APRENDERAPROGRAMAR.COM

FUNCIONES PHP: DECLARACIÓN Y LLAMADAS. PARÁMETROS, RETURN. EJERCICIOS EJEMPLOS RESUELTOS. (CU00827B)

Sección: Cursos
Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº27 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez

FUNCIONES EN PHP

Una de las herramientas más importantes en cualquier lenguaje de programación son las funciones. Una función es un conjunto de instrucciones que a lo largo del programa van a ser ejecutadas multitud de veces. Es por ello, que este conjunto de instrucciones se agrupan en una función. Las funciones pueden ser llamadas y ejecutadas desde cualquier punto del programa.

Además, una función puede recibir parámetros externos de los cuales dependa el resultado de dicha función. Es decir, según el parámetro o parámetros con los que se invoque a la función, ésta devolverá un resultado u otro.

Las funciones deben estar definidas antes de realizar la llamada a la función (como es lógico).

Sintaxis general para declarar una función en PHP:

```
function nombre (parámetro1, parámetro2, ..., parámetroN) {
        instrucción1
        instrucción2
        .
        .
        .
        instrucciónN
}
```

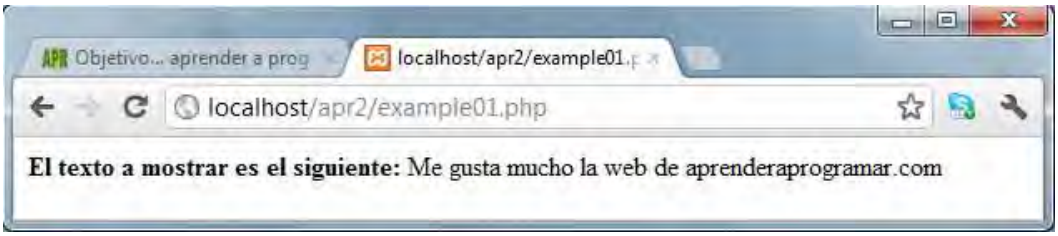
Para llamar (hacer que se ejecute) la función usaremos esta sintaxis: nombre(par1, par2, par3, ..., parN); donde par1, par2, par3, ..., parN son los parámetros (información) que le pasamos a la función. Una función puede necesitar de ningún, uno o varios parámetros para ejecutarse.

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo1.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```php
<?php //Ejemplo funciones  aprenderaprogramar.com
//Declaración de funciones
function mostrarTexto($texto) {
        echo "<strong>El texto a mostrar es el siguiente: </strong>";
        echo $texto;
}
//Fin de la declaración de funciones

