Programación Web con PHP y MySQL– Nivel 2

Unidad 7: Funciones en PHP / Session

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

Indice

Unidad 7: Funciones en PHP

Objetivos

Que el alumno logre:

- Comprender la utilización de las funciones principales de PHP

FUNCIONES EN PHP

Funciones y procedimientos

El concepto de función podría ser definido como un conjunto de instrucciones que permiten procesar las variables para obtener un resultado. Puede que esta definición resulte un poco vaga si no nos servimos de un ejemplo para ilustrarla.

Supongamos que queremos calcular el valor total de un pedido a partir de la simple suma de los precios de cada uno de los artículos.

Tendríamos que seguir el siguiente razonamiento:

```
definir función suma(art1,art2,art3)
suma=art1+art2+art3
imprimir(suma)
fin función
```

Este supuesto programa nos permitiría calcular la suma de tres elementos e imprimir el resultado en pantalla. Lo interesante de utilizar este tipo de funciones es que ellas nos permiten su utilización sistemática tantas veces como queramos sin necesidad de escribir las instrucciones tantas veces como veces queremos utilizarla. Por supuesto, podemos prescindir de esta declaración de función e introducir una línea del siguiente tipo:

```
imprimir (art1+art2+art3)
```

Evidentemente, cuanto más complicada sea la función y más a menudo la utilicemos en nuestros scripts más útil resulta definirlas.

Esta función suma podría ser utilizada en cualquier lugar de nuestro script haciendo una llamada del siguiente tipo:

```
ejecuta suma(4,6,9)
```

Cuyo resultado sería:

19

Del mismo modo, los procedimientos son parecidos a las funciones. La diferencia consiste tan solo en que en estos últimos el interés no radica en el resultado obtenido sino más bien en las operaciones realizadas al ejecutarla (creación de un archivo, reenvío a otra página,...).

En lenguajes como el PHP las funciones y los procedimientos son considerados como la misma cosa y para definirlos se hace usando los mismos comandos.

Tanto las variables como las funciones y los procedimientos deben ser nombradas sin servirse de acentos, espacios ni caracteres especiales para no correr riesgos de error.

Existen en PHP una gran variedad de funciones que ya vienen programadas, no obstante nos permite la creación de nuestras propias funciones, veamos ambos casos.

FUNCIONES DEL LENGUAJE

Desarrollaremos a continuación algunas de las funciones más utilizadas.

Función mail()

Para el envío de correos electrónicos utilizando PHP disponemos de una función bastante potente, incluida en todas las versiones de PHP, que nos permite mandar mails sin necesidad de instalar ningún añadido, distinto a lo que nos permitía HTML quien necesita de un programa de correo instalado en la correspondiente máquina.

En concreto, en PHP disponemos de una función llamada ***mail()*** que permite configurar y enviar el mensaje de correo. La función recibe tres parámetros de manera obligatoria y otro parámetro que podemos colocar opcionalmente. Devuelve true si se envió el mensaje correctamente y false en caso contrario.

Parámetros necesarios en todos los casos

Destinatario: la dirección de correo o direcciones de correo que han de recibir el mensaje. Si incluimos varias direcciones debemos separarlas por una coma.

Asunto: para indicar una cadena de caracteres que queremos que sea el asunto del correo electrónico a enviar.

Cuerpo: el cuerpo del mensaje, lo que queremos que tenga escrito el correo.

Ejemplo:

```
<?php
mail("lorena.bernis@gmail.com", "Este es el asunto del mail", "Este es el cuerpo del mensaje");
?>
```

Parámetros opcionales del envío de correo

Headers: Cabeceras del correo. Datos como la dirección de respuesta, las posibles direcciones que recibirán copia del mensaje, las direcciones que recibirán copia oculta, si el correo está en formato HTML, etc.

Ejemplo complejo de envío de correo:

Vamos a enviar un correo con formato HTML a Lorena@gmail.com, con copia a lorena@hotmail.com.ar y con copia oculta para lorena@yahoo.com.ar y lorena@elearning-total.com.ar

La dirección de respuesta la configuraremos a lorena@gmail.com

```
<?php
$destinatario = "lorena@gmail.com";
$asunto = "Este mensaje es una prueba";
$cuerpo = '
<html>
<head>
<title>Prueba de correo</title>
</head>
<body>
<h1>Hola amigos!</h1>
<p>
```

```
<strong>Bienvenidos a este correo electrónico de prueba</strong>. Estamos haciendo una prueba del funcionamiento de mail(). Este cuerpo del mensaje corresponde a la prueba de envío de mails por PHP. Habría que cambiarlo para poner tu propio cuerpo. De igual manera, debería cambiarse también la cabecera del mensaje.
</p>
</body>
</html>
';
//para el envío en formato HTML
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";
//dirección del remitente
$headers .= "From: Lorena < lorena@gmail.com >\r\n";
//dirección de respuesta, si queremos que sea distinta que la del remitente
$headers .= "Reply-To: lorena@gmail.com.ar\r\n";
//direcciones que recibián copia
$headers .= "Cc: lorena@hotmail.com \r\n";
//direcciones que recibirán copia oculta
$headers .= "Bcc: lorena@yahoo.com.ar, lorena@elearning-total.com \r\n";
mail($destinatario,$asunto,$cuerpo,$headers);
?>
```

