra un formulario válido (ejemplo.html):

<form action="ejemplo.php">

<p>Nombre: <input type="text" name="nombre" /></p>

<p><input type="submit" value="Enviar" /></p>

</form>

**Envió y Recepción de datos**

* El lenguaje PHP nos proporciona una manera sencilla para enviar y recibir datos a través de un formulario, para procesar la información que el usuario ha ingresado.
* El proceso para el manejo de FORMULARIOS requiere generalmente dos páginas, una que **implementa** el formulario y otra que **procesa** los datos cargados en el formulario.
* Los **Formularios** no forman parte de PHP, sino del lenguaje estándar de Internet, HTML

En PHP podemos compartir información entre las distintas páginas del sitio web de las siguientes maneras:

* Utilizando los métodos **GET** y **POST**
* Utilizando **Cookies**
* Creando **sesiones de trabajo** *(esta forma de comunicación la veremos de forma más detallada en el próximo módulo del curso)*

**Acceso a formularios desde PHP**

* **$\_GET**, **$\_POST** y **$\_REQUEST**

Para empezar, hablaremos de $\_GET Y $\_POST.

Las dos formas se utilizan como método de paso de datos en formularios

Tratare de explicar lo mejor posible sus diferencias y la correcta aplicación de ambas.

**$\_GET**

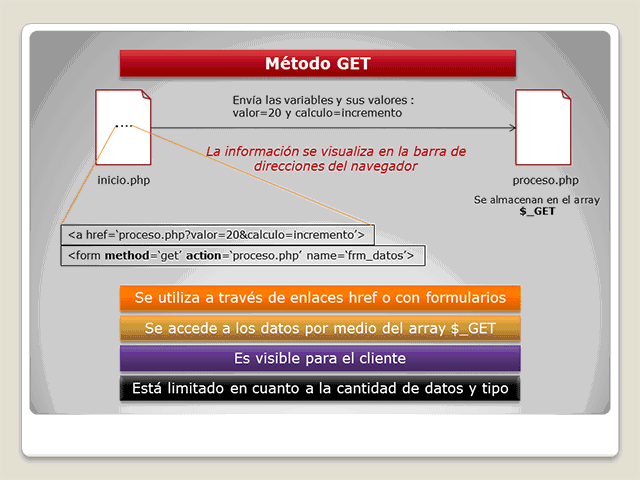
Si utilizamos el método GET por medio de formularios HTML no nos tendremos que preocupar de la sintaxis ya que el propio navegador se encargará de construir la URL.

Si se utiliza con formularios la sintaxis sería la siguiente:

<form method='get' action='pagina.php'>

Tendremos que indicar en el atributo ‘**method’** del formulario la palabra get y en la acción ‘**action’** el nombre de la página php a la que le deseamos enviar los datos.

La página que recibe la información, tendrá acceso a los mismos consultando el **array global** **$\_GET** donde se han almacenado. Este array está formado por claves (nombre de la variable que se envía) y valores (valor de la variable).



Veamos unos ejemplos:

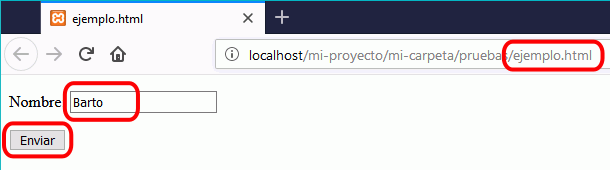
<form action="ejemplo.php" method="get">

        <p>Nombre: <input type="text" name="nombre" /></p>

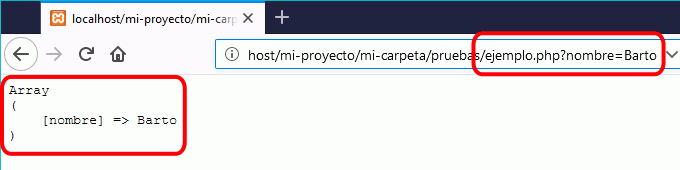
        <p><input type="submit" value="Enviar" /></p>

    </form>

En este pequeño formulario el usuario tendría que completar con su nombre.

