Formularios con PHP

Actividad 1.4. Formularios y estructuras de datos en PHP

Elaborado por: Juan Pablo Ortiz González

Método POST

Además de poder enviar valores en las peticiones al servidor por medio de la url (método GET), existe el método POST.

A diferencia del método GET en el método POST los valores no son visibles para el usuario, esto lo hace muy útil si queremos enviar datos sensibles como lo es una contraseña.

Para enviar información a través del método POST tenemos que hacer uso de formularios.

Formularios

Estos se definen usando HTML:

```
1  <html>
2  <body>
3    <form method="post">
4      <label>Campo 1</label>
5      <input type="text" name="campo_1" />
6      <label>Campo 2</label>
7      <input type="text" name="campo_2" />
8      <button>Enviar</button>
9    </form>
10 </body>
11 </html>
```

Es importante notar que en el formulario (línea 3) se especifica method=”post”, get es usado por default

Superglobal $_POST

Análogamente a la variable superglobal $_GET, PHP define $_POST la cual también es un arreglo asociativo.

```
1  <?php
2  if (isset($_POST["campo_1"]))
3  {
4      echo "Enviaste el valor: " . $_POST["campo_1"];
5  }
6  ?>
```

Ejemplo:

```html
<html>
<body>
  <form method="post">
    <label for="campo_1">Campo 1</label>
    <input type="text" name="campo_1" />
    <br>
    <label for="opcion">Selecciona:</label>
    <select id="opcion" name="opcion">
     <option value="1">Opción 1</option>
     <option value="2">Opción 2</option>
     <option value="3">Opción 3</option>
    </select>
    <br>
    <textarea name="mensaje" rows="10" cols="30">
    Escribe aqui.
    </textarea>
    <br>
    <input type="radio" id="si" name="respuesta" value="si">
    <label for="si">Si</label><br>
    <input type="radio" id="no" name="respuesta" value="no">
    <label for="no">No</label><br>
    <input type="radio" id="otro" name="respuesta" value="otro">
    <label for="otro">Otro</label>
    <br>
    <button>Enviar</button>
  </form>
</body>
</html>
```

Para ver los datos enviados:

```php
<?php
var_dump($_POST);
?>
```

Archivos

Podemos subir archivos al servidor con PHP, para esto debemos definir un formulario de la siguiente forma:

```html
<form method="post" enctype="multipart/form-data">
    <label for="archivo">Selecciona un archivo</label>
    <input type="file" name="archivo">
    <button type="submit" name="upload">Subir</button>
</form>
```

Nótese en el formulario el atributo enctype="multipart/form-data", esto es esencial para que el envío de archivos funcione.

Luego usamos un *input* de tipo ***file*** y un botón llamado upload esto solo para verificar que el formulario ha sido enviado y no tratar de subir archivos la primera vez que se muestra la página.

En PHP tendremos que usar varias funciones:

```php
<?php
/* Guardar un archivo en el servidor desde un formulario HTML */
$message = '';
$error = '';
$uploads_dir = 'uploads';
//si $_POST tiene definido el índice upload quiere decir que el
//formulario fue enviado
if (isset($_POST["upload"]))
{
    //Verificamos si existe algún error
    $error = $_FILES["archivo"]["error"];
    //si no hay errores podemos procede
    if ($error == UPLOAD_ERR_OK) {
        //obtenemos el nombre temporal del archivo
        $tmp_name = $_FILES["archivo"]["tmp_name"];
        //la función basename nos regresa el nombre del archivo sin la ruta
        $name = basename($_FILES["archivo"]["name"]);
        //movemos el archivo subido a la carpeta de nuestra elección
        if (move_uploaded_file($tmp_name, "$uploads_dir/$name"))
        {
            //si todo ocurre con normalidad mostraremos al usuario un mensaje de éxito
            $message = basename( $_FILES["archivo"]["name"]) . " cargado correctamente";
        }
        else $error = "Ocurrió un error al subir el archivo";
    }
    else $error = "Ocurrió un error al subir el archivo";
}
?>
```

Lo primero a notar es el uso de la variable $_FILES que al igual que $_POST y $_GET es una variable superglobal y es un arreglo asociativo. Contiene la informacion de los archivos enviaods a traves del formulario.

La funcion move_uploaded_file guardara el archivo enviado en la ubicación que especifiquemos.

Por último, la función basename simplemente devuelve el nombre del archivo sin especificar la ruta por ejemplo basename("C:/xampp/htdocs/plb/index.php"); devolverá “index.php”

Ver la documentación oficial