

Funciones secretas para trabajar con arrays PHP

Las principales funciones para trabajar con arrays en PHP son:

Función	Descripción	Ejemplo
print_r(\$array)	Muestra todos los elementos de una array por pantalla	<pre>\$array = array(3, 1,2, 6,1); print_r(\$array);</pre>
asort(mi_array) y arsort(mi_array)	Ordena por orden alfabético directo o inverso en función de los valores.	<pre>\$array = array(3, 1,2, 6,1); sort(\$array); print_r(\$array);</pre>
count(mi_array)	Nos da el número de elementos de nuestro array.	<pre>\$array = array(3, 1,2, 6,1); count(\$array);</pre>
list (\$variable1, \$variable2...)=mi_array	Asigna una variable a cada uno de los valores del array.	<pre>\$array = array(3, 1,2, 6,1); list(\$a,\$b,\$c,\$d,\$e)=\$array;</pre>
next(mi_array), prev(mi_array), reset(mi_array) y end(mi_array)	Nos permiten movernos por dentro del array con un puntero hacia delante, atrás y al principio y al final.	<pre>\$array = array(3, 1,2, 6,1); end(\$array);</pre>
array_push(array,"variable")	Inserta un elemento al final del array.	<pre>\$array = array(3, 1,2, 6,1); array_push(\$array,9); print_r(\$array);</pre>
array_unshift(array,"variable")	Inserta un elemento al principio del array.	<pre>\$array = array(3, 1,2, 6,1); array_unshift(\$array,9); print_r(\$array);</pre>
array_slice(array, primer elemento lista, numero elementos)	Para disminuir el número de casillas de un arreglo	<pre>\$array = array(3, 1,2, 6,1); array_slice(\$array,1,3); print_r(\$array);</pre>
array_merge(array1,array2)	Esta función une dos o más arrays que pasen por parámetro y devuelve un array con todos los elementos	<pre>\$array1 = array(3, 1,2, 6,1); \$array2 = array(1,2); \$array3=array_merge(\$array1,\$array2); print_r(\$array3);</pre>
array_unique(array)	Elimina valores duplicados de un array	<pre>\$array = array(3, 1,2, 6,1); array_unique(\$array); print_r(\$array);</pre>
in_array(valor, array)	Comprueba si un valor existe en un array y devuelve verdadero o falso.	<pre>\$array = array(3, 1,2, 6,1); echo in_array(3,\$array);</pre>
\$str=implode(\$array)	Convierte array a cadena de texto	<pre>\$array = array('a','b','c','d','e'); \$str=implode(\$array);</pre>

Existen muchas más funciones para trabajar con arrays, las podemos encontrar en <http://php.net/manual/es/book.array.php>

- [array_change_key_case](#) — Cambia a mayúsculas o minúsculas todas las claves en un array
- [array_chunk](#) — Divide un array en fragmentos
- [array_column](#) — Devuelve los valores de una sola columna del array de entrada
- [array_combine](#) — Crea un nuevo array, usando una matriz para las claves y otra para sus valores
- [array_count_values](#) — Cuenta todos los valores de un array
- [array_diff_assoc](#) — Calcula la diferencia entre arrays con un chequeo adicional de índices
- [array_diff_key](#) — Calcula la diferencia entre arrays empleando las claves para la comparación
- [array_diff_uassoc](#) — Calcula la diferencia entre arrays con un chequeo adicional de índices que se realiza por una función de devolución de llamada suministrada por el usuario
- [array_diff_ukey](#) — Calcula la diferencia entre arrays usando una función de devolución de llamada en las keys para comparación
- [array_diff](#) — Calcula la diferencia entre arrays
- [array_fill_keys](#) — Llena un array con valores, especificando las keys
- [array_fill](#) — Llena un array con valores
- [array_filter](#) — Filtra elementos de un array usando una función de devolución de llamada
- [array_flip](#) — Intercambia todas las claves de un array con sus valores asociados
- [array_intersect_assoc](#) — Calcula la intersección de arrays con un chequeo adicional de índices
- [array_intersect_key](#) — Calcula la intersección de arrays usando sus claves para la comparación
- [array_intersect_uassoc](#) — Calcula la intersección de arrays con una comprobación adicional de índices, los cuales se comparan con una función de retrollamada
- [array_intersect_ukey](#) — Calcula la intersección de arrays usando una función de devolución de llamada en las claves para la comparación
- [array_intersect](#) — Calcula la intersección de arrays
- [array_key_exists](#) — Verifica si el índice o clave dada existe en el array
- [array_keys](#) — Devuelve todas las claves de un array o un subconjunto de claves de un array
- [array_map](#) — Aplica la retrollamada a los elementos de los arrays dados
- [array_merge_recursive](#) — Une dos o más arrays recursivamente
- [array_merge](#) — Combina dos o más arrays
- [array_multisort](#) — Ordena varios arrays, o arrays multidimensionales
- [array_pad](#) — Rellena un array a la longitud especificada con un valor
- [array_pop](#) — Extrae el último elemento del final del array
- [array_product](#) — Calcula el producto de los valores de un array
- [array_push](#) — Inserta uno o más elementos al final de un array
- [array_rand](#) — Seleccionar una o más entradas aleatorias de un array
- [array_reduce](#) — Reduce iterativamente un array a un solo valor usando una función llamada de retorno
- [array_replace_recursive](#) — Reemplaza los elementos de los arrays pasados al primer array de forma recursiva

