

Array functions

<code>array array_change_key_case (array \$array [, int \$case = CASE_LOWER])</code>	Returns an array with all keys from array lowercased or uppercased. Numbered indices are left as is.	Returns an array with its keys lower or uppercased, or FALSE if array is not an array.
<code>array array_chunk (array \$array , int \$size [, bool \$preserve_keys = false])</code>	Chunks an array into arrays with size elements. The last chunk may contain less than size elements.	Returns a multidimensional numerically indexed array, starting with zero, with each dimension containing size elements.
<code>array array_column (array \$input , mixed \$column_key [, mixed \$index_key = null])</code>	returns the values from a single column of the input, identified by the column_key. Optionally, an index_key may be provided to index the values in the returned array by the values from the index_key column of the input array.	Returns an array of values representing a single column from the input array.
<code>array array_combine (array \$keys , array \$values)</code>	Creates an array by using the values from the keys array as keys and the values from the values array as the corresponding values.	Returns the combined array, FALSE if the number of elements for each array isn't equal.
<code>array array_count_values (array \$array)</code>	<code>array_count_values()</code> returns an array using the values of array as keys and their frequency in array as values.	Returns an associative array of values from array as keys and their count as value.
<code>array array_diff_assoc (array \$array1 , array \$array2 [, array \$...])</code>	Compares array1 against array2 and returns the difference. Unlike <code>array_diff()</code> the array keys are also used in the comparison.	Returns an array containing all the values from array1 that are not present in any of the other arrays.
<code>array array_diff_key (array \$array1 , array \$array2 [, array \$...])</code>	Compares the keys from array1 against the keys from array2 and returns the difference. This function is like <code>array_diff()</code> except the comparison is done on the keys instead of the values.	Returns an array containing all the entries from array1 whose keys are not present in any of the other arrays.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 1 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

array array_diff_uassoc (array \$array1 , array \$array2 [, array \$...], callable \$key_compare_func)	Compares array1 against array2 and returns the difference. Unlike array_diff() the array keys are used in the comparison. Unlike array_diff_assoc() an user supplied callback function is used for the indices comparison, not internal function.	Returns an array containing all the entries from array1 that are not present in any of the other arrays.
array array_diff_ukey (array \$array1 , array \$array2 [, array \$...], callable \$key_compare_func)	Compares the keys from array1 against the keys from array2 and returns the difference. This function is like array_diff() except the comparison is done on the keys instead of the values. Unlike array_diff_key() a user supplied callback function is used for the indices comparison, not internal function.	Returns an array containing all the entries from array1 that are not present in any of the other arrays.
array array_diff (array \$array1 , array \$array2 [, array \$...])	Compares array1 against one or more other arrays and returns the values in array1 that are not present in any of the other arrays.	Returns an array containing all the entries from array1 that are not present in any of the other arrays.
array array_fill_keys (array \$keys , mixed \$value)	Fills an array with the value of the value parameter, using the values of the keys array as keys.	Returns the filled array
array array_fill (int \$start_index , int \$num , mixed \$value)	Fills an array with num entries of the value of the value parameter, keys starting at the start_index parameter.	Returns the filled array
array array_filter (array \$array [, callable \$callback [, int \$flag = 0]])	Iterates over each value in the array passing them to the callback function. If the callback function returns true, the current value from array is returned into the result array. Array keys are preserved.	Returns the filtered array.



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 2 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Array functions (cont)

<code>array_array_flip (array \$array)</code>	<code>array_flip()</code> returns an array in flip order, i.e. keys from array become values and values from array become keys. Note that the values of array need to be valid keys, i.e. they need to be either integer or string. A warning will be emitted if a value has the wrong type, and the key/value pair in question will not be included in the result. If a value has several occurrences, the latest key will be used as its value, and all others will be lost.	Returns the flipped array on success and NULL on failure.
<code>array_array_intersect_assoc (array \$array1 , array \$array2 [, array \$...])</code>	<code>array_intersect_assoc()</code> returns an array containing all the values of array1 that are present in all the arguments. Note that the keys are also used in the comparison unlike in <code>array_intersect()</code> .	Returns an associative array containing all the values in array1 that are present in all of the arguments.
<code>array_array_intersect_key (array \$array1 , array \$array2 [, array \$...])</code>	<code>array_intersect_key()</code> returns an array containing all the entries of array1 which have keys that are present in all the arguments.	Returns an associative array containing all the entries of array1 which have keys that are present in all arguments.
<code>array_array_intersect_uassoc (array \$array1 , array \$array2 [, array \$...], callable \$key_compare_func)</code>	<code>array_intersect_uassoc()</code> returns an array containing all the values of array1 that are present in all the arguments. Note that the keys are used in the comparison unlike in <code>array_intersect()</code> .	Returns the values of array1 whose values exist in all of the arguments.
<code>array_array_intersect_ukey (array \$array1 , array \$array2 [, array \$...], callable \$key_compare_func)</code>	<code>array_intersect_ukey()</code> returns an array containing all the values of array1 which have matching keys that are present in all the arguments.	Returns the values of array1 whose keys exist in all the arguments.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 3 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