mostrarTexto("Me gusta mucho la web de aprenderaprogramar.com");
?>
```

© aprenderaprogramar.com, 2006-2029



En este ejemplo hemos visto cómo hemos definido una función cuyo nombre es mostrarTexto. Esta función espera un parámetro cuando es invocada (parámetro que se ha denominado $texto). Una vez se ejecuta, la función ejecuta una serie de instrucciones y devuelve el control al punto desde el que fue invocada.

Podemos hacer varios comentarios:

a) En algunos lenguajes de programación se distinguen los términos "procedimiento" cuando un fragmento de código de este tipo ejecuta una serie de instrucciones sin devolver un valor, frente al término "función" que se aplica cuando un fragmento de código de este tipo ejecuta una serie de instrucciones y devuelve un valor. En PHP no se distingue entre una cosa y otra, simplemente se habla de "función" en general.

b) En algunos lenguajes de programación como Java el tipado o especificación de tipos que se van a recibir por parte de la función (o el tipo de dato que va a devolver la función) es mucho más fuerte. Si te fijas, la función mostrarTexto recibe un parámetro denominado $texto, pero en ningún lado se especifica si dicho parámetro es tipo integer, float, double ó string. ¿De qué tipo es? Realmente no lo sabemos: el intérprete PHP se encarga de automáticamente reconocer el tipo que se le pasa a la función. Además, intentará ejecutar el código sea como sea el tipo del parámetro pasado. Si le resultara imposible ejecutar el código, devolvería un error.

La utilidad fundamental de las funciones es no tener que repetir partes de código comunes, que sería necesario repetir varias veces. Esas partes de código comunes se agrupan en funciones y simplemente llamaremos a la función cada vez que necesitamos ejecutar ese código. De esta manera, evitamos la repetición que hace más largo y difícil de entender un programa o desarrollo web.

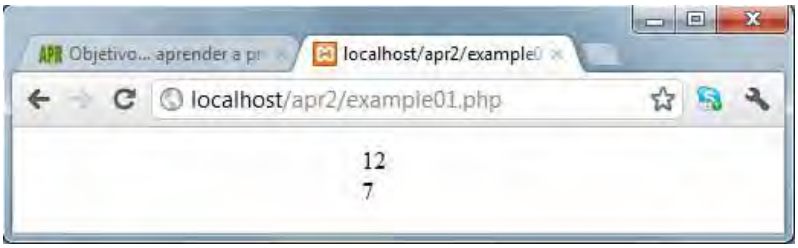
También podemos crear funciones que devuelvan datos (valores concretos). Estas funciones, que podríamos denominar "funciones en sentido estricto", son aquellas que ejecutan un código y como punto final de dicho código incluyen una sentencia return seguida del resultado de la función. La sentencia return indica que cuando se alcanza se ha llegado al final de la función y se devuelve como resultado de la misma el contenido especificado a continuación del return. Después de un return puede devolverse una variable, un número, una cadena de texto, etc.

Por ejemplo return "No dispone de permisos" significa que la función devuelve esta cadena de texto. Otro ejemplo: return $calculo; indica que la función devuelve el contenido que se encuentre almacenado en la variable $calculo. Otro ejemplo: return "Lo sentimos ".$usuario." pero no dispone de permisos. Para solicitar información puede escribir a ".$emailAdministrador; haría que la función devuelva una cadena de texto donde intervienen diversas variables.

© aprenderaprogramar.com, 2006-2029

Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo2.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```php
<?php // Ejemplo funciones  aprenderaprogramar.com
function operaciones($n1, $n2, $operacion) {
	$resultado = 0;
	if($operacion == "Sumar") {
		$resultado = $n1 + $n2;
	}else if($operacion == "Restar") {
		$resultado = $n1 - $n2;
	}else if($operacion == "Multiplicar") {
		$resultado = $n1 * $n2;
	}
	return $resultado; // Devolver el resultado
}
// Llamar a la función operaciones
$r = operaciones(5, 7, "Sumar");
echo $r . "<br>";
// O podemos imprimir directamente
echo operaciones(15, 8, "Restar");
 ?>
```



Fíjate que a diferencia de la función mostrarTexto, la función operaciones nos devuelve un valor concreto, de forma que se sustituye su invocación allí donde aparece por el valor que devuelve. Así, la instrucción echo operaciones(15, 8, "Restar"); equivale a lo que sería escribir echo <<aquí el valor devuelto por la función operaciones invocada con los parámetros 15, 8 y "Restar" >>, es decir, sería lo mismo que escribir echo (15-8); ó echo 7;

Además, fíjate que la función mostrarTexto requería un parámetro, mientras que la función operaciones requiere tres parámetros. Si invocas la función sin pasarle el número de parámetros adecuado obtendrás un error del tipo <<Warning: Missing argument 3 for operaciones()>>.

Por último, tener en cuenta que una función puede ser invocada sin parámetros.

```php
<?php //Ejemplo funciones  aprenderaprogramar.com
function mostrarTextoError {
	echo "<strong>Se ha producido un error </strong>";
	// Aquí pueden venir varias líneas de instrucciones
}
?>
```

© aprenderaprogramar.com, 2006-2029

Esta función carece de parámetros. Para invocarla escribiríamos mostrarTextoError(). Cada vez que realizáramos la invocación se ejecutaría el código dentro de la función. Esta función podemos decir que es "tipo procedimiento" porque no devuelve un resultado (no tiene sentencia return).

EJERCICIO

Crear las siguientes funciones en PHP y código para comprobar su funcionamiento:

a) Una función que reciba cinco números enteros como parámetros y muestre por pantalla el resultado de sumar los cinco números (tipo procedimiento, no hay valor devuelto).

b) Una función que reciba cinco números enteros como parámetros y devuelva el resultado de sumar los cinco números (tipo función, hay un valor devuelto). Asigna el resultado de una invocación a la función con los números 2, 5, 1, 8, 10 a una variable de nombre $tmp y muestra por pantalla el valor de la variable.

c) Una función que reciba como parámetros el valor del radio de la base y la altura de un cilindro y devuelva el volumen del cilindro, teniendo en cuenta que el volumen de un cilindro se calcula como Volumen = númeroPi * radio * radio * Altura siendo númeroPi = 3.1416 aproximadamente.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00828B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193

© aprenderaprogramar.com, 2006-2029