

Tema 3 : Funciones



Ciclo Superior DAW

Asignatura: Desarrollo web en entorno servidor

Curso 20/21

Introducción



- En este capítulo veremos cómo crear y utilizar las funciones
- Aprenderemos a pasar parámetros
- Veremos cómo referenciar varias páginas



Funciones

- La definición de funciones en PHP sirve para evitar escribir el mismo código varias veces
- Las funciones deben definirse antes de utilizarlas
- Las funciones pueden no tener argumento o tener argumentos por valor o por referencia



Funciones

- Los nombres de las funciones siguen las mismas reglas de los identificadores de PHP:
 - Deben comenzar por una letra o un guión bajo (_)
 - El resto de caracteres pueden ser letras, números o guiones bajos
 - Los nombres de funciones no distinguen entre mayúsculas o minúsculas.



Funciones. Sintaxis

- La sintaxis de una función es:

```
<?php
function suma($arg_1, $arg_2, ..., $arg_n)
{
    echo "Función suma.\n";
    return $retval;
}
?>
```



Funciones. Parámetros

- Los parámetros son variables y/o constantes separadas por comas.
- Se pueden pasar parámetros por valor, por referencia, y parámetros por defecto:
 - **Por valor:** El valor del argumento dentro de la función no cambia fuera de ella
 - **Por referencia:** El valor del argumento dentro de la función Sí cambia fuera de ella. Se debe añadir un & al parámetro. Por ejemplo: `function anhadir_algo(&variable)`
 - **Por defecto:** Se le puede asignar un valor por defecto a un parámetro



Funciones. Parámetros por valor

- **El valor del argumento dentro de la función no cambia fuera de ella**
- Es decir, si cambias el valor del argumento dentro de la función, no se ve modificado fuera de ella



Funciones. Parámetros por valor

- Por ejemplo:

```
<?php
function raiz_cuadrada($num)
{
    return $num * $num;
}
echo raiz_cuadrada(4);    // muestra '16'.
?>
```




Actividad 1

Crea una función Insertar_Persona, donde le pases los argumentos:

- *Nombre*
- *Apellidos*
- *Edad*
- *DNI*

Esta función debe mostrar estos argumentos por pantalla



Funciones. Parámetros por referencia

- El valor del argumento dentro de la función **SÍ** cambia fuera de ella.
- Si queremos que un parámetro de una función siempre **se pase por referencia**, debemos anteponer un ampersand (&) al nombre del parámetro en la definición de la función.



Funciones. Parámetros por referencia

- Por ejemplo:

```
<?php
function anhadе_algo(&$string)
{
    $string .= ' y algo más.';
}
$str = 'Esto es una cadena, ';
anhadе_algo($str);
echo $str; // Muestra 'Esto es una cadena, y algo más.'
?>
```



Actividad 2

Crea una función Aumentar_Edad, donde le pases la variable \$edad y la aumente 10 años.

Debes mostrar por pantalla la variable antes y después de modificarla



Funciones. Parámetros por defecto

- Podemos definir funciones con argumentos predeterminados, de manera que, si en la llamada a la función no se envía un argumento, la función asume un valor predeterminado.
- Los argumentos predeterminados deben ser los últimos en la lista de argumentos, para evitar ambigüedades.



Funciones. Parámetros por defecto

- Por ejemplo:

```
<?php
// ESTA ES LA DEFINICIÓN DE LA FUNCIÓN calculaMedia
function calculaMedia($arg1, $arg2, $arg3 = "aritmética")
{
    if ($arg3 == "aritmética") {
        $media = ($arg1 + $arg2) / 2;
    } elseif ($arg3 == "geométrica") {
        $media = sqrt($arg1 * $arg2) / 2;
    } elseif ($arg3 == "armónica") {
        $media = 2 * ($arg1 * $arg2) / ($arg1 + $arg2);
    }
    return $media;
}

// ESTO SON EJEMPLOS DE USO DE LA FUNCIÓN calculaMedia
$dato1 = 12;
$dato2 = 16;

// EL TERCER ARGUMENTO INDICA EL TIPO DE MEDIA A CALCULAR
$media = calculaMedia($dato1, $dato2, "geométrica");
print "<p>La media geométrica de $dato1 y $dato2 es $media.</p>\n";

// AL NO PONER EL TERCER ARGUMENTO, LA FUNCIÓN DEVUELVE LA MEDIA ARITMÉTICA
$media = calculaMedia($dato1, $dato2);
print "<p>La media aritmética de $dato1 y $dato2 es $media.</p>\n";
?>
```

Actividad 3



Modifica la Actividad 2 para que la edad sea un parámetro por defecto



Funciones. Devolviendo valores

- Por ejemplo:

```
<?php
function raiz_cuadrada($num)
{
    return $num * $num;
}
echo raiz_cuadrada(4);    // muestra '16'.
?>
```




Funciones. Devolviendo valores

- **Para devolver valores usamos la instrucción opcional return.**
- Puede devolverse cualquier tipo de valor, incluyendo listas y objetos.



