

Funciones

Programando con PHP

Introducción

Funciones

Introducción

- * Sección de código separada a la que se le ha dado un nombre.
- * Puede ser llamada tantas veces como se quiera.
- * Dividen tareas para hacer más comprensible el script PHP
- * Pueden recibir / devolver valores

Introducción

* Sintaxis:

```
Function nombreFuncion (parametros)
{
    Sentencias
}
```

Introducción

- * Para llamar a una función sólo hay que escribir una sentencia que contenga el nombre de la función, seguida de los parámetros.
- * La ejecución salta a la línea en la que empieza la función.

Ejemplo

```
<?php
```

```
function cuentaAtras()
```

1

```
for ($i=10; $i>0; $i--)
```

1

```
echo $i. "...<br>"
```

}

```
echo "Booooooooooooo"
```

3

?

 > |

```
<?php
```

```

cuentaAtras();

```

?

```
<td bgcolor = 'pink'>
```

```
<?php
```

```

cuentaAtras();

```

?

Ejemplo

10...

9...

8...

7...

6...

5...

4...

3...

2...

1...

¡ Booooooooommmmm !

10...

9...

8...

7...

6...

5...

4...

3...

2...

1...

¡ Booooooooommmmm !

Introducción

- *En la mayoría de las ocasiones, necesitaremos parámetros.
- *Por ejemplo...

Ejemplo

```
<?php
function cuentaAtras($inicio)
{
    for ($i=$inicio; $i>0; $i--)
    {
        echo $i."...<br>";
    }
    echo "¡   Booooooooooooommmmm   !";
}

?>
```

```
<td bgcolor = '#FFBBAA'>
```

```
<?php
    cuentaAtras(8);
```

```
?>
```

```
</td>
```

```
<td bgcolor = 'pink'>
```

```
<?php
    cuentaAtras(15);
```

```
?>
```

Ejemplo

8...
7...
6...
5...
4...
3...
2...
1...

¡ Booooooooommmmm !

15...
14...
13...
12...
11...
10...
9...
8...
7...
6...
5...
4...
3...
2...
1...

¡ Booooooooommmmm !

Paso de parámetros

Funciones

Paso de parámetros

- *PHP permite pasar los parámetros:
 - *Por valor
 - *Por referencia
 - *Por defecto

Paso de parámetros por valor

Paso de parámetros por valor

- * Es la opción por defecto en PHP
- * La función recibe una copia del valor de la variable.
- * No hay problema en modificarla.

```
<?php
```

```
function muestraMasUno ($numero)
```

```
{
```

```
    $numero++;
```

```
    echo $numero;
```

```
    echo "<br>";
```

```
}
```

```
$num = 3;
```

```
echo "num contiene: $num<br>";
```

```
echo "llamando a muestraMasUno ";
```

```
muestraMasUno ($num) ;
```

```
echo "num todavia contiene: $num<br>";
```

```
?>
```

num contiene: 3

llamando a muestraMasUno 4

num todavia contiene: 3

Paso de parámetros por referencia

Paso de parámetros por referencia

- * En algunas ocasiones necesitaremos modificar el valor original del parámetro recibido.
- * PHP permite recibir la **dirección** de memoria de la variable original
- * Sólo habrá que añadir un **&** delante del parámetro en la declaración de la función

```
<?php
```

```
function sumaUno (&$num)
{
    $num++;
    echo "Dentro de la funcion: $num<br>";
}
```

```
$numero = 4;
sumaUno ($numero) ;
```

```
echo "Después de la funcion: $numero";
```

```
?>
```

Dentro de la funcion: 5
Después de la funcion: 5

Paso de parámetros por defecto

Paso de parámetros por defecto

- * Son la forma en que PHP implementa los parámetros opcionales.
- * Toman el valor predefinido, cuando al llamar la función, no se les pasa un argumento.

Paso de parámetros por defecto

- * Para definirlos hay que escribir, en la cabecera de la función:
 - * Nombre del parámetro
 - * Operador de asignación (=)
 - * Valor por defecto
- * Los parámetros pasados por defecto deben ir **SIEMPRE** los últimos.

```
<?php
```

```
function aumenta ($numero, $suma=1)
{
    $numero=$numero+$suma;
    echo $numero;
    echo "<br>";
}
```

```
$num = 3;
echo "num contiene: $num<br>";
echo "llamando a aumenta con uno más ";
aumenta($num);

echo "Llamando a aumenta con 3 más ";
aumenta ($num, 3);
```

```
?>
```

num contiene: 3
llamando a aumenta con uno más 4
Llamando a aumenta con 3 más 6

Ámbito de las variables

Funciones

Ámbito de las variables

- * Variables globales.
 - * Definidas fuera del cuerpo de una función.
 - * Accesibles desde cualquier punto del código.
 - * Para ello deben ser declaradas:
 - * Anteponiendo la palabra clave *'global'*
 - * Usando el array asociativo `$GLOBALS` (con el nombre de la variable como clave y su contenido como valor)