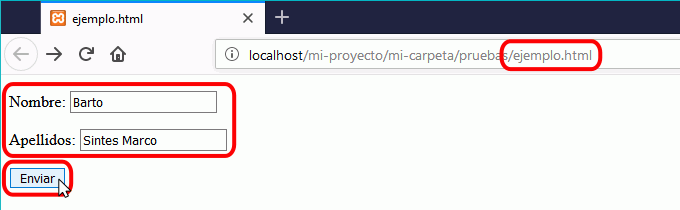


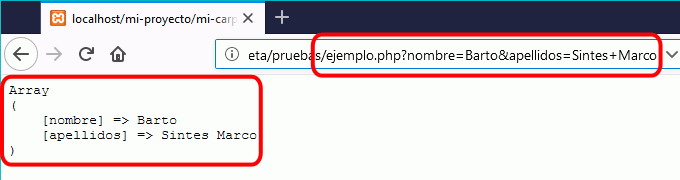
Después de pasar los datos ( Pulsar enviar) la url quedaría así:



Por lo tanto si nosotros escribiéramos la url directamente cambiando "Barto" por otro nombre también funciona.

En caso de que haya varios controles que envíen información en un formulario con get, los nombres y valores aparecen en la barra de dirección separados por el carácter ampersand (&), como nombre1=valor1&nombre2=valor2&...





**$\_POST**

El método POST nos permite también enviar información entre páginas como el método GET, pero sin sus limitaciones.

* La cantidad de información que podemos enviar mediante POST es bastante más grande (varios Mb) y vendrá limitada sólo por la propia configuración del PHP.
* El envío de datos no es visible, ni se almacena en la cache del navegador.
* Nos permite el envío de ficheros (*típico subir archivo*).

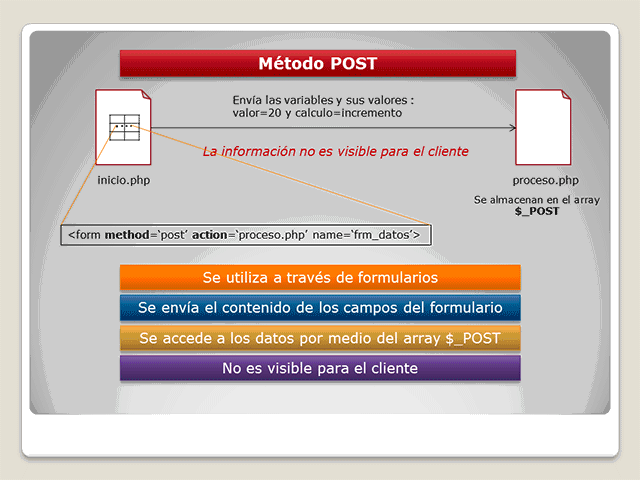
Se utiliza con formularios  y la sintaxis es la siguiente:

<form method='post' action='pagina.php'>

* Tendremos que indicar en el atributo ‘method’ del formulario la palabra post y en la acción ‘action’ el nombre de la página php a la que le deseamos enviar los datos.
* Si se van a enviar archivos por medio de este método, tendremos que establecer el atributo ***enctype*** del formulario con el valor ***multipart/form-data***

<form method='post' action='pagina.php'  enctype='multipart/form-data'></form>

La página que recibe la información, tendrá acceso a los mismos consultando el array global ***$\_POST*** donde se han almacenado. Este array está formado por claves *(nombre del campo del formulario que se envía)* y valores *(valor del campo del formulario)*.



Veamos un ejemplo:

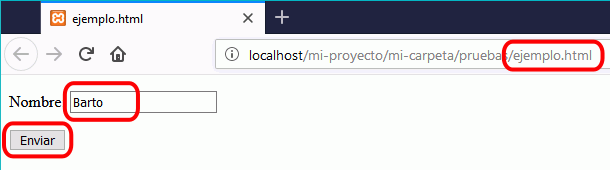
<form action="ejemplo.php" method="post">

        <p>Nombre: <input type="text" name="nombre" /></p>

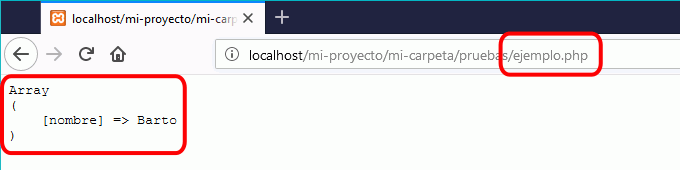
        <p><input type="submit" value="Enviar" /></p>

</form>

En este pequeño formulario el usuario también tendría que completar con su nombre,



Por lo tanto nosotros no podriamos modificar la url para pasar otro tipo de datos,pero a diferencia del anterior, los datos no quedan en la url.



**Diferencias GET y POST:**

**GET**

Tiene un **límite** de caracteres al pasar datos.

Los datos **son visibles**.

**POST**

Se puede pasar **grandes cantidades** de datos.

Los datos **no son visibles**.

