

## Funciones de salida

### Funciones de salida

Ya conocemos algunas de las funciones que PHP utiliza para mostrar información –salidas– en la ventana del navegador del **cliente**.

Recordémoslas:

#### echo

La función **echo**, aunque admite también la forma **echo()**, no requiere de forma obligatoria el uso de los **paréntesis**.

Detrás de la **instrucción echo** pueden insertarse: variables, cadenas (éstas entre comillas) y números (éstos sin comillas) separadas por **comas**.

Este es un ejemplo de código:

```
$a=24; $b="Pepe";
$c="<br>";
echo $a,$b,25,
"Luis",$c;
```

que produciría esta salida:

24Pepe25Luis

Observa los **valores** que hay detrás de **echo**. Como verás, no es necesario insertar todos los valores en la misma línea.

#### print()

La función **print()** sólo puede contener **dentro del paréntesis** una sola variable, o el conjunto de varias de ellas enlazadas por un punto.

Aquí tienes algunos ejemplos:

```
print(25.3)
produciría esta salida
25.3
```

```
print("Gonzalo")
escribiría
Gonzalo
```

```
$z=3.1416;
print($z);
escribiría
3.1416
```

Recuerda también que es posible utilizar **dentro del paréntesis** el **concatenador de cadenas**.

```
$h=3;
$f="hermanos"
print("Heladería ".$h.$f)
```

que escribiría

Aquí tienes un resumen de las diversas opciones de la función **printf()**:

Sintaxis				
printf( <i>cadena de formato</i> , <i>variable1</i> , <i>variable2</i> ,...)				
Cadena de formato				
"%[relleno][alineación][ancho][precisión][tipo]"				
	Carácter	Valor	Sintaxis	Resultado
<b>Relleno</b>	0	0	printf("%020d",32)	00000000000000000032
	*	*	printf("%*20d",32)	*****32
	espacio <sup>1)</sup>	'	printf("%' 20d",32)	32
	-	'	printf("%'-20d",32)	-----32
	<b>Observaciones</b> En este apartado <b>prestaremos atención únicamente a los caracteres marcados en rojo</b> , que son los que corresponden a las diferentes formas de relleno. Los demás parámetros los iremos tratando uno en los apartados siguientes.  Cuando se pretende <i>rellenar con ceros</i> –a la izquierda– basta escribir el <b>0</b> inmediatamente detrás del signo %  Si se trata de rellenar con un carácter distinto de cero debe escribirse <b>inmediatamente después de % una comilla simple '</b> seguida del carácter de relleno.  Si se pretende rellenar con <b>espacios forzados</b> se puede escribir la comilla simple <b>'e</b> <b>inmediatamente después</b> teclear la combinación <b>ALT+0160</b> (carácter ASCII 160) usando el teclado numérico. Aunque obviamente no se visualiza el espacio si se conserva tal como puede verse en el ejemplo <sup>1)</sup>  Obsérvese que como la tipografía es de <i>ancho variable</i> y que según el carácter que se use como relleno se modifica el ancho de la presentación.  Quizá convenga recordar que <b>32</b> es en este caso la variable a la que pretendemos dar formato y que ese valor podría ser sustituido por el nombre de una variable que contenga valores numéricos.			
	Carácter	Valor	Sintaxis	Resultado
<b>Alineación</b>	Ninguno	Dcha	printf("%020d",32)	00000000000000000032
	-	Izda	printf("%0-20d",32)	32
	Ninguno	Dcha	printf("%*20d",32)	*****32
	-	Izda	printf("%*-20d",32)	32*****
	Ninguno	Dcha	printf("%020s",32)	00000000000000000032
	-	Izda	printf("%0-20s",32)	32000000000000000000
	Ninguno	Dcha	printf("%*20s",32)	*****32
	-	Izda	printf("%*-20s",32)	32*****
	<b>Observaciones</b> En los casos en que figura <i>Ninguno</i> en la columna <i>Carácter</i> tratamos de indicar que no es necesario escribir <i>nada</i> en la cadena de formato. Cuando aparece un signo (-) estamos indicando que debe insertarse un <i>signo menos</i> .  Fíjate que en los cuatro primeros supuestos el identificador de tipo es <b>d</b> , lo cual hace que considere la variable como <b>numérica</b> , mientras que en los cuatro últimos ese valor es <b>s</b> , con lo cual considera la variable como tipo <b>cadena</b> .  Cuando tratamos de rellenar una <b>variable numérica</b> con <b>ceros</b> por la derecha PHP los omite para no alterar el valor numérico en la presentación  Con cualquier otro <b>carácter de relleno</b> (incluidos los caracteres numéricos con ' delante) sí efectúa el relleno.			
	Carácter	Valor	Sintaxis	Resultado
<b>Ancho</b>	Entero	14	printf("%*14d",32)	*****32
	Entero	17	printf("%*-17d",32)	32*****
	Decimal	14.5	printf("%*14.5d",32)	*****32
	Decimal	17.8	printf("%*-17.8d",32)	32*****
	Decimal	14.5	printf("%*14.5f",32)	*****32.00000
	Decimal	11.8	printf("%*-11.8f",32)	32.00000000
	<b>Observaciones</b> El ancho (nº de caracteres totales) puede especificarse mediante un <b>número entero</b> para todo tipo de variables  Si se expresa mediante un <b>número decimal</b> y la variable es tipo <b>coma flotante</b> la parte decimal indica la <b>precisión</b> (nº de cifras decimales) y la parte entera el <b>ancho</b> como número de caracteres de la			

Salidas con formato

Ni la función **echo**, ni tampoco **print** permiten establecer una *presentación* (formato) en sus salidas, excepto que alguna de las variables que se use contenga el resultado de una función **number\_format**.

La función **printf()** ofrece un gran número de posibilidades en este sentido. Tanto la sintaxis como los valores de los diferentes parámetros –cuando se trate de presentar números– las tienes resumidas aquí a la derecha.

En la página siguiente veremos el uso de **printf()** para el tratamiento de variables tipo *cadena*.

parte entera o de la parte decimal, según se rellene a la derecha o a la izquierda.				
Tipo	Tipo	Valor	Sintaxis	Resultado
	Presentación en forma binaria	<b>b</b>	printf("%*14 <b>b</b> ",17)	*****10001
	Caracter correspondiente al código ASCII	<b>c</b>	printf("%*14 <b>c</b> ",97)	<b>a</b>
	Número presentado como entero	<b>d</b>	printf("%*14 <b>d</b> ",17.83)	*****17
	Número presentado con decimales	<b>f</b>	printf("%*14 <b>f</b> ",17.45)	*****17.450000
	Presentación en forma octal	<b>o</b>	printf("%*14 <b>o</b> ",17)	*****21
	Presentación en hexadecimal	<b>x</b>	printf("%*14 <b>x</b> ",170)	*****aa
	Presentación en hexadecimal	<b>X</b>	printf("%*14 <b>X</b> ",170)	*****AA
	Presentación como >cadena	<b>s</b>	printf("%*14 <b>s</b> ",170)	*****170

Anterior



Índice



Siguiente

