Focus search box

Funciones de Arrays »
« Constantes predefinidas

- Manual de PHP
- Referencia de funciones
- Extensiones relacionadas con variable y tipo
- Arrays

Change language: Spanish

Submit a Pull Request Report a Bug

Ordenamiento de arrays ¶

PHP tiene varias funciones que se ocupan de ordenar arrays (matrices) y este documento existe para ayudar a aclararlo todo.

Las principales diferencias son:

- Algunas ordenan basados en las key de la array, mientras que otras por los valores: `$array['key'] = 'valor';`
- Si se mantiene o no la correlación entre las key y los valores después de la clasificación, lo cual puede significar que las key se restablecen numéricamente (0,1,2 ...)
- El orden de la clasificación: alfabético, de menor a mayor (ascendente), de mayor a menor (descendente), numérico, natural, aleatorio o definido por el usuario
- Nota: Todas estas funciones de clasificación actúan directamente sobre la variable de array misma, en lugar de devolver un nuevo array ordenado.
- Si alguna de estas funciones de clasificación evalúa a dos miembros como iguales, entonces el orden no queda definido (la clasificación no es estable).

Atributos de las funciones de clasificación

Nombre de la función	Ordena por	Mantiene asociación con las key	Orden de la clasificación	Funciones relacionadas
array_multisort()	valor	sí si es asociativa, no si es númerica	primer array u opciones de clasificación	array_walk()
asort()	valor	sí	menor a mayor	arsort()
arsort()	valor	sí	mayor a menor	asort()
krsort()	key	sí	mayor a menor	ksort()
ksort()	key	sí	menor a mayor	asort()
natcasesort()	valores	sí	natural, insensible a mayúsculas y minúsculas	natsort()
natsort()	valor	sí	natural	natcasesort()
rsort()	valor	no	mayor a menor	sort()
shuffle()	valor	no	aleatorio	array_rand()
sort()	valor	no	menor a mayor	rsort()
uasort()	valor	sí	definido por el usuario	uksort()
uksort()	key	sí	definido por el usuario	uasort()
usort()	valor	no	definido por el usuario	uasort()

+ add a note

User Contributed Notes 3 notes

up
down
138
"Matthew Rice" ¶
9 years ago
While this may seem obvious, user-defined array sorting functions (uksort(), uasort(), usort()) will *not* be called if the array does not have *at least two values in it*.

The following code:

```
<?php

function usortTest($a, $b) {
    var_dump($a);
    var_dump($b);
    return -1;
}

$test = array('val1');
usort($test, "usortTest");

$test2 = array('val2', 'val3');
usort($test2, "usortTest");

?>
```

Will output:

```
string(4) "val3"
string(4) "val2"
```

The first array doesn't get sent to the function.

Please, under no circumstance, place any logic that modifies values, or applies non-sorting business logic in these functions as they will not always be executed.

up
down
18
oculiz at gmail dot com ¶
11 years ago
Another way to do a case case-insensitive sort by key would simply be:

```
<?php
uksort($array, 'strcasecmp');
?>
```

Since strcasecmp is already predefined in php it saves you the trouble to actually write the comparison function yourself.

up
down
-41
Hayley Watson ¶
6 years ago
Stabilizing the sort functions (in this case, usort).

```
<?php
function stable_usort(&$array, $cmp)
{
```

```
    $i = 0;
    $array = array_map(function($elt)use(&$i)
    {
        return [$i++, $elt];
    }, $array);
    usort($array, function($a, $b)use($cmp)
    {
        return $cmp($a[1], $b[1]) ?: ($a[0] - $b[0]);
    });
    $array = array_column($array, 1);
}
?>
```

Tags each array element with its original position in the array so that when the comparison function returns 0 the tie can be broken to put the earlier element first.

+ add a note