Ejemplo: Variables globales

```
<?php
    function saludo ()
    {
        $nombre = $GLOBALS['nom'];
        echo "hola $nombre";
    }

    global $nom;
    $GLOBALS['nom']='cristina';
    saludo();
```

```
?>
```

Ámbito de las variables

- * Locales

- * Definidas dentro del cuerpo de una función.
- * Accesibles **SÓLO** desde esa función.
- * Si se declara una variable 'local' dentro de una función, con el mismo nombre que una variable 'global', se trabajará con la variable 'local'

Ámbito de las variables

- * Variables estáticas
 - * Son variables locales.
 - * No pierden su valor entre las diferentes llamadas a la función que las definen.
 - * Hay que anteponerle la palabra *static*.

Ejemplo

```
<?php
    $inicio = 9;
    $final = 0;
    function cuentaAtras()
    {
        //variable local
        $inicio = 7;
        //variable global
        global $final;
        //variable estática
        static $num=0;
        for (; $inicio>$final; $inicio--)
            echo $inicio."...<br>";
        $num++;
        echo "¡Booommm - $num- !";
    }

```

?>

```
<?php
    // $num valdrá 1
    cuentaAtras();
?>
</td>
<td bgcolor = 'pink'>
<?php
    // $num valdrá 2
    cuentaAtras();

```

...

Ejemplo

7...

6...

5...

4...

3...

2...

1...

¡Booommm -1- !

7...

6...

5...

4...

3...

2...

1...

¡Booommm -2- !

Devolución de variables

Funciones

Devolución de valores

- * Se pueden devolver cualquier tipo de valor.
- * **SÓLO** se puede devolver un valor
- * Para devolver varios valores, devolveremos un array.
- * Utilizaremos la palabra reservada *return*
- * Cuando se encuentra un *return* la función termina.

Ejemplo

```
<?php
```

```
function elMayor ($dato1, $dato2, $dato3)
{
    $elmayor=($dato1>$dato2)?$dato1:$dato2;
    $elmayor=($elmayor>$dato3)?$elmayor:$dato3;
    return $elmayor;
}
```

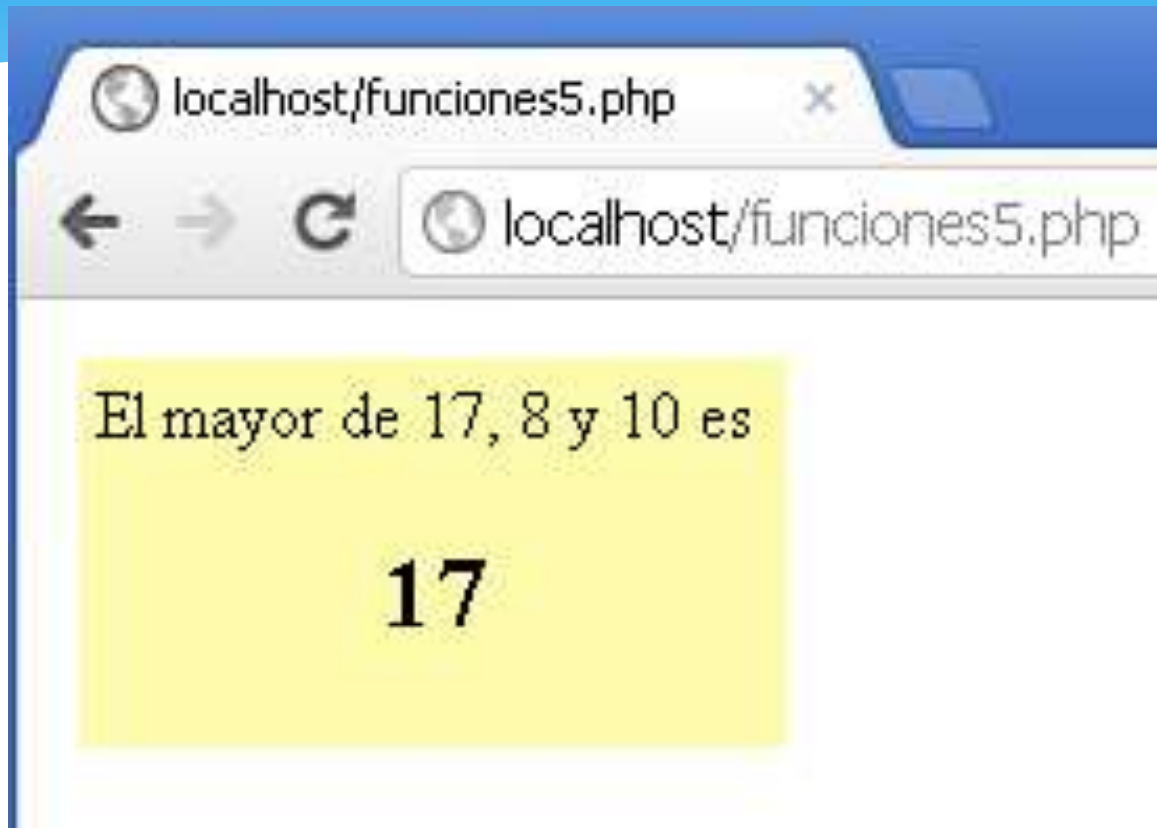
```
?>
```

```
<?php
```

```
$mayor = elMayor(17, 8, 10);
echo "El mayor de 17, 8 y 10 es ";
echo "<br><h2>$mayor</h2>";
```

```
?>
```