array array_ intersect (array \$array1 , array \$array2 [, array \$...])	array_intersect() returns an array containing all the values of array1 that are present in all the arguments. Note that keys are preserved.	Returns an array containing all of the values in array1 whose values exist in all of the parameters.
bool array_ key_exists (mixed \$key , array \$array)	array_key_exists() returns TRUE if the given key is set in the array. key can be any value possible for an array index.	Returns TRUE on success or FALSE on failure.
array array_ keys (array \$array [, mixed \$search_value = null [, bool \$strict = false]])	array_keys() returns the keys, numeric and string, from the array. If the optional search_value is specified, then only the keys for that value are returned. Otherwise, all the keys from the array are returned.	Returns an array of all the keys in array.
array array_ map (callable \$callback , array \$array1 [, array \$...])	array_map() returns an array containing all the elements of array1 after applying the callback function to each one. The number of parameters that the callback function accepts should match the number of arrays passed to the array_map()	Returns an array containing all the elements of array1 after applying the callback function to each one.
array array_ merge_recursive (array \$array1 [, array \$...])	array_merge_recursive() merges the elements of one or more arrays together so that the values of one are appended to the end of the previous one. It returns the resulting array. If the input arrays have the same string keys, then the values for these keys are merged together into an array, and this is done recursively, so that if one of the values is an array itself, the function will merge it with a corresponding entry in another array too. If, however, the arrays have the same numeric key, the later value will not overwrite the original value, but will be appended.	An array of values resulted from merging the arguments together.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 4 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

array array_merge (array \$array1 [, array \$...])	Merges the elements of one or more arrays together so that the values of one are appended to the end of the previous one. It returns the resulting array. If the input arrays have the same string keys, then the later value for that key will overwrite the previous one. If, however, the arrays contain numeric keys, the later value will not overwrite the original value, but will be appended. Values in the input array with numeric keys will be renumbered with incrementing keys starting from zero in the result array.	Returns the resulting array.
bool array_multisort (array &\$array1 [, mixed \$array1_sort_order = SORT_ASC [, mixed \$array1_sort_flags = SORT_REGULAR [, mixed \$...]]])	array_multisort() can be used to sort several arrays at once, or a multi-dimensional array by one or more dimensions. Associative (string) keys will be maintained, but numeric keys will be re-indexed.	Returns TRUE on success or FALSE on failure.
array array_pad (array \$array , int \$size , mixed \$value)	array_pad() returns a copy of the array padded to size specified by size with value value. If size is positive then the array is padded on the right, if it's negative then on the left. If the absolute value of size is less than or equal to the length of the array then no padding takes place. It is possible to add at most 1048576 elements at a time.	Returns a copy of the array padded to size specified by size with value value. If size is positive then the array is padded on the right, if it's negative then on the left. If the absolute value of size is less than or equal to the length of the array then no padding takes place.
mixed array_pop (array &\$array)	array_pop() pops and returns the last value of the array, shortening the array by one element. Note: This function will reset() the array pointer of the input array after use.	Returns the last value of array. If array is empty (or is not an array), NULL will be returned.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 5 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

