

## Formatos de las variables numéricas

Para asignar valores numéricos a una variable, en PHP, puede utilizarse uno de los siguientes sistemas de numeración:

Los **números enteros** pueden escribirse en una cualquiera de estas bases:

### • Base decimal

#### \$a=número

No se pueden insertar *ceros a la izquierda* cuando se escriben números en base decimal.

### • Base octal

#### \$a=0número octal

Basta poner un CERO delante del número para que sea interpretado como escrito en base OCTAL. Obviamente, sólo admite los dígitos de 0 a 7.

### • Base hexadecimal

#### \$a=0xnúmero hexadecimal

Si se escribe CERO EQUIS (0x) delante del número, PHP lo interpretará como expresado en hexadecimal. En este caso, admitirá como dígitos de 0 a 9 y de A a F.

Un número de **coma flotante** puede escribirse de cualquiera de estas formas:

### • Notación decimal.

#### \$a=número

Se pueden utilizar un cero a la izquierda del punto decimal.

### • Notación científica

#### \$a=número e *exponente*

Se puede utilizar un cero a la izquierda del punto decimal

Ejemplo: **\$a=1.2e5** asigna a **\$a** el valor: **1.2 x 10<sup>5</sup>**

Otro ej: **\$a=1.2e-5** asigna a **\$a** el valor: **1.2 x 10<sup>-5</sup>**

## Cambios de base

PHP permite hacer todo tipo de **cambios de base**. Para evitar ser reiterativos, observa los ejemplos. Allí tienes las diferentes funciones mediante las que se puede realizar ese proceso.

## Presentaciones numéricas

La *presentación* de los valores numéricos permite una gran variedad de formatos.

El número de cifras decimales, los separadores de decimales y los separadores de mil pueden configurarse a voluntad.

Los ejemplos de aplicación y la sintaxis son los que tienes en la parte derecha de esta página.

## Formas de asignar valores a la variables

Cuando se asignan valores numéricos a una variable PHP cabe la posibilidad de hacerlo en distintas bases. Estos son algunos ejemplos.

Asignación de valores en distintas bases			
Base	Sintaxis	Valor decimal	Aplicable a
Base Decimal	<b>\$a=17</b>	17	Números enteros
Base Octal	<b>\$a=017</b>	15	Números enteros
Base Hexadecimal	<b>\$a=0x17</b>	23	Números enteros
Base Hexadecimal	<b>\$a=0x1A3B</b>	6715	Números enteros
Notación decimal	<b>\$a=123000;</b>	123000	Coma flotante
Base Decimal	<b>\$a=0.174</b>	0.174	Coma flotante
Notación científica	<b>\$a=1.23e5;</b>	123000	Coma flotante
Notación científica	<b>\$a=23.4e-2;</b>	0.234	Coma flotante

## Cambios de base

PHP dispone de funciones que permiten obtener **una cadena de caracteres** con la expresión, en una **nueva base**, de un número escrito en **otra base cualquiera**.

Asignación de valores en distintas bases				
Valor de la variable	Base	Nueva base	Sintaxis	Expresión
\$a=1234	10	8	<b>decoct(\$a)</b>	2322
\$a=1234	10	16	<b>dechex(\$a)</b>	4d2
\$a=1234	10	2	<b>decbin(\$a)</b>	10011010010
\$a=1234	8	10	<b>octdec(\$a)</b>	668
\$a=1234	16	10	<b>hexdec(\$a)</b>	4660
\$a=1010011	2	10	<b>bindec(\$a)</b>	83
\$a=1234	7	14	<b>base_convert(\$a,7,14)</b>	254
\$a=1234	5	18	<b>base_convert(\$a,5,18)</b>	ae
\$a=1234	18	5	<b>base_convert(\$a,18,5)</b>	202123

## Formato de presentación de números

En PHP es posible establecer el formato de la presentación de los valores numéricos utilizando alguna de estas funciones.

### number\_format (número)

Presenta la parte entera del número (sin decimales) y utiliza como separador de miles una coma (,).

### number\_format (número , número de cifras decimales)

Presenta el número de cifras decimales que se indiquen y utiliza como separador decimal un punto (.) y el separador de miles es una coma (,).

### number\_format (número , núm decimales , "sep. decimal" , "sep. miles")

Permite establecer el número de cifras decimales de la presentación así como el carácter que se establezca como separador de decimales y como separadores de miles.

**¡Cuidado!** No te olvides de escribir los caracteres de separación *entre comillas*.

Aquí tienes algunos ejemplos.

Formatos de presentación de número					
Valor inicial	Nº de decimales	Sep. dec.	Sep. miles	Sintaxis	Resultado
\$a=1234567.234	0		,	<b>number_format(\$a)</b>	1,234,567
\$a=1234567.234	2	.	,	<b>number_format(\$a,2)</b>	1,234,567.23
\$a=1234567.234	1	,	.	<b>number_format(\$a,2,"," ".")</b>	1.234.567,2
\$a=1234567.234	1	'	esp	<b>number_format(\$a,2,"' "," ")</b>	1 234 567'2

### Ejercicio nº 12

Modifica el **ejercicio11.php** de forma que los resultados obtenidos al realizar los cálculos aparezcan con un espacio como separador de miles, un punto como separador de decimales y cuatro cifras decimales.