**Recuperar Datos:**

Tanto GET como POST tienen sus variables para recuperar los datos pasados por el formulario.

Para recuperar los datos pasados por el método GET usamos la variable **$\_GET**.

Para recuperar los datos pasados por el método POST usamos la variable **$\_POST**.

Usando el ejemplo anterior seria así:

$\_GET['nombre'];

$\_POST['nombre'];

Esas variables nos estarían devolviendo el valor pasado en el formulario, en este caso el nombre

**REQUEST**

* La variable **$\_REQUEST** es un **array asociativo** el cual contiene todos los datos pasados por

**GET, POST y COOKIE**

* ¿Y para que usar $\_REQUEST si podemos usar las variables $\_GET $\_POST Y $\_COOKIE?

Muy simple, esta variable es muy útil en formularios multi-metodos (Por así decirlo).

Vamos a dar un simple ejemplo:

Como verán en este pequeño formulario a pesar de solo tener que ingresar un dato, se están pasando 3 datos en verdad.

setcookie("mes","octubre");

<form action="ejemplo.php?seccion=nombre" method="POST">

<p>Nombre: <input type="text" name="nombre" /></p>

<p><input type="submit" value="Enviar" /></p>

</form>

* + Un dato por GET (En la url del formulario) que sería: seccion
  + Un dato por POST (El que completaría el usuario) que seria: nombre
  + Un dato por COOKIE (El que está definido en setcookie) que seria: mes

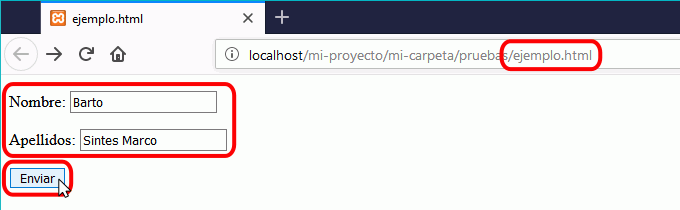
Entonces para poder obtener todos los datos sin tener que usar las tres variables correspondientes utilizaríamos solamente $\_REQUEST.

$\_REQUEST['seccion'];

$\_REQUEST['nombre'];

$\_REQUEST['mes'];

El siguiente ejemplo muestra un ejemplo de formulario:



print\_r($\_REQUEST);

print "</pre>\n";

print "<p>Tu nombre es". $\_REQUEST['nombre']."</p>\n";

print "<p>Tus apellidos son". $\_REQUEST['apellidos']."</p>\n";

Otro Ejemplo

Fichero **uno.php**

<html>

<body>

<form action="dos.php" method="POST">

Edad: <input type="text" name="edad">

<input type="submit" value="aceptar">

</form>

</body>

</html>

Fichero **dos.php**

<html>

<head>

</head>

<body>

<?php

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

print ("La edad es: $edad");

?>

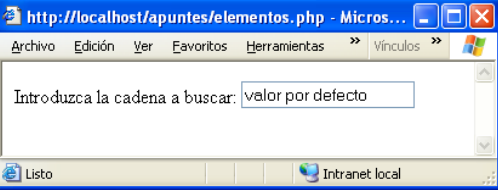
</body>

</html>

**Acceso a los diferentes tipos de elementos de entradas de formulario**

* Elementos de tipo **INPUT**
  + TEXT
  + RADIO
  + CHECKBOX
  + BUTTON
  + FILE
  + HIDDEN
  + PASSWORD
  + SUBMIT
* Elemento SELECT
  + Simple / múltiple
* Elemento TEXTAREA

**TEXT**



Introduzca la cadena a buscar:

<input TYPE="text" name="cadena" value="valor por defecto" size="20">

<?php

$cadena = $\_REQUEST[‘cadena’];

print ($cadena);

?>

**SELECT simple**

Color:

<select name="color">

<option value="rojo" selected>Rojo

<option value="verde">Verde

<option value="azul">Azul

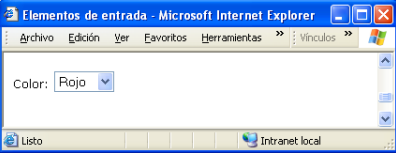
</select>

<?php

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

print ($color);