number array_product (array \$array)	array_product() returns the product of values in an array.	Returns the product as an integer or float.
int array_push (array &\$array , mixed \$value1 [, mixed \$...])	array_push() treats array as a stack, and pushes the passed variables onto the end of array. The length of array increases by the number of variables pushed. Note: If you use array_push() to add one element to the array it's better to use \$array[] = because in that way there is no overhead of calling a function. Note: array_push() will raise a warning if the first argument is not an array. This differs from the \$var[] behaviour where a new array is created.	Returns the new number of elements in the array.
mixed array_rand (array \$array [, int \$num = 1])	Picks one or more random entries out of an array, and returns the key (or keys) of the random entries. It uses a pseudo random number generator that is not suitable for cryptographic purposes.	When picking only one entry, array_rand() returns the key for a random entry. Otherwise, an array of keys for the random entries is returned. This is done so that random keys can be picked from the array as well as random values. Trying to pick more elements than there are in the array will result in an E_WARNING level error, and NULL will be returned.
mixed array_reduce (array \$array , callable \$callback [, mixed \$initial = NULL])	array_reduce() applies iteratively the callback function to the elements of the array, so as to reduce the array to a single value.	Returns the resulting value. If the array is empty and initial is not passed, array_reduce() returns NULL.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 6 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

array_replace_recursive (array \$array1 , array \$array2 [, array \$...])	<p><code>array_replace_recursive()</code> replaces the values of <code>array1</code> with the same values from all the following arrays. If a key from the first array exists in the second array, its value will be replaced by the value from the second array. If the key exists in the second array, and not the first, it will be created in the first array. If a key only exists in the first array, it will be left as is. If several arrays are passed for replacement, they will be processed in order, the later array overwriting the previous values. <code>array_replace_recursive()</code> is recursive : it will recurse into arrays and apply the same process to the inner value. When the value in <code>array1</code> is scalar, it will be replaced by the value in <code>array2</code>, may it be scalar or array. When the value in <code>array1</code> and <code>array2</code> are both arrays, <code>array_replace_recursive()</code> will replace their respective value recursively.</p>	Returns an array, or NULL if an error occurs.
--	---	---



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 7 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

array array_ replace (array \$array1 , array \$array2 [, array \$...])	array_replace() replaces the values of array1 with values having the same keys in each of the following arrays. If a key from the first array exists in the second array, its value will be replaced by the value from the second array. If the key exists in the second array, and not the first, it will be created in the first array. If a key only exists in the first array, it will be left as is. If several arrays are passed for replacement, they will be processed in order, the later arrays overwriting the previous values. array_replace() is not recursive : it will replace values in the first array by whatever type is in the second array.	Returns an array, or NULL if an error occurs.
array array_ reverse (array \$array [, bool \$preserve_ keys = false])	Takes an input array and returns a new array with the order of the elements reversed.	Returns the reversed array.
mixed array_ search (mixed \$needle , array \$haystack [, bool \$strict = false])	Searches haystack for needle.	Returns the key for needle if it is found in the array, FALSE otherwise. If needle is found in haystack more than once, the first matching key is returned. To return the keys for all matching values, use array_keys() with the optional search_value parameter instead. Warning This function may return Boolean FALSE, but may also return a non-Boolean value which evaluates to FALSE. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 8 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

mixed <code>array_shift (array &\$array)</code>	<p><code>array_shift()</code> shifts the first value of the array off and returns it, shortening the array by one element and moving everything down. All numerical array keys will be modified to start counting from zero while literal keys won't be touched.</p> <p>Note: This function will <code>reset()</code> the array pointer of the input array after use.</p>	Returns the shifted value, or NULL if array is empty or is not an array.
array <code>array_slice (array \$array , int \$offset [, int \$length = NULL [, bool \$preserve_keys = false]])</code>	<p><code>array_slice()</code> returns the sequence of elements from the array array as specified by the offset and length parameters.</p>	Returns the slice. If the offset is larger than the size of the array then returns an empty array.
array <code>array_splice (array &\$input , int \$offset [, int \$length = 0 [, mixed \$replacement = array()]])</code>	<p>Removes the elements designated by offset and length from the input array, and replaces them with the elements of the replacement array, if supplied.</p> <p>Note that numeric keys in input are not preserved.</p> <p>Note: If replacement is not an array, it will be typecast to one (i.e. <code>(array) \$replacement</code>). This may result in unexpected behavior when using an object or NULL replacement.</p>	Returns an array consisting of the extracted elements.
number <code>array_sum (array \$array)</code>	<p><code>array_sum()</code> returns the sum of values in an array.</p>	Returns the sum of values as an integer or float.