Ejemplo



Funciones con número variable de parámetros

Funciones

Funciones con número variable de parámetros

- * PHP nos permite definir funciones en las que el número de parámetros no está fijado a priori.
- * No se necesita una sintaxis específica, pero su funcionamiento se basa en el uso de:

Funciones con número variable de parámetros

- *func_num_args()

- * Devuelve el número de parámetros recibidos por la función

Funciones con número variable de parámetros

- * `func_get_args()`
 - * Devuelve un array con los argumentos pasados a la función
- * `func_get_arg()`
 - * Devuelve un elemento de la lista de argumentos pasados.
 - * Los argumentos empiezan en la posición CERO
 - * Si se solicita una posición que no existe => *false*

Ejemplo

El mayor de 17, 8 , 10 y 35 es

35

```
function elMayor ($dato1, $dato2)
{
    $num_argumentos = func_num_args();
    $argum = func_get_args();

    $elmayor=($dato1>$dato2)?$dato1:$dato2;
    for($i=2; $i<$num_argumentos; $i++)
        $elmayor=($elmayor>$argum[$i])?$elmayor:$argum[$i];
    return $elmayor;
}
```

```
$mayor = elMayor(17, 8, 10, 35);
echo "El mayor de 17, 8 , 10 y 35 es ";
echo "<br><h2>$mayor</h2>";
```

Incluir ficheros

Funciones

Incluir ficheros

- * En PHP podemos crear un archivo .php que incluya una serie de funciones o variables que usamos frecuentemente.
- * Podremos incluir ese fichero en cualquier otro fichero utilizando la función **require_once**
- * Con **require_once** indicamos la ruta relativa del archivo que queremos incluir.


```
<?php
```

```
$color='rojo';  
$edadMinima=12;
```

```
Function suma ($num1, $num2)
```

```
{  
    $sum = $num1+$num2;  
    return $sum;  
}
```

```
Function cuentaNumeros ($num1, $num2)
```

```
{  
    if($num1<$num2)  
    {  
        for ($i = $num1; $i<$num2; $i++)  
        {  
            echo "$i <br>";  
        }  
    }  
    else  
    {  
        for ($i = $num2; $i<$num1; $i++)  
        {  
            echo "$i <br>";  
        }  
    }  
}
```

```
?>
```

Fichero funciones.php

```
<?php
```

```
    require_once "funciones.php";
```

```
    $numero1 = 2;
```

```
    $numero2 = 15;
```

```
    $sum = suma($numero1, $numero2);
```

```
    echo "La suma de $numero1 y $numero2 es igual a $sum<br>";
```

```
    echo "<h1> Mostrando números entre $numero1 y $numero2</h1><br>";  
    cuentaNumeros($numero1, $numero2);
```

```
    echo "<br><b>La variable color contiene $color";
```

```
    echo "<br>La variable edadMinima contiene $edadMinima";
```

```
?>
```

Fichero principal.php

La suma de 2 y 15 es igual a 17

Mostrando numeros entre 2 y 15

2
3
4
5
6
7
8
9
10
11
12
13
14

Ejecución de
principal.php

La variable color contiene rojo

La variable edadMinima contiene 12

Funciones

Programando con PHP