- [array_replace](#) — Reemplaza los elementos de los arrays pasados en el primer array
- [array_reverse](#) — Devuelve un array con los elementos en orden inverso
- [array_search](#) — Busca un valor determinado en un array y devuelve la primera clave correspondiente en caso de éxito
- [array_shift](#) — Quita un elemento del principio del array
- [array_slice](#) — Extraer una parte de un array
- [array_splice](#) — Elimina una porción del array y la reemplaza con otra cosa
- [array_sum](#) — Calcular la suma de los valores de un array
- [array_udiff_assoc](#) — Computa la diferencia entre arrays con una comprobación de índices adicional, compara la información mediante una función de llamada de retorno
- [array_udiff_uassoc](#) — Computa la diferencia entre arrays con una verificación de índices adicional, compara la información y los índices mediante una función de llamada de retorno
- [array_udiff](#) — Computa la diferencia entre arrays, usando una llamada de retorno para la comparación de datos
- [array_uintersect_assoc](#) — Calcula la intersección de arrays con una comprobación de índices adicional, compara la información mediante una función de retrollamada
- [array_uintersect_uassoc](#) — Calcula la intersección de arrays con una comprobación de índices adicional, compara la información y los índices mediante funciones de retrollamada por separado
- [array_uintersect](#) — Computa una intersección de arrays, compara la información mediante una función de llamada de retorno
- [array_unique](#) — Elimina valores duplicados de un array
- [array_unshift](#) — Añadir al inicio de un array uno a más elementos
- [array_values](#) — Devuelve todos los valores de un array
- [array_walk_recursive](#) — Aplicar una función de usuario recursivamente a cada miembro de un array
- [array_walk](#) — Aplicar una función proporcionada por el usuario a cada miembro de un array
- [array](#) — Crea un array
- [arsort](#) — Ordena un array en orden inverso y mantiene la asociación de índices
- [asort](#) — Ordena un array y mantiene la asociación de índices
- [compact](#) — Crear un array que contiene variables y sus valores
- [count](#) — Cuenta todos los elementos de un array o algo de un objeto
- [current](#) — Devuelve el elemento actual en un array
- [each](#) — Devolver el par clave/valor actual de un array y avanzar el cursor del array
- [end](#) — Establece el puntero interno de un array a su último elemento
- [extract](#) — Importar variables a la tabla de símbolos actual desde un array
- [in_array](#) — Comprueba si un valor existe en un array
- [key_exists](#) — Alias de array_key_exists
- [key](#) — Obtiene una clave de un array
- [krsort](#) — Ordena un array por clave en orden inverso
- [ksort](#) — Ordena un array por clave
- [list](#) — Asignar variables como si fueran un array
- [natcasesort](#) — Ordenar un array usando un algoritmo de "orden natural" insensible a mayúsculas-minúsculas

- [natsort](#) — Ordena un array usando un algoritmo de "orden natural"
- [next](#) — Avanza el puntero interno de un array
- [pos](#) — Alias de current
- [prev](#) — Rebobina el puntero interno del array
- [range](#) — Crear un array que contiene un rango de elementos
- [reset](#) — Establece el puntero interno de un array a su primer elemento
- [rsort](#) — Ordena un array en orden inverso
- [shuffle](#) — Mezcla un array
- [sizeof](#) — Alias de count
- [sort](#) — Ordena un array
- [uasort](#) — Ordena un array con una función de comparación definida por el usuario y mantiene la asociación de índices
- [uksort](#) — Ordena un array según sus claves usando una función de comparación definida por el usuario
- [usort](#) — Ordena un array según sus valores usando una función de comparación definida por el usuario