

# Algunas funciones de PHP

En este capítulo no van a incluirse todas y cada una de las funciones que soportan PHP y sus módulos. Se trata de crear una guía general de las funciones más comunes del lenguaje.

## Funciones básicas

### echo y print

A lo largo de todo el texto hemos utilizado las funciones *echo* y *print*, sin preocuparnos de explicar su significado y sintaxis. Dichas funciones, como ya habrá supuesto el lector, imprimen en la salida estándar el argumento que se les pasa. Hay que hacer notar que su sintaxis es muy flexible, siendo válidas las cuatro siguientes llamadas:

```
$valor = 52;
echo "El valor es $valor";
echo ("El valor es $valor");
print "El valor es $valor";
print ("El valor es $valor");
```

## Funciones de vectores

Ya hemos visto con anterioridad el manejo de vectores así como su inicialización. En este apartado vamos a estudiar una serie de funciones que van a simplificar el manejo de vectores. Además, prestaremos especial atención a una función de inicialización que aparece en esta sección.

### array()

Esta función devuelve un vector que forma en base a los parámetros que se le pasan. A los parámetros se les puede dar un índice asociativo por medio del símbolo `=>`. Vamos a ver un ejemplo donde creamos un vector de dos dimensiones:

```
$varios = array(
    "colores" => array("a"=>"amarillo", "b"=>"rojo"),
    "sentidos" => array("tacto", "gusto", "olfato", "oído", "vista")
);
```

### El resto de funciones

nombre	parámetros	descripción
<b>array_count_values()</b>	vector A	Devuelve un vector usando los valores del vector de entrada como índices asociativos y su frecuencia como valor
<b>array_flip()</b>	vector A	Devuelve el vector inverso a A
<b>array_keys()</b>	vector A	Devuelve un vector con todos los valores de clave de A
<b>array_merge()</b>	vectores A; B, ...	Devuelve el vector concatenación de los vectores A, B, ...
<b>array_pop()</b>	vector V	Devuelve y extrae el valor del último elemento de V
<b>array_push</b>	vector V; gen A, B, ...	Añade al final de V los elementos A, B, ... devolviendo el número de elementos añadidos
<b>array_reverse()</b>	vector V	Devuelve un vector con los elementos de V en orden inverso

<b>array_shift</b>	vector V	Devuelve y extrae el valor del primer elemento de V
<b>array_slice()</b>	vector V; ent A, B	Extrae de un vector los elementos que van desde la posición A hasta A+B, devolviendo un nuevo vector con dichos elementos
<b>array_splice()</b>	vector V; ent A, B; vector W	Elimina los elementos de V desde A hasta A+B e inserta en su lugar los elementos del vector W
<b>array_unshift()</b>	vector V; gen A, B, ...	Inserta los elementos A, B, ... al principio de V y devuelve la cantidad de elementos insertados
<b>array_values()</b>	vector V	Devuelve un vector con todos los valores de V
<b>array_walk()</b>	vector V; cad f; gen A, B, ...	Aplica la función f a todos los elementos del vector V añadiendo los parámetros A, B, ... en cada llamada
<b>arsort()</b>	vector V	Ordena un vector de forma inversa manteniendo la asociación clave-valor
<b>asort()</b>	vector V	Ordena un vector manteniendo la asociación clave-valor
<b>count()</b>	gen V	Devuelve el número de elementos de una variable
<b>current()</b>	vector V	Devuelve el elemento actual de V
<b>each()</b>	vector V	Devuelve la próxima pareja clave-valor de V
<b>end()</b>	vector V	Lleva el puntero interno del vector a su último elemento
<b>in_array()</b>	gen A; vector V	Devuelve verdadero si A se encuentra en V; Falso en otro caso
<b>krsort()</b>	vector V	Ordena V de forma inversa en base a la clave
<b>ksort()</b>	vector V	Ordena V en base a la clave
<b>list()</b>	gen A, B...	Asigna variables como si fueran un vector
<b>next()</b>	vector V	Avanza el puntero interno de un vector a la siguiente posición
<b>pos()</b>	vector V	Devuelve el elemento señalado por el puntero interno de V
<b>prev()</b>	vector V	Retrocede el puntero interno de un vector V a la posición anterior
<b>reset()</b>	vector V	Coloca el puntero interno de V en la primera posición
<b>rsort()</b>	vector V	Ordena V de forma inversa
<b>shuffle()</b>	vector V	"Desordena" los elementos de V
<b>sizeof()</b>	vector V	Devuelve el número de elementos de V
<b>sort</b>	vector V	Ordena el vector V
<b>uasort()</b>	vector V	Ordena un vector en base a una función de comparación definida por el usuario y manteniendo la asociación clave-valor
<b>uksort()</b>	vector V	Ordena un vector en base a una función de comparación definida por el usuario sobre las claves
<b>usort()</b>	vector V	Ordena un vector en base a una función de comparación definida por el usuario sobre los valores