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 9 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Array functions (cont)

array array_udiff_assoc (array \$array1 , array \$array2 [, array \$...], callable \$value_compare_func)	<p>Computes the difference of arrays with additional index check, compares data by a callback function.</p> <p>Note: Please note that this function only checks one dimension of a n-dimensional array. Of course you can check deeper dimensions by using, for example, array_udiff_assoc(\$array1[0], \$array2[0], "some_comparison_func");.</p>	array_udiff_assoc() returns an array containing all the values from array1 that are not present in any of the other arguments. Note that the keys are used in the comparison unlike array_diff() and array_udiff(). The comparison of arrays' data is performed by using an user-supplied callback. In this aspect the behaviour is opposite to the behaviour of array_diff_assoc() which uses internal function for comparison.
array array_udiff_uassoc (array \$array1 , array \$array2 [, array \$...], callable \$value_compare_func , callable \$key_compare_func)	<p>Computes the difference of arrays with additional index check, compares data and indexes by a callback function. Note that the keys are used in the comparison unlike array_diff() and array_udiff().</p>	Returns an array containing all the values from array1 that are not present in any of the other arguments.
array array_udiff (array \$array1 , array \$array2 [, array \$...], callable \$value_compare_func)	<p>Computes the difference of arrays by using a callback function for data comparison. This is unlike array_diff() which uses an internal function for comparing the data.</p>	Returns an array containing all the values of array1 that are not present in any of the other arguments.
array array_uintersect_assoc (array \$array1 , array \$array2 [, array \$...], callable \$value_compare_func)	<p>Computes the intersection of arrays with additional index check, compares data by a callback function. Note that the keys are used in the comparison unlike in array_uintersect(). The data is compared by using a callback function.</p>	Returns an array containing all the values of array1 that are present in all the arguments.
array array_uintersect_uassoc (array \$array1 , array \$array2 [, array \$...], callable \$value_compare_func , callable \$key_compare_func)	<p>Computes the intersection of arrays with additional index check, compares data and indexes by separate callback functions.</p>	Returns an array containing all the values of array1 that are present in all the arguments.



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 10 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Array functions (cont)

<pre>array array_uintersect (array \$array1 , array \$array2 [, array \$...], callable \$value_compa- re_func)</pre>	<p>Computes the intersection of arrays, compares data by a callback function.</p>	<p>Returns an array containing all the values of array1 that are present in all the arguments.</p>
<pre>array array_unique (array \$array [, int \$sort_ flags = SORT_STRING])</pre>	<p>Takes an input array and returns a new array without duplicate values. Note that keys are preserved. array_unique() sorts the values treated as string at first, then will keep the first key encountered for every value, and ignore all following keys. It does not mean that the key of the first related value from the unsorted array will be kept.</p> <p>Note: Two elements are considered equal if and only if (string) \$elem1 === (string) \$elem2 i.e. when the string representation is the same, the first element will be used.</p>	<p>Returns the filtered array.</p>
<pre>int array_unshift (array &\$array , mixed \$value1 [, mixed \$...])</pre>	<p>array_unshift() prepends passed elements to the front of the array. Note that the list of elements is prepended as a whole, so that the prepended elements stay in the same order. All numerical array keys will be modified to start counting from zero while literal keys won't be touched.</p>	<p>Returns the new number of elements in the array.</p>
<pre>array array_values (array \$array)</pre>	<p>array_values() returns all the values from the array and indexes the array numerically.</p>	<p>Returns an indexed array of values.</p>
<pre>bool array_walk_recu- rsive (array &\$array , callable \$callback [, mixed \$userdata = NULL])</pre>	<p>Applies the user-defined callback function to each element of the array. This function will recurse into deeper arrays.</p>	<p>Returns TRUE on success or FALSE on failure.</p>



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 11 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

<code>bool array_walk (array &\$array , callable \$callback [, mixed \$userdata = NULL])</code>	Applies the user-defined callback function to each element of the array array. array_walk() is not affected by the internal array pointer of array. array_walk() will walk through the entire array regardless of pointer position.	Returns TRUE on success or FALSE on failure. If function callback requires more parameters than given to it, an error of level E_WARNING will be generated each time array_walk() calls callback.
<code>array array ([mixed \$...])</code>	Creates an array. Read the section on the array type for more information on what an array is.	Returns an array of the parameters. The parameters can be given an index with the => operator. Read the section on the array type for more information on what an array is.
<code>bool arsort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	This function sorts an array such that array indices maintain their correlation with the array elements they are associated with. This is used mainly when sorting associative arrays where the actual element order is significant. Note: If two members compare as equal, their relative order in the sorted array is undefined.	Returns TRUE on success or FALSE on failure.
<code>bool asort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	This function sorts an array such that array indices maintain their correlation with the array elements they are associated with. This is used mainly when sorting associative arrays where the actual element order is significant. Note: If two members compare as equal, their relative order in the sorted array is undefined.	Returns TRUE on success or FALSE on failure.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 12 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