?>



**SELECT múltiple**

<form action="trataformulario.php" method="POST">

Nombre: <input type="text" name="nombre"><br>

Apellidos: <input type="text" name="apellidos"><br>

Email: <input type="text" name="email"> <br>

Cerveza: <br>

<select multiple name="cerveza[]">

<option value="SanMiguel">San Miguel</option>

<option value="Mahou">Mahou</option>

<option value="Heineken">Heineken</option>

<option value="Carlsberg">Carlsberg</option>

<option value="Aguila">Aguila</option>

</select><br>

<input type="submit" value="Enviar datos!" >

</form>

**trataformulario.php**

<?php

echo "Nombre: ". $\_REQUEST["nombre"];

echo "<br>Apellidos: ". $\_REQUEST["apellidos"];

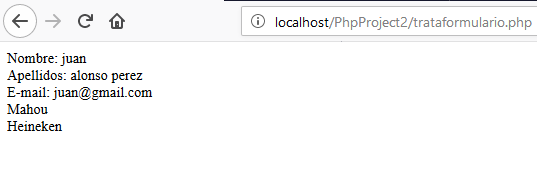
echo "<br>E-mail: ". $\_REQUEST ["email"];

$cervezas=$\_REQUEST["cerveza"];

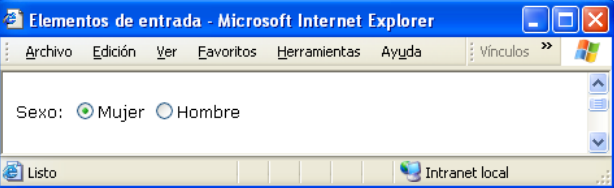
echo'<br>';

foreach ($cervezas as $cerveza)

print ("$cerveza<BR>\n");



**RADIO**



Sexo:

<input type="radio" name="sexo" value="M" checked>Mujer

<input type="radio" name="sexo" value="H">Hombre

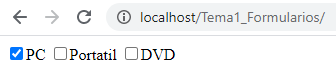
<?php

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

print ($sexo);

?>

**Checkbox**



<input  type="checkbox"  name="pc"  value="pc"  checked="checked">PC

<input  type="checkbox"  name="portatil"  value="portatil">Portatil

<input  type="checkbox"  name="DVD"  value="DVD">DVD

Se tiene que **evaluar** cada uno de los **checkbox** para ver si se ha seleccionado o no.

<?php

    echo "El material informático que tengo es:","<br>";

    if (isset($\_REQUEST  ['pc']))

        echo $\_REQUEST  ['pc'],"<br >";

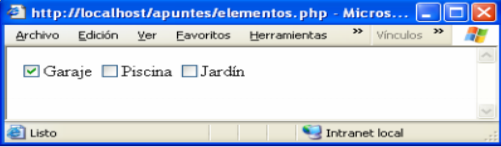
    if (isset($\_REQUEST  ['portatil']))

        echo $\_REQUEST['portatil'],"<br>";

    if (isset($\_REQUEST  ['DVD']))

        echo $\_REQUEST['DVD'],"<br>";

Se podría evaluar los **checkbox** si en vez de darles un nombre diferente a cada uno de ellos como en el ejemplo anterior, se le diera a todos el ***mismo nombre***, que sería el nombre de un array. En este caso se podría recorrer el array utilizando un foreach



<input type="checkbox" name ="extras[]" value ="garaje" checked="checked">Garaje

<input type ="checkbox" name ="extras[]" value ="piscina">Piscina

<input type ="checkbox" name ="extras[]" value ="jardin">Jardín

<?php

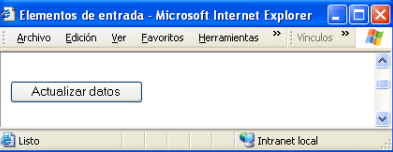
$extras = $\_REQUEST['extras'];

foreach ($extras as $extra)

print (“$extra<BR>\n”);

?>

**BUTTON**



<input type="button" name="actualizar" value="Actualizar datos">

<?php

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

if ($actualizar)

print ("Se han actualizado los datos");

?>

Otro ejemplo es cuando el formulario contiene varios botones y queremos saber cuál se ha pulsado, como en el ejemplo siguiente, en el que los atributos **name** son iguales, pero los atributos **value** son distintos:

<form action="ejemplo.php" method="get">

<p>¿Sí o no?</p>

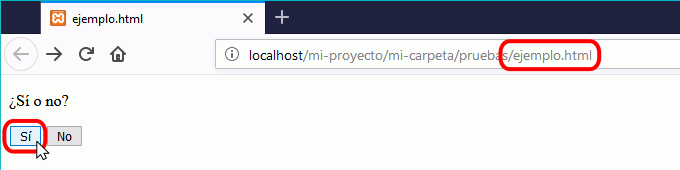
<p>

<input type="submit" name="respuesta" value="Si" />

<input type="submit" name="respuesta" value="No" />

</p>

</form>



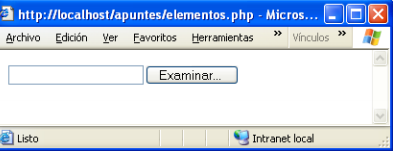
**FILE**

El selector de archivo permite enviar un archivo desde el ordenador del cliente al servidor.

