

## FUNCIONES PARA ARRAYS PHP

Lo que viene entre corchetes significa que es opcional, no es necesario ponerlo al llamar a las funciones.

Cuando una variable se pase como referencia (&) significa que se modifica en el transcurso de la función

array **array\_diff**(array \$array1, array \$array2 [, array \$... ])

Devuelve un array con los elementos que no estén repetidos en cualquiera de los arrays pasados como argumento

array **array\_merge**(array \$array1 [, array \$... ])

Devuelve una array uniendo los arrays pasados como argumento. También se puede usar el operador suma (+). Si son arrays asociativos y coincide una key se sobrescribe el valor.

array **array\_combine**(array \$keys, array \$values )

Toma dos arrays y genera un array asociativo con las claves los valores del primer array, y los valores con los valores del segundo array.

## FUNCIONES DE ORDENACIÓN DE ARRAYS

Para estas funciones se puede añadir un segundo parámetro indicando cómo queremos ordenar.

Modo de ordenación	Descripción
<b>SORT_REGULAR</b>	Compara elementos <b>normalmente</b>
<b>SORT_NUMERIC</b>	Compara elementos de <b>forma numérica</b>
<b>SORT_STRING</b>	Compara elementos como <b>cadenas</b>
<b>SORT_NATURAL</b>	Compara elementos como cadenas usando el orden natural como <b>natsort()</b>
<b>SORT_LOCALE_STRING</b>	Compara elementos como cadenas basándose en la <b>configuración regional</b> en uso
<b>SORT_FLAG_CASE</b>	Se puede combinar con SORT_STRING o SORT_NATURAL para ordenar <b>cadenas</b> de forma <b>insensible a mayúsculas y minúsculas</b>

bool **asort**(array &\$array [, int \$sort\_flags = SORT\_REGULAR ])

Devuelve true en caso de que pueda ordenar de menor a mayor *los valores* del array pasado como parámetro. Como se pasa por referencia, el array quedaría ordenado tras ejecutar la función. **rsort()** igual pero ordena de forma inversa, de mayor a menor.

bool **ksort**(array &\$array [, int \$sort\_flags = SORT\_REGULAR ])

Devuelve true en caso de que pueda ordenar de menor a mayor *las claves* del array pasado como parámetro. Como se pasa por referencia, el array quedaría ordenado tras ejecutar la función. **krsort()** igual pero ordena de forma inversa, de mayor a menor.

La función **usort()** nos permitiría ordenar pero habría que pasarle una función para comparar dos valores.

array **array\_fill**(int \$start\_index, int \$num, mixed \$value)

Desde el índice start, rellena tantas veces como diga num, con el valor value.

array **range**(mixed \$start, mixed \$end)

Genera un array empezando con un elemento start, y terminando con end. Funciones con números y letras.

array **array\_slice**(array \$array, int \$offset, int \$length)

Devuelve un nuevo array que comience en offset y tiene un tamaño de length desde ahí.

array **array\_splice**(array &\$input, int \$offset [, int \$length [, mixed \$replacement = array() ] ] )

Elimina de un array lo que hay después de offset, hasta una longitud, y lo puede sustituir por lo que le pases como replacement

array **array\_chunk** (array \$array, int \$size)

Divide un array en trozos del tamaño size

#### FUNCIONES DE MODIFICAR ELEMENTOS EN UN ARRAY

mixed **array\_shift**(array &\$array)

Quita el primer elemento de un array, y devuelve un array con el elemento que quita

int **array\_unshift**(array &\$array, mixed \$value1 [, mixed \$... ])

Añade al principio del array pasado como parámetro los elementos value1, value2, ...

mixed **array\_pop**(array &\$array)

En el array pasado como parámetro se quita el último elemento. Devuelve ese elemento.

int **array\_push** (array &\$array, mixed \$value1 [, mixed \$... ])

Inserta uno o más elementos value1, ..., en un array.

bool **in\_array**(mixed \$needle, array \$haystack)

Devuelve TRUE si el elemento needle está como valor en el array haystack

bool **array\_key\_exists**(mixed \$key, array \$array)

Devuelve TRUE si la clave key está en el array asociativo array.

array **array\_keys**(array \$array)

Devuelve un array con las claves del array pasado como parámetro

array **array\_values**(array \$array)

Devuelve un array con los valores del array pasado como parámetro