<code>array</code>	Creates an array containing variables and their values. For each of these,	Returns the output array with all the variables added to it.
<code>compact (</code>	<code>compact()</code> looks for a variable with that name in the current symbol table and	
<code>mixed</code>	adds it to the output array such that the variable name becomes the key and	
<code>\$varname1</code>	the contents of the variable become the value for that key. In short, it does the	
<code>[, mixed \$...]</code>	opposite of <code>extract()</code> . Any strings that are not set will simply be skipped.	
<code>)</code>		
<code>int count (</code>	Counts all elements in an array, or something in an object. For objects, if you	Returns the number of elements in <code>array_or_</code>
<code>mixed</code>	have SPL installed, you can hook into <code>count()</code> by implementing interface	<code>countable</code> . If the parameter is not an array or
<code>\$array_or-</code>	<code>Countable</code> . The interface has exactly one method, <code>Countable::count()</code> , which	not an object with implemented <code>Countable</code>
<code>_countable [,</code>	returns the return value for the <code>count()</code> function. Please see the Array section of	interface, 1 will be returned. There is one
<code>int \$mode =</code>	the manual for a detailed explanation of how arrays are implemented and used	exception, if <code>array_or_countable</code> is <code>NULL</code> , 0
<code>COUNT_-</code>	in PHP.	will be returned.
<code>NORMAL])</code>		Caution <code>count()</code> may return 0 for a variable
		that isn't set, but it may also return 0 for a
		variable that has been initialized with an
		empty array. Use <code>isset()</code> to test if a variable is
		set.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 13 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

mixed current (array &\$array)	Every array has an internal pointer to its "current" element, which is initialized to the first element inserted into the array.	The current() function simply returns the value of the array element that's currently being pointed to by the internal pointer. It does not move the pointer in any way. If the internal pointer points beyond the end of the elements list or the array is empty, current() returns FALSE. Warning This function may return Boolean FALSE, but may also return a non-Boolean value which evaluates to FALSE. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.
array each (array &\$array)	Return the current key and value pair from an array and advance the array cursor. After each() has executed, the array cursor will be left on the next element of the array, or past the last element if it hits the end of the array. You have to use reset() if you want to traverse the array again using each.	Returns the current key and value pair from the array array. This pair is returned in a four-element array, with the keys 0, 1, key, and value. Elements 0 and key contain the key name of the array element, and 1 and value contain the data. If the internal pointer for the array points past the end of the array contents, each() returns FALSE.
mixed end (array &\$array)	end() advances array's internal pointer to the last element, and returns its value.	Returns the value of the last element or FALSE for empty array.
int extract (array &\$array [, int \$flags = EXTR_OVERWRITE [, string \$prefix = NULL]])	Import variables from an array into the current symbol table. Checks each key to see whether it has a valid variable name. It also checks for collisions with existing variables in the symbol table.	Returns the number of variables successfully imported into the symbol table.
bool in_array (mixed \$needle , array \$haystack [, bool \$strict = FALSE])	Searches haystack for needle using loose comparison unless strict is set.	Returns TRUE if needle is found in the array, FALSE otherwise.



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 14 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Array functions (cont)

