Índice

Comparar arrays 2
array_diff 2
array_diff_assoc 2
Unir arrays 2
array_merge 2
array_merge_recursive 2
array_combine 2
Rellenar arrays 3
array_fill 3
range 3
Dividir arrays 3
array_slice 3
array_splice 3
array_chunk 4
Modificar elementos en arrays 4
array_shift 4
array_unshift 4
array_pop 4
array_push 4
array_flip 4
Contar elementos de un array 5
Ordenar un array 5
Buscar un valor en un array 6
Reindexar un array 6
Barajar los elementos de un array 6
Extraer al azar un elemento de un array 6
Eliminar valores repetidos 6

Comparar arrays

array_diff

array_diff (array $array1, array $array2 [, array $...]): array
Compara **$array1** con uno o más arrays y devuelve los valores de **$array1** que no estén presentes en cualquiera de los demás arrays.

array_diff_assoc

array_diff_assoc (array $array1, array $array2 [, array $...]): array
Compara **$array1** con **$array2** y devuelve la diferencia, pero esta vez comparando valores y claves.

array_diff_key

array_diff_key (array $array1, array $array2 [, array $...]): array
Compara las claves de **$array1** con las claves de **$array2** y devuelve la diferencia. Es como la función array_diff(), pero sólo comparando las claves en lugar de los valores.

Unir arrays

array_merge

array_merge (array $array1 [, array $...]): array
Combina dos o más arrays, de modo que los valores de uno se unen al final del anterior. Devuelve el array resultante. Al contrario que con el operador suma, si los arrays de entrada tienen las mismas claves de tipo string, el último valor para esa clave sobreescribirá al anterior. Pero si son arrays numéricas, el último valor no sobreescribirá al valor original, sino que se añadirá al final.

array_merge_recursive

array_merge_recursive (array $array1 [, array $...]): array
Es igual que **array_merge()**, salvo que cuando los arrays de entrada tienen las mismas claves de tipo string, los valores de estas claves se unen en un array de forma recursiva, de forma que si uno de los valores es un array, la función unirá también ésta con la correspondiente entrada de otro array. Si los arrays tienen la misma clave numérica, el valor posterior no sobreescribirá el valor original, sino que se añadirá al final.

array_combine

array_combine (array $keys, array $values): array
Crea un array utilizando un array para sus claves y otro array para sus valores. Coge los valores para crear el nuevo array (clave => valor) de los arrays de origen.

Rellenar arrays

array_fill

array_fill (int $start_index, int $num, mixed $value): array
Llena un array $num veces (debe ser mayor que cero) con el valor **$value** desde la clave **$startindex**. Si **$startindex** es negativo, el primer valor tendrá como clave ese valor negativo y los demás comenzarán desde cero.

range

range (mixed $start, mixed $end [, number $step = 1]): array
Esta función genera un array de valores en sucesión aritmética, es decir los valores desde **$start** hasta **$end** contando de paso en paso **$step** (sin superar el valor $final).

Dividir arrays

array_slice

array_slice (array $array, int $offset [, int $length = null [, bool $preserve_keys = false]]): array
Extrae una parte de un array, que comienza en **$offset** (si es negativo comienza por el final) y con una longitud de **$length**.

Si **$length** es positivo, la secuencia tendrá tantos elementos como indique el valor. Si es mayor que el propio array, se devolverán los elementos disponibles en el array. Si es negativo, la secuencia terminará en tantos elementos comenzando por el final del array. Si se omite, contendrá todos los elementos del array desde **$offset**.

El parámetro opcional **$preserve_keys** reiniciará los índices numéricos de forma predeterminada. Se pueden preservar cambiándolo a true.

array_splice

array_splice (array &$input, int $offset [, int $length [, mixed $replacement = array()]]): array
Elimina una porción del array y la reemplaza por algo. Comienza la eliminación desde **$offset** (si es negativo, comienza por el final).

Si se omite **$length**, se elimina todo desde **$offset** hasta el final del array. Si es positivo, se eliminan tantos elementos como indique su longitud. Si es negativo, el final de la porción eliminada será de tantos elementos como indique la longitud desde el final del array. Si es cero no se elimina ningún elemento.

Si se especifica el **array $replacemen**t, los elementos eliminados se reemplazarán por los elementos de este array.

Para eliminar todo desde **$offset** hasta el final de array cuando se ha especificado **$replacement**, se puede utilizar **count($input)** para **$length**.

array_chunk

array_chunk (array $array, int $size [, bool $preserve_keys = false]): array
Divide un array en fragmentos de tamaño **$size**. El último fragmento puede contener menos elementos que size.
Devuelve un array multidimensional indexado numéricamente, empezando desde cero.

