

Funciones de cadenas

Algunas de las funciones que permiten manejar los formatos de las cadenas de caracteres son estas:

chr(n)

Devuelve el carácter cuyo código ASCII es n.

ord(cadena)

Devuelve el código ASCII del primero de los caracteres de la cadena.

strlen(cadena)

Devuelve la longitud (número de caracteres) de la cadena. Los espacios son considerados como un carácter más.

strtolower(cadena)

Cambia todos los caracteres de la cadena a minúsculas.

strtoupper(cadena)

Convierte en mayúsculas todos los caracteres de la cadena.

ucwords(cadena)

Convierte a mayúsculas la primera letra de cada palabra.

ucfirst(cadena)

Convierte a mayúsculas la primera letra de la cadena y pone en minúsculas todas las demás.

ltrim(cadena)

Elimina todos los espacios que pudiera haber al principio de la cadena.

rtrim(cadena)

Elimina todos los espacios que existieran al final de la cadena.

trim(cadena)

Elimina los espacios tanto al principio como al final de la cadena

chop(cadena)

Elimina los espacios al final de la cadena. Es idéntica a rtrim.

Advertencia

Formatos en cadenas

printf(cadena de formato,variable1,variable2,..)
Cadena de formato
Dentro de la cadena de formatos deben repetirse tantos formatos como variables se pretenda manejar
"%[rell1][alin1][anc1][prec1][tipo1][sepa1]%"[rell1][alin1][anc1][prec1][tipo1][sepa1]"
Hemos de mencionar aquí los separadores ya que no fueron mencionados en la página anterior
Se puede introducir una cadena de separación al final de una cadena de formato que puede hacer, entre otras, función de separación entre dos cadenas. Por ejemplo, printf("%*15.2f Euros",1475.875) nos devolvería:
*****1475.88 Euros
La función printf() permite presentar varios valores o variables con distintos formatos utilizando la sintaxis que se indica más arriba.
Este ejemplo :
printf("%*15.2f Euros=%*18.0f Pesetas",1475.875,1475.875*166.386)
devuelve como resultado:
*****1475.88 Euros=*****245565 Pesetas

Existe otra función PHP con características muy similares a la anterior. Se trata de **sprintf()**.

La sintaxis es idéntica **sprintf (cadena de formato, variable1,variable2, ...)** y su única diferencia con **printf** es que, mientras que **printf()** imprime las variables utilizando el formato indicado, **sprintf()** puede guardar en una nueva variable la cadena resultante de la aplicación del formato.

Otras funciones con cadenas

Estos son algunos ejemplos de aplicación de las funciones de manejo de cadenas

Código ASCII y viceversa		
Función	Ejemplo	Resultado
chr(código ASCII)	chr(97)	a
ord("cadena")	ord("abadesa")	97
Longitudes y conversiones mayúsculas/minúsculas		
Función	Ejemplo	Resultado
strlen("cadena")	strlen("Mide la longitud de esta cadena")	31
strtolower("cadena")	strtolower("CONVIERTE A MINÚSCULAS")	convierte a minúsculas
strtoupper("cadena")	strtoupper("pasa a mayúsculas")	PASA A MAYÚSCULAS
ucwords("cadena")	ucwords("todas empiezan por mayúscula")	Todas Empiezan Por Mayúscula
ucfirst("cadena")	ucfirst("mayúscula al principio")	Mayúscula al principio
Eliminar espacios		

Tanto **trim**, como **ltrim** y **rtrim** eliminan, además de los espacios, las secuencias: `\n`, `\r`, `\t`, `\v` y `\0`; llamadas también **caracteres protegidos**.

substr(cadena,n)

Si el valor de **n** es **positivo** extrae **todos los caracteres** de la cadena a partir del que ocupa la posición **enésima** a contar desde la izquierda.

Si el valor de **n** es **negativo** serán extraídos **los n últimos caracteres** contenidos en la cadena.

substr(cadena,n,m)

Si **n** y **m** son **positivos** extrae **m** caracteres a partir del que ocupa la posición **enésima**, de izquierda a derecha.

Si **n** es **negativo** y **m** es **positivo** extrae **m** (contados de izquierda a derecha) a partir del que ocupa la posición **enésima** contada de derecha a izquierda.

Si **n** es **positivo** y **m** es **negativo** extrae la cadena comprendida entre el **enésimo** carácter (contados de izquierda a derecha) hasta el **emésimo**, contando en este caso de derecha a izquierda.

