Programación Web con PHP y MySQL– Nivel 2

Unidad 5: Estructuras de control

Contacto: consutlas@elearning-total.com
Web: www.elearning-total.com

Indice

Unidad 5: Estructuras de control

Objetivos

Que el alumno logre:

- Aprender los conceptos de paso de valores entre archivos

Links con variables

Vamos a ver a continuación como podemos pasar datos, a través de variables, entre distintas páginas de nuestro sitio.

La primera de las formas de "definirle un valor" o "setear" una variable: es a través los links. Cuando en HTML hacemos un link, habitualmente se parece a esto:

```
<html>
<head>
<title>Paso de variables por PHP</title>
</head>
<body>
<a href="destino.html">Esto es un link</a>
</body>
</html>
```

El efecto del tag <a> ("anchor", o "ancla", que nos enlaza a otra página) es pedirle al servidor que nos "muestre" o nos entregue la página especificada en el atributo "href" (es la abreviatura de "hiper-referencia"), que en este caso es la página "destino.htm" (por supuesto, si no hemos creado ninguna página llamada "destino.htm", nos dará un error cuando ejecutemos este link).

Pero en el mundo de las páginas dinámicas, nosotros podemos aprovechar la posibilidad de "decirle algo" al servidor junto con el link, para que, además de pedirle ver una página en particular, nos deje ponerle valor a variables que luego estarán disponibles para que las use en esa misma página el intérprete de PHP, cuando le sea "pasada" la página para que la procese antes de devolverla a nuestro navegador.

La forma es la siguiente: supongamos que queremos definir una variable que se llame "nombre" y darle el valor "Pepe". Supongamos también que el link apunta a la página "recibe.php". El link que nos lleve a esa página debería ser así:

```
<a href="recibe.php?nombre=Pepe">Este es el link</a>
```

Notemos algo, luego de especificar la hiper-referencia (la página "recibe.php") se ha agregado un signo de pregunta. Este signo indica que a continuación le vamos a enviar variables al servidor web.

La forma de pasarle o "definirle" una variable es colocando primero el **nombre** de la variable, luego un **signo igual**, y luego **el valor** que queremos que almacene esa variable (recordar que una variable es como una "cajita" donde depositar un dato por unos instantes)

VARIABLE $_GET

Volvamos a tomar un concepto visto en la unidad anterior, las variables supe globales, en este caso la variable ***$_GET.***

La variable $_GET es un vector asociativo, (veremos a fondo vectores en la unidad tres de este mismo módulo) digamos por ahora, muy a grandes rasgos que es un tipo de variable, que guarda toda la información enviada de una página a otra bajo el método "get".

Por ejemplo, si nosotros tenemos un formulario con el valor "get" en la propiedad "method", que contenga los campos con nombre "Usuario" e "Email", al enviar ese formulario podemos ver que nos dirige a la página especificada en en la propiedad "action" con algunas variables en su url.

Ejemplo:
Cuando vemos que tras la url de un documento hay un signo de interrogación, lo que sigue es el nombre de una variable con su valor correspondiente (Variable=valor), y a cada variable a partir de la segunda en vez de un signo de interrogación se le antepondrá un ampersand (&).

La variable $_GET guarda cada una de estas variables, a las que se puede acceder usando el nombre de cada una de ellas como clave del array $_GET.

No sólo se usa la variable $_GET para recuperar la información enviada por un formulario con método "get". Uno mismo puede "filtrar" información en una página enviándole información en la URL.

En el siguiente ejemplo tenemos una página *alumnos.htm* donde hay 2 links.

Ambos links están dirigidos a la misma página, *detallealumnos.php*, con la diferencia de que en el primero se envía la variable nombre=juan y en el segundo nombre=pepe.

Al hacer click en alguno de los dos, nos dirigimos a la página *detallealumnos.php*, donde un script PHP verifica que la variable nombre enviada sea igual a "juan" o "pepe", y depende el caso muestra los datos del alumno correspondiente.

alumnos.htm

```
<a href="detallealumnos.php?nombre=juan">Información de Juan</a>
<a href="detallealumnos.php?nombre=pepe">Información de Pepe</a>
```

detallealumnos.php

```
<?php
if($_GET['nombre']==”juan”) { //Evaluamos si la variable “nombre” enviada tiene el valor “juan”.
?>
<p>Nombre: Juan</p>
<p>Apellido: Pérez</p>
<p>Edad: 24</p>
<?php
} elseif ($_GET['nombre ']==”pepe”) { //Evaluamos si la variable “nombre” enviada tiene el valor
“pepe”.
?>
<p>Nombre: Pepe</p>
<p>Apellido: González</p>
<p>Edad: 30</p>
<?php
}
?>
```

VARIABLE $_POST

$_POST es otro vector asociativo que también guarda información transferida de una página a otra como la variable $_GET, pero en este caso guarda la información transferida bajo el método “post” de un formulario.

Otra diferencia con $_GET es que las variables enviadas bajo “post” no se verán en la url de la página a la que se envían los datos, por eso es más cómodo usar el método “post” en formularios en vez de “get”.

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.

Este es otro método por excelencia para pasar valores entre dos páginas, “el formulario”.

Se trata de un tag del lenguaje HTML, por lo que no es preciso que la página donde está el formulario lleve extensión ".php", puede ser un archivo ".html".

Sus elementos principales son los siguientes:

1) El atributo "***action***", que indica "a qué página" le está pasando las variables, siendo la página que nos va a mostrar cuando pulsemos en el botón "Enviar", tal como si fuese un link hacia esa página.

2) El atributo "***method***", que especifica uno de los dos posibles "métodos" o formas de pasar las variables: a la vista de todos en la URL del navegador (method="GET", el mismo que usamos en la sección anterior para los links que pasan variables) o si las vamos a pasar ocultas (method="POST").
Este último método es el más utilizado en formularios.

3) Algún campo que permita al usuario el ingreso de datos (input). Lo fundamental de cada campo será su nombre (atributo "name"), ya que ése será el nombre de la variable que enviará a la página de destino.

4) Un botón para enviar los datos.

Esos son todos los elementos básicos de un formulario.

Vamos a un ejemplo que llamaremos ***"formu.htm"***

```
<html>
<head>
<title> Paso de variables por PHP </title>
</head>
<body>
        <form action="muestra.php" method="POST">
        <input type="text" name="direccion">
        <input type="submit" value="Enviar">
        </form>
</body>
</html>
```

Y ahora crearemos ***"muestra.php"***, que es la página que recibe la/s variable/s con lo que haya escrito el usuario en el formulario, y la/s muestra:

```
<?php
print ("Su direccion es: ");
print ($direccion);
?>
```

Notemos que ha llegado la variable **$direccion** a esta página mediante el formulario. Cada **<input>** nos genera **una variable**, por lo que si quisiéramos permitir que el usuario escriba varios datos, sólo tenemos que agregar **varios inputs**, de cualquier tipo (texto, textarea multilínea, select desplegable, radio buttons, checkboxes, etc.).

Recomendamos repasar algún manual o tutorial de HTML para refrescar este tema. Ya que a partir de esta unidad nos manejaremos con PHP intercalándolo entre las etiquetas y atributos de HTML

Resumen

En esta Unidad…

En la presente unidad trabajamos con estructuras de control en PHP

En la próxima Unidad…

En la próxima unidad trabajaremos con bucles.