PHP Array Functions

Function	Description
array()	Creates an array
array_change_key_case() array_change_key_case(*array,case*); Case: CASE_LOWER - Default value. Changes the keys to lowercase CASE_UPPER - Changes the keys to uppercase	Changes all keys in an array to lowercase or uppercase <?php $age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43"); print_r(array_change_key_case($age,CASE_UPPER)); ?> **Output** Array ([PETER] => 35 [BEN] => 37 [JOE] => 43)
array_chunk() array_chunk(*array,size*);	Splits an array into chunks of arrays <?php $cars=array("Volvo","BMW","Toyota","Honda","Mercedes","Opel"); print_r(array_chunk($cars,2)); ?> **Output** Array ([0] => Array ([0] => Volvo [1] => BMW) [1] => Array ([0] => Toyota [1] => Honda) [2] => Array ([0] => Mercedes [1] => Opel))
array_combine() array_combine(*keys array,values array*);	Creates an array by using the elements from one "keys" array and one "values" array <?php $fname=array("Peter","Ben","Joe"); $age=array("35","37","43"); $c=array_combine($fname,$age); print_r($c); ?> **Output** Array ([Peter] => 35 [Ben] => 37 [Joe] => 43)
array_fill() array_fill(*index,number,value*);	Fills an array with values <?php $a1=array_fill(3,4,"blue"); print_r($a1); ?> **Output** Array ([3] => blue [4] => blue [5] => blue [6] => blue) Array ([0] => red)
array_flip()	Flips/Exchanges all keys with their associated values in an array <?php $a1=array("a"=>"red","b"=>"green","c"=>"blue"

PHP Array Functions

	,"d"=>"yellow"); $result=array_flip($a1); print_r($result); ?> **output** Array ([red] => a [green] => b [blue] => c [yellow] => d)
array_intersect()	Compare arrays, and returns the matches (compare values only) This function compares the values of two or more arrays, and return an array that contains the entries from *array1* that are present in *array2*, *array3*, etc. <?php $a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow"); $a2=array("e"=>"red","f"=>"green","g"=>"blue"); $result=array_intersect($a1,$a2); print_r($result); ?> **Output** Array ([a] => red [b] => green [c] => blue)

array_key_exists()	Checks if the specified key exists in the array
	The array_key_exists() function checks an array for a specified key, and returns true if the key exists and false if the key does not exist.
	<?php
	$a=array("Volvo"=>"XC90","BMW"=>"X5");
	if (array_key_exists("Volvo",$a))
	{
	echo "Key exists!";
	}
	else
	{
	echo "Key does not exist!";
	}
	?>
	Output
	Key exists!

array_keys()	Returns all the keys of an array <?php $a=array("Volvo"=>"XC90","BMW"=>"X5","Toyota"=>"Highlander"); print_r(array_keys($a)); ?> **Output** Array ([0] => Volvo [1] => BMW [2] => Toyota)

PHP Array Functions

array_merge()	Merges one or more arrays into one array
	`<?php`
	`$a1=array("red","green");`
	`$a2=array("blue","yellow");`
	`print_r(array_merge($a1,$a2));`
	`?>`
	Ouput
	Array ([0] => red [1] => green [2] => blue [3] => yellow)

array_multisort()
array_multisort(*array1,sorting order,sorting type,array2,array3...*)
sorting order Optional. Specifies the sorting order. Possible values:
SORT_ASC - Default. Sort in ascending order (A-Z)
SORT_DESC - Sort in descending order (Z-A)
sorting type Optional. Specifies the type to use, when comparing elements. Possible values:
SORT_REGULAR - Default. Compare elements normally (Standard ASCII)
SORT_NUMERIC - Compare elements as numeric values
SORT_STRING - Compare elements as string values
	Sorts multiple or multi-dimensional arrays
	`<?php`
	`$a1=array("Dog","Cat");`
	`$a2=array("Fido","Missy");`
	`array_multisort($a1,$a2);`
	`print_r($a1);`
	`print_r($a2);`
	`?>`
	Output
	Array ([0] => Cat [1] => Dog) Array ([0] => Missy [1] => Fido)

array_pop()	Deletes the last element of an array
	`<?php`
	`$a=array("red","green","blue");`
	`array_pop($a);`
	`print_r($a);`
	`?>`
	Output
	Array ([0] => red [1] => green)

array_product()	Calculates the product of the values in an array
	`<?php`
	`$a=array(5,5);`
	`echo(array_product($a));`
	`?>`
	Output
	25