<code>key_exists</code>	This function is an alias of: <code>array_key_exists()</code> .	
<code>mixed key (array &\$array)</code>	<code>key()</code> returns the index element of the current array position.	The <code>key()</code> function simply returns the key of the array element that's currently being pointed to by the internal pointer. It does not move the pointer in any way. If the internal pointer points beyond the end of the elements list or the array is empty, <code>key()</code> returns <code>NULL</code> .
<code>bool krsort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	Sorts an array by key in reverse order, maintaining key to data correlations. This is useful mainly for associative arrays.	Returns <code>TRUE</code> on success or <code>FALSE</code> on failure.
<code>bool ksort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	Sorts an array by key, maintaining key to data correlations. This is useful mainly for associative arrays.	Returns <code>TRUE</code> on success or <code>FALSE</code> on failure.
<code>array list (mixed \$var1 [, mixed \$...])</code>	Like <code>array()</code> , this is not really a function, but a language construct. <code>list()</code> is used to assign a list of variables in one operation. Note: <code>list()</code> only works on numerical arrays and assumes the numerical indices start at 0. Warning In PHP 5, <code>list()</code> assigns the values starting with the right-most parameter. Warning Modification of the array during <code>list()</code> execution (e.g. using <code>list(\$a, \$b) = \$b</code>) results in undefined behavior.	Returns the assigned array.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 15 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

bool natcasesort (array &\$- array)	<p>natcasesort() is a case insensitive version of natsort(). This function implements a sort algorithm that orders alphanumeric strings in the way a human being would while maintaining key/value associations. This is described as a "natural ordering".</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	Returns TRUE on success or FALSE on failure.
bool natsort (array &\$- array)	<p>This function implements a sort algorithm that orders alphanumeric strings in the way a human being would while maintaining key/value associations. This is described as a "natural ordering". An example of the difference between this algorithm and the regular computer string sorting algorithms (used in sort()) can be seen in the example below.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	Returns TRUE on success or FALSE on failure.
mixed next (array &\$- array)	<p>next() behaves like current(), with one difference. It advances the internal array pointer one place forward before returning the element value. That means it returns the next array value and advances the internal array pointer by one.</p>	<p>Returns the array value in the next place that's pointed to by the internal array pointer, or FALSE if there are no more elements.</p> <p>Warning This function may return Boolean FALSE, but may also return a non-Boolean value which evaluates to FALSE. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.</p>
pos	This function is an alias of: current()	



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 16 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

mixed <code>prev (array &\$array)</code>	<p>Rewind the internal array pointer. <code>prev()</code> behaves just like <code>next()</code>, except it rewinds the internal array pointer one place instead of advancing it.</p> <p>Note: You won't be able to distinguish the beginning of an array from a boolean <code>FALSE</code> element. To properly traverse an array which may contain <code>FALSE</code> elements, see the <code>each()</code> function.</p>	<p>Returns the array value in the previous place that's pointed to by the internal array pointer, or <code>FALSE</code> if there are no more elements.</p> <p>Warning This function may return Boolean <code>FALSE</code>, but may also return a non-Boolean value which evaluates to <code>FALSE</code>. Please read the section on Booleans for more information. Use the <code>===</code> operator for testing the return value of this function.</p>
array <code>range (mixed \$start , mixed \$end [, number \$step = 1])</code>	<p>Create an array containing a range of elements.</p>	<p>Returns an array of elements from start to end, inclusive.</p>
mixed <code>reset (array &\$array)</code>	<p><code>reset()</code> rewinds array's internal pointer to the first element and returns the value of the first array element.</p>	<p>Returns the value of the first array element, or <code>FALSE</code> if the array is empty.</p>
bool <code>rsort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	<p>This function sorts an array in reverse order (highest to lowest).</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	<p>Returns <code>TRUE</code> on success or <code>FALSE</code> on failure.</p>
bool <code>shuffle (array &\$array)</code>	<p>This function shuffles (randomizes the order of the elements in) an array. It uses a pseudo random number generator that is not suitable for cryptographic purposes.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p> <p>Note: This function assigns new keys to the elements in array. It will remove any existing keys that may have been assigned, rather than just reordering the keys.</p>	<p>Returns <code>TRUE</code> on success or <code>FALSE</code> on failure.</p>
<code>sizeof</code>	<p>This function is an alias of: <code>count()</code>.</p>	



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 17 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Array functions (cont)

