

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				
<code>printf(cadena de formato,variable1,variable2,..)</code>				
Cadena de formato				
<code>"%[relleno][alineación][ancho][precisión][tipo]"</code>				
Relleno	Carácter	Valor	Sintaxis	Resultado
		0	<code>printf("%020d",32)</code>	00000000000000000032
		*	<code>printf("%*20d",32)</code>	*****32
		espacio ¹⁾	<code>printf("% 20d",32)</code>	32
		-	<code>printf("%-20d",32)</code>	-----32
Observaciones				
En este apartado prestaremos atención únicamente a los caracteres marcados en rojo , 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 0 inmediatamente detrás del signo %				
Si se trata de rellenar con un carácter distinto de cero debe escribirse inmediatamente después de % una comilla simple ' seguida del carácter de relleno.				
Si se pretende rellenar con espacios forzados se puede escribir la comilla simple 'e inmediatamente después teclear la combinación ALT+0160 (carácter ASCII 160) usando el teclado numérico. Aunque obviamente no se visualiza el espacio si se conserva tal como puede verse en el ejemplo ¹⁾				
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 32 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.				
Alineación	Carácter	Valor	Sintaxis	Resultado
	Ninguno	Dcha	<code>printf("%020d",32)</code>	00000000000000000032
	-	lzda	<code>printf("%0-20d",32)</code>	32
	Ninguno	Dcha	<code>printf("%*20d",32)</code>	*****32
	-	lzda	<code>printf("%*-20d",32)</code>	32*****
	Ninguno	Dcha	<code>printf("%020s",32)</code>	00000000000000000032
	-	lzda	<code>printf("%0-20s",32)</code>	3200000000000000000
	Ninguno	Dcha	<code>printf("%*20s",32)</code>	*****32
	-	lzda	<code>printf("%*-20s",32)</code>	32*****
	Observaciones			
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 d , lo cual hace que considere la variable como numérica , mientras que en los cuatro últimos ese valor es s , con lo cual considera la variable como tipo cadena .				
Cuando tratamos de rellenar una variable numérica con ceros por la derecha PHP los omite para no alterar el valor numérico en la presentación				
Con cualquier otro carácter de relleno (incluidos los caracteres numéricos con ' delante) sí efectúa el relleno.				
Ancho	Carácter	Valor	Sintaxis	Resultado
	Entero	14	<code>printf("%*14d",32)</code>	*****32
	Entero	17	<code>printf("%*-17d",32)</code>	32*****
	Decimal	14.5	<code>printf("%*14.5d",32)</code>	*****32
	Decimal	17.8	<code>printf("%*-17.8d",32)</code>	32*****
	Decimal	14.5	<code>printf("%*14.5f",32)</code>	*****32.00000
	Decimal	11.8	<code>printf("%*-11.8f",32)</code>	32.00000000
Observaciones				
El ancho (nº de caracteres totales) puede especificarse mediante un número entero para todo tipo de variables				
Si se expresa mediante un número decimal y la variable es tipo coma flotante la parte decimal indica la precisión (nº de cifras decimales) y la parte entera el ancho 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	<code>printf("%*14b",17)</code>	*****10001
	Caracter correspondiente al código ASCII	c	<code>printf("%*14c",97)</code>	a
	Número presentado como entero	d	<code>printf("%*14d",17.83)</code>	*****17
	Número presentado con decimales	f	<code>printf("%*14f",17.45)</code>	*****17.450000
	Presentación en forma octal	o	<code>printf("%*14o",17)</code>	*****21
	Presentación en hexadecimal	x	<code>printf("%*14x",170)</code>	*****aa
	Presentación en hexadecimal	X	<code>printf("%*14X",170)</code>	*****AA
Presentación como >cadena	s	<code>printf("%*14s",170)</code>	*****170	

Anterior



Índice



Siguiente