Nota: Antes de poner en marcha el script en el servidor, por favor, cambien los datos de configuración de las direcciones de correo que van a recibir el mensaje y colocar unas direcciones que sean de uds. para que puedan comprobar si los mensajes se envían correctamente y para evitar que me lleguen cientos de mensajes de prueba!

La utilización del operador de asignación ".=" se usa para tomar el valor que ya posee una variable y agregarle más contenido, es decir conserva lo que tiene y le suma otro contenido adicional.

INCLUYENDO ARCHIVOS

Las construcciones include y require son de las más conocidas en php. Con ellas puedes reutilizar porciones de código (script, o simple html) cuantas veces quieras, siendo uno de sus usos más sencillos y típicos el de incluir cabeceras y pies de páginas en un sistema de plantillas.

Include ()

La sentencia include() inserta y evalúa el archivo especificado. Se puede incluir aquí no solamente un archivo de nuestro servidor, sino también una página web remota (indicando la url).

Su uso típico sería:

```
<?php
include("header.php");
?>
```

Include(); llama al archivo header.php y lo inserta en el propio punto del script donde hacemos la llamada.

Tanto si insertamos un archivo con include() o require(), debemos tener en cuenta que PHP pasa a *modo html* hasta el final del mismo, por lo que si el archivo a insertar contiene código php que deba ser evaluado (ejecutado), debe ser encerrado dentro de etiquetas de comienzo y fin de PHP.

Podemos también utilizar varios include anidados (es decir, utilizar include para llamar a otro archivo, dentro del archivo a incluir), con la única precaución de tener en cuenta que los archivos que se van insertando se ejecutan en el entorno del archivo primero que contiene la llamada.

Ejemplo:

Vamos a usar tres archivos, que fusionaremos. Luego observaremos el código de salida.

Archivo 1 : header.php :

```
<html>
<head>
<title> Muestra de includes </title>
```

```
</head> <body>
```

Archivo 2: footer.php :

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

Archivo 3: union.php :

```
<?php include("header.php"); ?>
<p>
Hola, este es el contenido.
</p>
<?php include("footer.php"); ?>
```

Ejecutamos unión.php y el resultado sería la suma de los 3 archivos, es decir:

```
<html>
<head>
<title> Muestra de includes </title>
</head>
<body>
<p>
Hola, este es el contenido.
</p>
</body>
</html>
```

SESSION

Las sesiones son una forma sencilla de almacenar datos para usuarios de manera individual usando un ID de sesión único. Esto se puede usar para hacer persistente la información de estado entre peticiones de páginas.

Los ID de sesiones normalmente son enviados al navegador mediante cookies de sesión, y el ID se usa para recuperar los datos de sesión existente. La ausencia de un ID o una cookie de sesión permite saber a PHP para crear una nueva sesión y generar un nuevo ID de sesión.

Las sesiones siguen un flujo de trabajo sencillo. Cuando una sesión se inicia, PHP recuperará una sesión existente usando el ID pasado (normalmente desde una cookie de sesión) o, si no se pasa una sesión, se creará una sesión nueva.

PHP rellenará la variable superglobal **$_SESSION** con cualesquiera datos de la sesión iniciada. Cuando PHP se cierra, automáticamente toma el contenido de la variable superglobal **$_SESSION**, la serializa, y la envía para almacenarla usando el gestor de almacenamiento de sesiones.

Las sesiones se puede iniciar manualmente usando la función **session_start().** Esta función debe incluirse al inicio del archivo .php

Ejemplo:

```
<?php
session_start();
if (!isset($_SESSION['count'])) {
  $_SESSION['count'] = 0;
} else {
  $_SESSION['count']++;
}
?>
```

Desregistrar una session

Para borrar una session usamos la función unset()

```
<?php
session_start();
unset($_SESSION['count']);
?>
```

Destruir una session

session_destroy() destruye toda la información registrada de una sesión

```
<?php
session_start();
session_destroy();
?>
```

Resumen

En esta Unidad…

En la presente unidad trabajamos con Funciones de PHP.

En la próxima Unidad…

Trabajaremos con la integración de PHP y MySQL