<code>bool sort (array &\$array [, int \$sort_flags = SORT_REGULAR])</code>	<p>This function sorts an array. Elements will be arranged from lowest to highest when this function has completed.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	Returns TRUE on success or FALSE on failure.
<code>bool uasort (array &\$array , callable \$value_compare_func)</code>	<p>This function sorts an array such that array indices maintain their correlation with the array elements they are associated with, using a user-defined comparison function. This is used mainly when sorting associative arrays where the actual element order is significant.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	Returns TRUE on success or FALSE on failure.
<code>bool uksort (array &\$array , callable \$key_compare_func)</code>	<p><code>uksort()</code> will sort the keys of an array using a user-supplied comparison function. If the array you wish to sort needs to be sorted by some non-trivial criteria, you should use this function.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p>	Returns TRUE on success or FALSE on failure.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 18 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Array functions (cont)

bool usort (array &\$array , callable \$value_compare_func)	<p>This function will sort an array by its values using a user-supplied comparison function. If the array you wish to sort needs to be sorted by some non-trivial criteria, you should use this function.</p> <p>Note: If two members compare as equal, their relative order in the sorted array is undefined.</p> <p>Note: This function assigns new keys to the elements in array. It will remove any existing keys that may have been assigned, rather than just reordering the keys.</p>	Returns TRUE on success or FALSE on failure.
---	--	--

Array constants

ARRAY_FILTER_USER_KEY	pass key as the only argument to callback instead of the value
ARRAY_FILTER_BOTH	pass both value and key as arguments to callback instead of the value
SORT_REGULAR	compare items normally (don't change types)
SORT_NUMERIC	compare items numerically
SORT_STRING	compare items as strings
SORT_LOCALE_STRING	compare items as strings, based on the current locale. It uses the locale, which can be changed using setlocale()
SORT_NATURAL	compare items as strings using "natural ordering" like natsort()
SORT_FLAG_CASE	can be combined (bitwise OR) with SORT_STRING or SORT_NATURAL to sort strings case-insensitively
EXTR_OVERWRITE	If there is a collision, overwrite the existing variable.
EXTR_SKIP	If there is a collision, don't overwrite the existing variable.
EXTR_PREFIX_SAME	If there is a collision, prefix the variable name with prefix.
EXTR_PREFIX_ALL	Prefix all variable names with prefix.
EXTR_PREFIX_INVALID	Only prefix invalid/numeric variable names with prefix.
EXTR_IF_EXISTS	Only overwrite the variable if it already exists in the current symbol table, otherwise do nothing. This is useful for defining a list of valid variables and then extracting only those variables you have defined out of \$_REQUEST, for example.
EXTR_PREFIX_IF_EXISTS	Only create prefixed variable names if the non-prefixed version of the same variable exists in the current symbol table.



Array constants (cont)

EXTR_REFS Extracts variables as references. This effectively means that the values of the imported variables are still referencing the values of the array parameter. You can use this flag on its own or combine it with any other flag by OR'ing the flags.

Filesystem Functions

string basename (string \$path [, string \$suffix])	Given a string containing the path to a file or directory, this function will return the trailing name component.	Returns the base name of the given path.
bool chgrp (string \$filename , mixed \$group)	Attempts to change the group of the file filename to group. Only the superuser may change the group of a file arbitrarily; other users may change the group of a file to any group of which that user is a member.	Returns TRUE on success or FALSE on failure.
bool chmod (string \$filename , int \$mode)	Attempts to change the mode of the specified file to that given in mode.	Returns TRUE on success or FALSE on failure.
bool chown (string \$filename , mixed \$user)	Attempts to change the owner of the file filename to user user. Only the superuser may change the owner of a file.	Returns TRUE on success or FALSE on failure.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 20 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Filesystem Functions (cont)

<pre>void clearstat- cache ([bool \$clear- _realp- ath- _cache = false [, string \$filename])</pre>	<p>When you use <code>stat()</code>, <code>lstat()</code>, or any of the other functions listed in the affected functions list (below), PHP caches the information those functions return in order to provide faster performance. However, in certain cases, you may want to clear the cached information. For instance, if the same file is being checked multiple times within a single script, and that file is in danger of being removed or changed during that script's operation, you may elect to clear the status cache. In these cases, you can use the <code>clearstatcache()</code> function to clear the information that PHP caches about a file. You should also note that PHP doesn't cache information about non-existent files. So, if you call <code>file_exists()</code> on a file that doesn't exist, it will return <code>FALSE</code> until you create the file. If you create the file, it will return <code>TRUE</code> even if you then delete the file. However <code>unlink()</code> clears the cache automatically.</p> <p>Note: This function caches information about specific filenames, so you only need to call <code>clearstatcache()</code> if you are performing multiple operations on the same filename and require the information about that particular file to not be cached.</p>	<p>No value is returned.</p>
<pre>bool copy (string \$source , string \$dest [, resource \$context])</pre>	<p>Makes a copy of the file source to dest. If you wish to move a file, use the <code>rename()</code> function.</p>	<p>Returns <code>TRUE</code> on success or <code>FALSE</code> on failure.</p>



