



# Unidad Didáctica 2

## *GENERACIÓN DE DOCUMENTOS WEB CON GUIONES DE SERVIDOR*

### 2.5 Funciones

[www.docencia.taboadaleon.es](http://www.docencia.taboadaleon.es)

# ¿Qué es una función?

- Conjunto de sentencias que se ejecutan al ser invocadas desde otra parte del script
- Permiten reutilizar el código
- Pueden devolver un valor de retorno
- Pueden ser predefinidas o definidas por el usuario

Definición

```
<?php
// función que comprueba si el usuario es correcto
function loginCorrecto($nombre,$clave){
    if ($nombre=="juan" && $clave=="qwerty"){
        return 1;
    }else {return 0;}
}
```

Invocación

```
if (loginCorrecto("juan","qwerty")){
    echo "logedado correctamente, ". "saludos ";
}else {echo "acceso denegado";}

?>
```

# Funciones predefinidas

- PHP tiene una gran biblioteca de funciones que pueden utilizarse en cualquier momento:
  - funciones útiles para el tipo de variables
  - funciones que permiten manipular los arrays
  - funciones útiles sobre las cadenas de caracteres, las fechas, calendarios y los números, etc

# Ejemplo de funciones sobre cadenas

<b>strlen()</b>	<b>\$num = strlen(\$texto);</b> Devuelve el <b>número de caracteres</b> de una cadena.
<b>strtoupper()</b>	<b>\$texto2 = strtoupper(\$texto);</b> Devuelve la cadena de texto pasada en el argumento \$texto con todas sus letras en mayúsculas.
<b>str_replace()</b>	<b>\$nuevoTexto = str_replace(\$buscar,\$reemplazar,\$texto);</b> Reemplaza un trozo de texto por otro. Dentro de la cadena \$texto (tercer argumento), busca la subcadena \$buscar (primer argumento) y la reemplaza por la subcadena \$reemplazar (segundo argumento).
<b>trim()</b>	<b>\$nuevoTexto = trim(\$texto,"lista_caracteres");</b> Devuelve el \$texto (1º argumento), en el cual se eliminan tanto los primeros como los últimos caracteres cuando éstos están en la "lista_caracteres" (2º argumento).

# Ejemplo de funciones sobre fechas

- **Fechas:** PHP permite el manejo de fechas y horas.

`getdate()` : Devuelve un array asociativo con la información de la fecha y hora.

**Índices:** hours, minutes, seconds, mday, wday, weekday, mon, month, year, yday  
Esta función acepta un parámetro que indica los segundos desde 01/01/1970

`time()` : Devuelve el número de segundos transcurridos desde 01/01/1970

`mktime(hr, mt, sg, m, d, a)` : Devuelve los segundos transcurridos desde el 01/01/1970

`date(formato_fecha)` : Devuelve la fecha actual en una cadena formateada, según:

<i>d</i>	Día del mes, 2 dígitos con ceros iniciales	01 a 31
<i>D</i>	Una representación textual de un día, tres letras	<u>Mon</u> a <u>Sun</u>
<i>j</i>	Día del mes sin ceros iniciales	1 a 31
<i>l</i>	Una representación textual completa del día de la semana	<u>Sunday</u> a <u>Saturday</u>
<i>z</i>	El día del año (comenzando en 0)	0 a 365
<i>F</i>	Una representación textual completa de un mes	<u>January</u> a <u>December</u>
<i>m</i>	Representación numérica de un mes, con ceros iniciales	01 a 12
<i>M</i>	Una representación textual corta de un mes, tres letras	<u>Jan</u> a <u>Dec</u>
<i>n</i>	Representación numérica de un mes, sin ceros iniciales	1 a 12
<i>t</i>	Número de días en el mes dado	28 a 31
<i>Y</i>	Una representación numérica de un año, 4 dígitos	Ejemplos: 1999 o 2003
<i>g</i>	formato de 12-horas de una hora sin ceros iniciales	1 a 12
<i>G</i>	formato de 24-horas de una hora sin ceros iniciales	0 a 23
<i>h</i>	formato de 12-horas de una hora con ceros iniciales	01 a 12
<i>H</i>	formato de 24-horas de una hora con ceros iniciales	00 a 23
<i>i</i>	Minutos con ceros iniciales	00 a 59
<i>s</i>	Segundos, con ceros iniciales	00 a 59

# Ejemplos de funciones para fecha

- `<? echo "Son las ", date("h : i : s"), " y hoy es ", date("j-n-Y")?>`
- devolvería: **Son las 12 : 48 : 11 y hoy es 1-11-2011**