Si **n** es **negativo** y **m** también es **negativo** extrae la porción de cadena comprendida entre el **emésimo** y el **enésimo** caracteres contando, en ambos casos, de derecha a izquierda. Si el valor absoluto de **n** es **menor** que el de **m** devuelve una **cadena vacía**.

strrev(cadena)

Devuelve la cadena **invertida**

str_repeat(cadena, n)

Devuelve la cadena **repetida** tantas veces como indica **n**.

str_pad(cad, n, rell, tipo)

Añade a la cadena **cad** los caracteres especificados en **rell** (uno o varios, escritos entre comillas) hasta que alcance la longitud que indica **n** (un número)

El parámetro **tipo** puede tomar uno de estos tres valores (sin comillas):
STR_PAD_BOTH (rellena por ambos lados)
STR_PAD_RIGHT (rellena por la derecha)
STR_PAD_LEFT (rellena por la izquierda).

Si se omite la cadena de **Relleno** utilizará **espacios** y si se omite el **tipo** rellenará por la **derecha**

Función	Ejemplo	Resultado
ltrim("cadena")	ltrim("\n\nEliminar espacios")	Eliminar espacios
rtrim("cadena")	rtrim("Eliminar espacios\n\n")	Eliminar espacios
trim("cadena")	trim("\n\nEliminar espacios\n\n")	Eliminar espacios
chop("cadena")	chop("\n\nEliminar espacios\n\n")	Eliminar espacios
Extraer porciones de una cadena		
Función	Ejemplo	Resultado
substr("cadena",n)	substr("Extrae caracteres",3)	rae caracteres
substr("cadena",n)	substr("Extrae caracteres",0)	Extrae caracteres
substr("cadena",n)	substr("Extrae caracteres",-5)	teres
substr("cadena",n)	substr("Extrae caracteres",-2)	es
substr("cadena",n,m)	substr("Extrae caracteres",2,6)	trae c
substr("cadena",n,m)	substr("Extrae caracteres",0,8)	Extrae c
substr("cadena",n,m)	substr("Extrae caracteres",2,-3)	trae caracte
substr("cadena",n,m)	substr("Extrae caracteres",-7,5)	acter
substr("cadena",n,m)	substr("Extrae caracteres",-7,-5)	ac
substr("cadena",n,m)	substr("Extrae caracteres",-5,-7)	
Modificaciones de cadenas		
Función	Ejemplo	Resultado
strrev("cadena")	strrev("Invierte la cadena")	anedac al etreivnl
str_repeat("cadena",n)	str_repeat("Rep",5)	RepRepRepRepRep
str_pad("cadena",n,"Relleno",Tipo)	str_pad("Pepe",10,"*",STR_PAD_BOTH)	***Pepe***
str_pad("cadena",n,"Relleno",Tipo)	str_pad("Pepe",10,"*",STR_PAD_LEFT)	*****Pepe
str_pad("cadena",n,"Relleno",Tipo)	str_pad("Pepe",10,"*",STR_PAD_RIGHT)	Pepe*****
str_pad("cadena",n,"Relleno",Tipo)	str_pad("Pepe",10,"*")	Pepe*****
str_replace ("lo que dice",lo que dira,"Cadena")	str_replace("e","a","Pepe")	Papa
str_replace ("lo que dice",lo que dira,"Cadena")	str_replace("pe","pa","Pepepe")	Pepapa
str_replace ("lo que dice",lo que dira,"Cadena")	str_replace("Pepe","Luis","Pepe")	Luis
substr_replace ("Cadena",lo que dira,n,m)	substr_replace("Pepe","Luis",2,-1)	PeLuisse

¡Cuidado!

Como tu *buen criterio* ya habrá podido advertir, no se trata aquí de *aprender* todas estas opciones de formato –ni las que veremos en las páginas siguientes– sino de que dispongas de una referencia de consulta a la que recurrir en el momento en el que necesites utilizar estas funciones.

Pensamos que **sí** es importante que sepas que PHP dispone de todos estos recursos y conozcas su potencialidad para hacer presentaciones de las formas más variadas.

Ejercicio nº 15

Crea un formulario **-formulario15.php-** mediante el que puedas transferir un valor numérico y un texto (un input tipo *text* y otro *textarea*).

Trata de que el script que los reciba **-visor15.php-**, permita visualizar sus valores de forma que el valor numérico tenga 2 decimales y acabe con la palabra euros, y que la parte entera se complete (por la izquierda) con asteriscos hasta tener una longitud de 12 caracteres.

El contenido del *textarea* debería visualizarse íntegramente en minúsculas, excepto los diez primeros caracteres que deberían verse en mayúsculas.

Anterior

Índice

Siguiente