Actividad 4

Crea una función Obtener_PVP, donde le pases los argumentos:

- *Precio bruto*
- *IVA*

Esta función debe DEVOLVER el PVP, lo recogeremos en una variable

\$precio_final y lo mostraremos por pantalla



Funciones integradas

- Son un amplio conjunto de funciones que forman parte de la librería PHP.
- Pueden ser de distinto tipo:
 - Manejo de arrays o cadenas
 - Manejo de fechas
 - Etc



Funciones integradas

- Algunos ejemplos de funciones integradas:
 - **strlen**: Obtiene la longitud de una cadena de texto

```
<?php
$str = 'abcdef';
echo strlen($str); // 6
```



Funciones integradas

- Algunos ejemplos de funciones integradas:
 - **max**: Encontrar el valor más alto

```
<?php  
echo max(2, 3, 1, 6, 21); // 21
```



Funciones integradas

- Algunos ejemplos de funciones integradas:
 - **sqrt**: Calcula la raíz cuadrada

```
<?php  
echo sqrt(64); // 8
```



Funciones integradas

- Podéis encontrar todas las funciones existentes en:

<https://www.php.net/manual/es/indexes.functions.php>



Funciones dentro de funciones

- En PHP, cualquier instrucción válida puede aparecer en el cuerpo de la función, incluso otras funciones y definiciones de clases.
- Esto nos permite encadenar funciones.
- PHP no soporta la redefinición (sobrecarga) de funciones previamente declaradas.



Funciones dentro de funciones

- Por ejemplo:

```
<?php
function funcion_principal()
{
    function funcion_secundaria()
    {
        echo "Yo no existiré hasta que funcion_principal() sea invocada.\n";
    }
}
/* No podemos llamar a funcion_secundaria() hasta que exista. */
funcion_principal();
/* Ahora podemos llamar a funcion_secundaria() */
funcion_secundaria();
?>
```



Actividad 5

Crea la función `Crear_Equipo`, y dentro de ella crea las funciones:

- *`Seleccionar_Portero`*
- *`Seleccionar_Defensa`*
- *`Seleccionar_Medio`*
- *`Seleccionar_Delantero`*

Asegúrate que las funciones interiores solamente se puedan utilizar si

previamente has invocado la función principal



Funciones variables

- Si una variable tiene un paréntesis al final, PHP buscará una función con el mismo nombre que la evaluación de la variable, e intentará ejecutarla.
- Las funciones variables no funcionarán con palabras reservadas, tal como `echo()`, `print()`, `unset()`, `isset()`, `empty()`, `include()`, `require()` y derivados.



Funciones variables

- Por ejemplo:

```
<?php
function suma()
{
    echo "Dentro de suma()<br>\n";
}
function resta($arg = '')
{
    echo "Dentro de resta(); El argumento era '$arg'.<br>\n";
}

$func = 'suma';
$func();      // Llama a suma()
$func = 'resta';
$func('resta'); // Llama a resta()
?>
```



Funciones de cadena

FUNCIÓN	DESCRIPCIÓN
strlen(\$cadena)	Devuelve el número de caracteres de una cadena
substr(\$cadena, \$inicio[, \$largo])	Devuelve una subcadena de una cadena, desde la posición \$inicio y con la longitud \$largo si está presente (si no, hasta el final)
strpos(\$cadena, \$subcadena[, \$inicio])	Devuelve la posición de \$subcadena dentro de \$cadena (a partir de \$inicio, si está presente). FALSE ó 0 si no se encuentra.