PHP Array Functions

array_push()	Inserts one or more elements to the end of an array and Returns the new number of elements in the array
	`<?php`
	`$a=array("red","green");`
	`array_push($a,"blue","yellow");`
	`print_r($a);`
	`?>`
	Output
	Array ([0] => red [1] => green [2] => blue [3] => yellow)

array_replace()	Replaces the values of the first array with the values from following arrays
	`<?php`
	`$a1=array("red","green");`
	`$a2=array("blue","yellow");`
	`print_r(array_replace($a1,$a2));`
	`?>`
	Output
	Array ([0] => blue [1] => yellow)

array_reverse()	Returns an array in the reverse order
	`<?php`
	`$a=array("a"=>"Volvo","b"=>"BMW","c"=>"Toyota");`
	`print_r(array_reverse($a));`
	`?>`
	Output
	Array ([c] => Toyota [b] => BMW [a] => Volvo)

array_search()
`array_search(`*value*`,`*array*`,`*strict*`)`
***strict*Optional. If this parameter is set to TRUE, then this function will search for identical elements in the array. Possible values:**
true
false - Default
When set to true, the number 5 is not the same as the string 5
	Searches an array for a given value and returns the key and FALSE otherwise. If the value is found in the array more than once, the first matching key is returned.
	`<?php`
	`$a=array("a"=>"red","b"=>"green","c"=>"blue");`
	`echo array_search("red",$a);`
	`?>`
	
	Output
	A

PHP Array Functions

array_shift()	Removes the first element from an array, and returns the value of the removed element
	`<?php`
	`$a=array("a"=>"red","b"=>"green","c"=>"blue");`
	`echo array_shift($a);`
	`print_r ($a);`
	`?>`
	Output
	red
	Array ([b] => green [c] => blue)

array_slice()
`array_slice(array,start,length)`
	Returns selected parts of an array
	`<?php`
	`$a=array("red","green","blue","yellow","brown");`
	`print_r(array_slice($a,2));`
	`?>`
	Output
	Array ([0] => blue [1] => yellow [2] => brown)

array_splice()
`array_splice(array,start,length,array)`
	Removes and replaces specified elements of an array
	`<?php`
	`$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");`
	`$a2=array("a"=>"purple","b"=>"orange");`
	`array_splice($a1,0,2,$a2);`
	`print_r($a1);`
	`?>`
	Output
	Array ([0] => purple [1] => orange [c] => blue [d] => yellow)

array_sum()	Returns the sum of the values in an array
	`<?php`
	`$a=array(5,15,25);`
	`echo array_sum($a);`
	`?>`
	Output
	45

array_unique()	Removes duplicate values from an array
	`<?php`
	`$a=array("a"=>"red","b"=>"green","c"=>"red");`
	`print_r(array_unique($a));`
	`?>`
	Output
	Array ([a] => red [b] => green)

PHP Array Functions

array_unshift()	Adds one or more elements to the beginning of an array
	`<?php`
	`$a=array("a"=>"red","b"=>"green");`
	`array_unshift($a,"blue");`
	`print_r($a);`
	`?>`
	Output
	Array ([0] => blue [a] => red [b] => green)

array_values()	Returns all the values of an array

arsort()	Sorts an associative array in descending order, according to the value
	`<?php`
	`$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");`
	`arsort($age);`
	`?>`
	Output
	Key=Joe, Value=43
	Key=Ben, Value=37
	Key=Peter, Value=35

asort()	Sorts an associative array in ascending order, according to the value
count()	Returns the number of elements in an array
current() or pos()	Returns the current element in an array
end()	Sets the internal pointer of an array to its last element
in_array() in_array(*search,array*)	Checks if a specified value exists in an array and returns TRUE or FALSE accordingly

key()	Fetches a key from an array
	`<?php`
	`$people=array("Peter","Joe","Glenn","Cleveland");`
	`echo "The key from the current position is: " . key($people);`
	`?>`
	Output
	The key from the current position is: 0

krsort()	Sorts an associative array in descending order, according to the key
ksort()	Sorts an associative array in ascending order, according to the key

PHP Array Functions

list()	Assigns variables as if they were an array
	`<?php`
	`$my_array = array("Dog","Cat","Horse");`
	
	`list($a, $b, $c) = $my_array;`
	`echo "I have several animals, a $a, a $b and a $c.";`
	`?>`
	Output
	I have several animals, a Dog, a Cat and a Horse.

next()	Advance the internal array pointer of an array
prev()	Rewinds the internal array pointer

range()
`range(low,high,step)`
	Creates an array containing a range of elements
	`<?php`
	`$number = range(0,5);`
	`print_r ($number);`
	`?>`
	Output
	Array ([0] => 0 [1] => 1 [2] => 2 [3] => 3 [4] => 4 [5] => 5)

reset()	Sets the internal pointer of an array to its first element
rsort()	Sorts an indexed array in descending order
sizeof()	Alias of count()
sort()	Sorts an indexed array in ascending order
uasort()	Sorts an array by values using a user-defined comparison function
uksort()	Sorts an array by keys using a user-defined comparison function

usort()	Sorts an array using a user-defined comparison function
	<?php
	function mysort($a,$b)
	{
	return $a-$b;
	}
	$a=array(4,7,2,9,10);
	usort($a,"mysort");
	print_r($a);
	?>
	//Output
	Array ([0] => 2 [1] => 4 [2] => 7 [3] => 9 [4] => 10)