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

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

</form>



**HIDDEN**

<?php

print("<input type='hidden' name='username' value='juan'>");

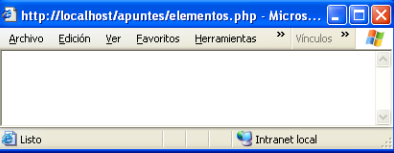
?>

<?php

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

print ($username);

?>



**PASSWORD**

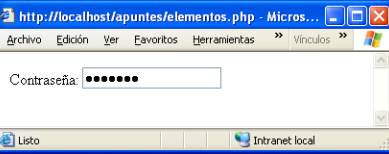
Contraseña: <input type="password" name="clave">

<?php

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

print ($clave);

?>



**SUBMIT**

<input type="submit" name="enviar" value="Enviar datos">

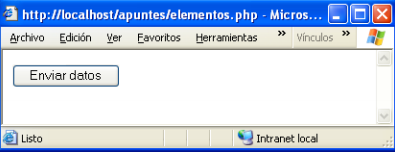
<?php

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

if ($enviar)

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

?>



**TEXTAREA**

Comentario:

<textarea cols="50" rows="4" name="comentario">

Este libro me parece ...

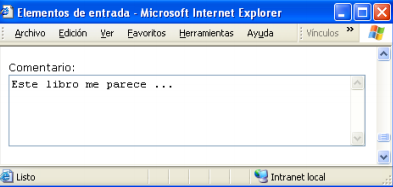
</textarea>

<?php

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

print ($comentario);

?>



Envió del formulario de PHP

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")

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

Ejemplo:

Usando la función isset y el bloque if, realizar una comprobación sobre una caja de texto cualquiera, advirtiendo al usuario en caso de que la caja se encuentre sin datos.

<form action="trataformulario.php" method="POST">

Nombre: <input type="text" name="nombre"><br>

<input type="submit" value="Enviar" />

</form>

<?php

if (isset($nombre) ){

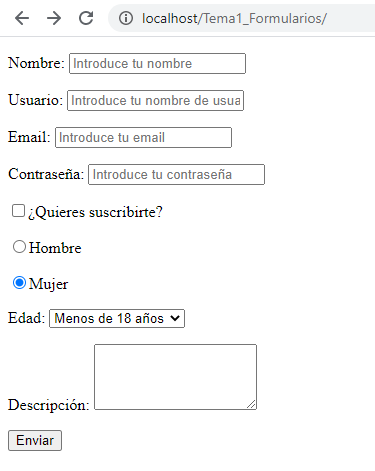
echo "Hola $nombre, bienvenido a PHP";}

else {

echo "No has escrito nada"; }

?>

**Ejemplo completo**



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Ejemplo Formularios</title>

</head>

<body>

        <p>Nombre: <input type="text" name="nombre" placeholder="Introduce tu nombre"></p>

        <p>Usuario: <input type="text" name="usuario" placeholder="Introduce tu nombre de usuario" required></p>

        <p>Email: <input type="email" name="correo" placeholder="Introduce tu email" required></p>

        <p>Contraseña: <input type="password" name="password" placeholder="Introduce tu contraseña" required></p>

        <p><input type="checkbox" name="suscripcion">¿Quieres suscribirte?</p>

        <p><input type="radio" name="genero" value="Hombre">Hombre</p>

        <p><input type="radio" name="genero" value="Mujer" checked>Mujer</p>

        <p>Edad: <select name="edad">

          <option value="menor">Menos de 18 años</option>

          <option value="adulto">Entre 18 y 67 años</option>

          <option value="jubilado">Más de 67 años</option>

        </select></p>

        <p>Descripción: <textarea name="descripcion" rows="4" cols="20"></textarea></p>

        <p><input type="submit" value="Enviar"></p>

      </form>

</body>

</html>

**Recogida de datos**

<?php

$nombre = $\_REQUEST['nombre'];

$usuario = $\_REQUEST['usuario'];

$correo = $\_REQUEST['correo'];

$password = $\_REQUEST['password'];

if (isset($\_REQUEST ['suscripcion']))

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

$genero = $\_REQUEST['genero'];

$edad = $\_REQUEST['edad'];

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