## Funciones de cadenas

PHP ofrece multitud de funciones para el manejo de cadenas. En esta sección mostramos sólo las más utilizadas.

nombre	parámetros	descripción
<b>print()</b>	cad A	Imprime A en la salida estándar

<b>printf()</b>	cad A	Imprime A en la salida estándar aplicándole un formato (estilo C)
<b>sprint()</b>	cad F; gen A, B, ...	Devuelve una cadena formateada según F con las variable A, B, ...
<b>strstr() / strchr()</b>	cad A, B	Busca B dentro de A y devuelve desde su primera aparición hasta el final de A.
<b>strcmp</b>	cad A, B	Devuelve un valor menor que 0 si A es menor que B; mayor que 1 si B es menor que A; y 0 si son idénticas
<b>stristr()</b>	cad A, B	Como strstr pero insensible a mayúsculas/minúsculas
<b>strlen()</b>	cad A	Devuelve la longitud de A
<b>strpos()</b>	cad A, B	Devuelve la posición de la primera aparición de B en A
<b>strrchr()</b>	cad A, B	Análoga a strchr() pero empezando por el final de la cadena
<b>strrev()</b>	cad A	Invierte una cadena
<b>strrpos()</b>	cad A, B	Análoga a strpos() pero empezando por el final de la cadena
<b>strrchr()</b>	cad A, B	Análoga a strchr() pero empezando por el final de la cadena
<b>strtok()</b>	cad A, B	Extrae un token de A en base a B
<b>strtolower() / strtoupper()</b>	cad A	Pone una cadena en minúsculas / mayúsculas
<b>str_replace()</b>	cad A, B, C	Reemplaza por B las apariciones de A en C
<b>substr()</b>	cad A; ent B, C	Devuelve la subcadena de A que va desde la posición B hasta la B+C.
<b>substr_replace()</b>	cad A, B; ent C, D	Sustituye en A la subcadena desde C hasta C+D por la subcadena B

## Funciones de variables

nombre	parámetros	descripción
<b>doubleval()</b>	esc A	Devuelve el valor <i>double</i> de A
<b>empty()</b>	gen A	Devuelve falso si la variable tiene valor vacío o cero
<b>gettype()</b>	gen A	Devuelve el tipo de la variable en forma de cadena: "integer", "double", "string", "array", "object", "unknown type"
<b>intval()</b>	est A; ent B	Devuelve el valor entero de A usando B como base numérica. Por defecto, B = 10
<b>is_array()</b>	gen A	Devuelve verdadero si A es un vector.
<b>is_float() / is_double() / is_real()</b>	gen A	Devuelve verdadero si A es un real.
<b>is_int() / is_integer() / is_long()</b>	gen A	Devuelve verdadero si A es un entero.
<b>is_string()</b>	gen A	Devuelve verdadero si A es una cadena.
<b>is_object()</b>	gen A	Devuelve verdadero si A es un objeto.
<b>isset()</b>	gen A	Devuelve verdadero si existe A
<b>settype()</b>	gen A; cad B	Fuerza la variable A al tipo B
<b>strval()</b>	gen A	Convierte el valor de A a una cadena

<code>unset()</code>	gen A	Destruye A
----------------------	-------	------------