Ejemplos de <i>getdate()</i>		
Funcionalidad	Sintaxis	Devuelve
Devuelve un <i>array asociativo</i>	<code>\$\$=getdate(); echo \$\$;</code>	Array
Este índice devuelve los segundos de la hora actual	<code>echo \$\$[seconds]</code>	11
Este índice devuelve los minutos de la hora actual	<code>echo \$\$[minutes]</code>	48
Este índice devuelve la hora de la hora actual	<code>echo \$\$[hours]</code>	12
Este índice devuelve el día del mes actual	<code>echo \$\$[mday]</code>	1
Este índice devuelve el nº del día de la semana	<code>echo \$\$[wday]</code>	1
Este índice devuelve el nº del mes	<code>echo \$\$[mon]</code>	6
Este índice devuelve el año	<code>echo \$\$[year]</code>	2009
Este índice devuelve nº del día en el año actual	<code>echo \$\$[yday]</code>	151
Este índice devuelve el día de la semana	<code>echo \$\$[weekday]</code>	Monday
Este índice devuelve el nombre del mes	<code>echo \$\$[month]</code>	June

Actividades: Haz funcionPre I

# Creación de nuevas Funciones

- PHP tiene una gran biblioteca de funciones que pueden utilizarse en cualquier momento.
- Por otra parte, es posible definir nuestras propias funciones, llamadas **funciones de usuario**.
- El uso de funciones brinda la capacidad **de agrupar varias instrucciones bajo un solo nombre**. Esto permite usar esas instrucciones repetidas veces, en lugar de escribirlas nuevamente, invocando a la función con el nombre asignado.

Definición

```
<?php
function TradicionalHolaMundo(){
    echo "hola mundo";
}
```

Invocación

```
TradicionalHolaMundo(); // imprime "hola mundo"
?>
```

# Funciones – Valores por defecto

Existe la posibilidad de que la función realice una acción por defecto. Por ejemplo, si se desea que la función ponga por defecto el texto en color azul, la definimos de la siguiente forma:

## Definición

```
function color($cadena,$color="0000FF")
{ //imprime una cadena con el color deseado
  echo "<font color=#".$color.">".$cadena."</font>";
}
```

## Invocación

```
//color por defecto
color("IES San Sebastián");

//indicamos el color verde
color("IES San Sebastián","#009933");
```



# Funciones – Parámetros

Podemos pasar a nuestras funciones valores para que operen sobre ellos. PHP permite pasar a las funciones variables, cadenas de texto, números o arrays.

```
function nombre_función (parámetro1,parámetro2, ... ,parámetron)
{
    instrucción1;
    instrucción2;
    instrucción3;
    return valor_retorno;
}
```

# Funciones - ejemplo

```
<html>
<head></head>
<body>
<?php
function media($a,$b)
{
    $result=($a+$b)/2;
    return $result;
}
$a= 3;
$b=13;
$semi= media($a,$b);
echo("La semisuma de $a y $b es:".$.$semi);
?>
</body>
</html
```

# Funciones – Parámetros por referencia

Por defecto los parámetros se pasan por valor, es decir, que no sufren modificaciones dentro de la función.

Si se desea que los parámetros conserven las modificaciones sufridas dentro de la función. Es necesario pasarlos por referencia.

```
function MiFuncion(&$var)
{
    $var++;
}

$a = 5;
MiFuncion($a);
// Aquí $a == 6
```

**¡cuidado! Puede dar error**  
php.ini del servidor si tiene configurada en **Off**  
la directiva:  
`allow_call_time_pass_reference`

Actividades: Haz funcion I a función3

# Otra forma de definir funciones de usuario

-Definición

-Contenido

-Cierre

Invocación

```
<? function Encabezado() { ?>
<!-- Hemos abierto la función y cerrado la etiqueta PHP
todo esto es código HTML //-->
    <HTML>
    <HEAD>
    <TITLE>Titulo de mi página</TITLE></HEAD>
    <BODY BGCOLOR="#FF0000">
<!-- Esta nueva llamada a PHP
insertando la llave de cierre de la función
indicará a PHP que debe escribir todo lo
contenido entre la { y esta } //-->
<? } ?>

<? function Pie() { ?>
    <HR>
    </BODY>
    </HTML>
<? } ?>
<!-- Utilizaremos esas dos funciones para
crear una página web. Llamamos a la función Encabezado
luego escribimos un texto y por último insertamos
el Pie de página con la función Pie //-->
<? Encabezado(); ?>
Este es texto que aparecerá en el cuerpo de la página.
Está fuera de los scripts de php y será considerado
como un texto HTML. Debajo aparecerá la línea horizontal
que insertaremos mediante una nueva llamada a la función Pie

<? Pie(); ?>
```

# Unidad Didáctica 2

## 2.5 Funciones

**FIN DE LA PRESENTACIÓN  
GRACIAS**