Funciones de cadena

FUNCIÓN	DESCRIPCIÓN
strtoupper(\$cadena)	Devuelve la cadena en mayúsculas
strtolower(\$cadena)	Devuelve la cadena en minúsculas
trim(\$cadena)	Devuelve la misma cadena pero eliminando los espacios del principio y el final
ltrim(\$cadena[, \$caracteres])	Elimina los espacios del principio del String. Se le puede pasar una lista de caracteres y también los eliminará.
str_replace(\$txtBuscar, \$txtReemplazar, \$cadena[, \$numero])	Cambia las ocurrencias de \$txtBuscar en \$cadena por \$txtReemplazar (si se indica \$numero, guarda el número de reemplazos realizados)

Funciones de cadena



FUNCIÓN	DESCRIPCIÓN
strrev(\$cadena)	Devuelve la cadena invertida.



Funciones de cadena

Ejemplo strlen()

```
$cadena = "Ciclos Montecastelo";  
$resultado = strlen($cadena);  
//resultado vale 19, la longitud de la cadena
```




Funciones de cadena

Ejemplo substr(\$cadena, \$inicio[, \$largo])

```
$cadena = "Ciclos Montecastelo";  
$prueba1 = substr($cadena,1); //prueba1 vale "Ciclos"  
$prueba2 = substr($cadena,2,4); //prueba2 vale "iclo"  
//Si se le pasa el tercer parámetro en negativo omite esas posiciones al final  
$prueba3 = substr($cadena,2,-1); //prueba3 vale "iclos Montecastel"  
$prueba4 = substr($cadena,-3); //prueba4 vale "elo"
```



Funciones de cadena

Ejemplo strpos(\$cadena, \$subcadena[, \$inicio])

```
$cadena = "Ciclos Montecastelo";  
$prueba = strpos($cadena, "Montecastelo");  
//Recuerda que si no la encuentra devuelve 0 o false, y si la encuentra puede estar en la posición 0.  
if($prueba === false){ //Hay que comprobar si no devuelve false.  
    echo "La cadena 'Montecastelo' no se encontró  
}  
else{  
    echo "La cadena está en la posición $prueba";  
}
```



Funciones de cadena

Ejemplo strtoupper(\$cadena)

```
$cadena = "Ciclos Montecastelo";  
$prueba = strtoupper($cadena);  
//prueba vale "CICLOS MONTECASTELO"
```



Funciones de cadena

Ejemplo strtolower(\$cadena)

```
$cadena = "Ciclos Montecastelo";  
$prueba = strtolower($cadena);  
//prueba vale "ciclos montecastelo"
```



Funciones de cadena

Ejemplo trim(\$cadena)

```
$cadena = "    Ciclos Montecastelo  ";  
$prueba = trim($cadena);  
//prueba vale "Ciclos Montecastelo"
```



Funciones de cadena

Ejemplo ltrim(\$cadena[\$caracteres])

```
$cadena = "    Ciclos Montecastelo  ";  
$prueba = ltrim($cadena);  
//prueba vale "Ciclos Montecastelo  ";
```



Funciones de cadena

Ejemplo `str_replace($txtBuscar, $txtReemplazar, $cadena[, $numero])`

```
$cadena = "Ciclos Montecastelo";  
$resultado = str_replace("Ciclos", "Colegio");  
//resultado vale "Colegio Montecastelo"
```



Funciones de cadena

Ejemplo strrev(\$cadena)

```
$cadena = "Ciclos Montecastelo";  
$prueba = strrev($cadena);  
//prueba vale "oletsaCetnoM solciC"
```




Actividad 6

Crea la siguiente función **string Funcion_Multifuncion(\$tipoFuncion, \$texto)**

- *Pásale una constante (numérica o de cadena) para que seleccione una función de cadena.*
- *Según lo recibido en \$tipoFuncion, llamaremos a cada una de las funciones de cadena que hemos visto (Son 9)*



Include y Require

- Siguiendo el concepto de encapsulación de código, y para facilitar el mantenimiento de una aplicación escrita en PHP, podemos colocar parte del código en distintos ficheros.
- En el programa principal hacemos referencia a ellos mediante sentencias **include** o **require**.



Actividad 7

- *Crea una funcion llamada Me_Llaman_Desde_Otra_Pagina() y métela en un fichero llamado Pagina2.php.*
- *Esta función debe mostrar por pantalla el texto “Me llaman desde la página 1”*
- *Crea un fichero llamado Pagina1.php y, utilizando Include o Require, invoca la función Me_Llaman_Desde_Otra_Pagina()*

Tema 3 : Funciones



Ciclo Superior DAW

Asignatura: Desarrollo web en entorno servidor

Curso 20/21