Si **$preserve_keys** es true se mantienen las claves, si no se reindexa numéricamente.

Modificar elementos en arrays

array_shift

array_shift (array &$array): mixed
Quita el primer valor del **$array** y lo devuelve. Los arrays asociativos no varían sus claves, mientras que los numéricos son reindexados y comienzan de cero.

array_unshift

array_unshift (array &$array, mixed $value1 [, mixed $...]): int
Añade los valores **$value** al inicio del array. Igual que en **array_shift()**, en los arrays asociativos no varían sus claves, mientras que los numéricos se reindexan.

array_pop

array_pop (array &$array): mixed
Quita el último valor del array y lo devuelve.

array_push

array_push (array &$array, mixed $value1 [, mixed $...]): int
Inserta uno o más elementos al final de un array. Viene a ser lo mismo que insertar un elemento a un array $array[] = valor, pero se pueden insertar varios de vez. Si se inserta sólo un valor, es mejor utilizar la forma tradicional.

array_flip

array_flip (array $array): array
Intercambia las claves de un array con sus valores.

Contar elementos de un array

La función **count($array)** permite contar los elementos de un array. **int count(mixed $array_or_countable, int $mode = COUNT_NORMAL)**

En un array multidimensional, la función **count($array)** considera el array como un vector de vectores y devuelve simplemente el número de elementos del primer nivel:

Para contar todos los elementos de un array multidimensional, habría que utilizar el segundo parámetro de la función, el modo, con el valor **COUNT_RECURSIVE**.

array_count_values(array $array): array
El array devuelto tiene como índices los valores del array original y como valores la cantidad de veces que aparece el valor.

max(array $values): mixed Devuelve el valor máximo de un array

min(array $values): mixed
Devuelve el valor máximo de un array

Ordenar un array

sort(array &$array, int $sort_flags = SORT_REGULAR): bool: ordena por orden alfabético / numérico de los valores y genera nuevos índices numéricos consecutivos a partir de cero:

rsort(array &$array, int $sort_flags = SORT_REGULAR): bool: ordena por orden alfabético / numérico inverso de los valores y genera nuevos índices numéricos consecutivos a partir de cero:

asort(array &$array, int $sort_flags = SORT_REGULAR): bool: ordena por orden alfabético / numérico de los valores:

arsort(array &$array, int $sort_flags = SORT_REGULAR): bool ordena por orden alfabético / numérico inverso de los valores:

ksort(array &$array, int $sort_flags = SORT_REGULAR): bool: ordena por orden alfabético / numérico de los índices:

krsort(array &$array, int $sort_flags = SORT_REGULAR): bool: ordena por orden alfabético / numérico de los índices

Array inicial	**sort()**	**rsort()**	**asort()**	**arsort()**	**ksort()**	**krsort()**
Array ([5] => cinco [1] => uno [9] => nueve)	Array ([0] =>cinco [1] =>nueve [2] =>uno)	Array ([0] =>uno [1] =>nueve [2] =>cinco)	Array ([5] => cinco [9] => nueve [1] => uno)	Array ([1] => uno [9] => nueve [5] => cinco)	Array ([1] => uno [5] => cinco [9] => nueve)	Array ([9] => nueve [5] => cinco [1] => uno)

Buscar un valor en un array

La función booleana **in_array(mixed $needle, array $haystack, bool $strict = false): bool** devuelve true si el valor se encuentra en el array. Si el argumento booleano **$strict** es true, in_array() comprueba además que los tipos coincidan.

array_search(mixed $needle, array $haystack, bool $strict = false): mixed busca el valor en el array y, si lo encuentra, devuelve un array cuyos valores son los índices de todos los elementos coincidentes.

array_keys(array $array, mixed $search_value, bool $strict = false): array busca el valor en el array y, si lo encuentra, devuelve un array cuyos valores son los índices de todos los elementos coincidentes. Si el argumento booleano **$strict** es true, array_keys() comprueba además que los tipos coincidan.

Reindexar un array

array_values(array $array): array devuelve los valores de un array en el mismo orden que en el array original, pero renumerando los índices desde cero:

Barajar los elementos de un array

shuffle(array &$array): bool baraja los valores de un array. Los índices del array original se pierden, ya que se renumeran desde cero.

Extraer al azar un elemento de un array

La función **array_rand(array $array, int $num = 1): mixed** extrae al azar uno de los índices del array.

Eliminar valores repetidos

array_unique(array $array, int $sort_flags = SORT_STRING): array devuelve un array en el que se han eliminado los valores repetidos.