By **sanis**
cheatography.com/sanis/

Not published yet.
 Last updated 27th August, 2016.
 Page 21 of 24.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

Filesystem Functions (cont)

delete	This is a dummy manual entry to satisfy those people who are looking for unlink() or unset() in the wrong place.	
string dirname (string \$path [, int \$levels = 1])	Given a string containing the path of a file or directory, this function will return the parent directory's path that is levels up from the current directory.	Returns the path of a parent directory. If there are no slashes in path, a dot ('.') is returned, indicating the current directory. Otherwise, the returned string is path with any trailing /component removed.
float disk_free- _space (string \$directory)	Given a string containing a directory, this function will return the number of bytes available on the corresponding filesystem or disk partition.	Returns the number of available bytes as a float or FALSE on failure.
float disk_tota- l_space (string \$directory)	Given a string containing a directory, this function will return the total number of bytes on the corresponding filesystem or disk partition.	Returns the total number of bytes as a float or FALSE on failure.
diskfreespace	This function is an alias of: disk_free_space().	
bool fclose (resource \$handle)	The file pointed to by handle is closed.	Returns TRUE on success or FALSE on failure.
bool feof (resource \$handle)	Tests for end-of-file on a file pointer.	Returns TRUE if the file pointer is at EOF or an error occurs (including socket timeout); otherwise returns FALSE.
bool fflush (resource \$handle)	This function forces a write of all buffered output to the resource pointed to by the file handle.	Returns TRUE on success or FALSE on failure.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 22 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Filesystem Functions (cont)

<code>string fgetc (resource \$handle)</code>	Gets a character from the given file pointer.	Returns a string containing a single character read from the file pointed to by handle. Returns FALSE on EOF. Warning This function may return Boolean FALSE, but may also return a non-Boolean value which evaluates to FALSE. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.
<code>array fgetcsv (resource \$handle [, int \$length = 0 [, string \$delimiter = "," [, string \$enclosure = "" [, string \$escape = "\"]]]])</code>	Similar to fgets() except that fgetcsv() parses the line it reads for fields in CSV format and returns an array containing the fields read.	Returns an indexed array containing the fields read. Note: A blank line in a CSV file will be returned as an array comprising a single null field, and will not be treated as an error. Note: If PHP is not properly recognizing the line endings when reading files either on or created by a Macintosh computer, enabling the auto_detect_line_endings run-time configuration option may help resolve the problem. fgetcsv() returns NULL if an invalid handle is supplied or FALSE on other errors, including end of file.
<code>string fgets (resource \$handle [, int \$length])</code>	Gets a line from file pointer.	Returns a string of up to length - 1 bytes read from the file pointed to by handle. If there is no more data to read in the file pointer, then FALSE is returned. If an error occurs, FALSE is returned.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 23 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Filesystem Functions (cont)

string <code>fgetss (resource \$handle [, int \$length [, string \$allowable_tags]])</code>	Identical to <code>fgets()</code> , except that <code>fgetss()</code> attempts to strip any NUL bytes, HTML and PHP tags from the text it reads.	Returns a string of up to <code>length - 1</code> bytes read from the file pointed to by <code>handle</code> , with all HTML and PHP code stripped. If an error occurs, returns <code>FALSE</code> .
bool <code>file_exists (string \$filename)</code>	Checks whether a file or directory exists.	Returns <code>TRUE</code> if the file or directory specified by <code>filename</code> exists; <code>FALSE</code> otherwise. Note: This function will return <code>FALSE</code> for symlinks pointing to non-existing files.
string <code>file_get_contents (string \$filename [, bool \$use_include_path = false [, resource \$context [, int \$offset = 0 [, int \$maxlen]]]])</code>	This function is similar to <code>file()</code> , except that <code>file_get_contents()</code> returns the file in a string, starting at the specified offset up to <code>maxlength</code> bytes. On failure, <code>file_get_contents()</code> will return <code>FALSE</code> . <code>file_get_contents()</code> is the preferred way to read the contents of a file into a string. It will use memory mapping techniques if supported by your OS to enhance performance. Note: If you're opening a URI with special characters, such as spaces, you need to encode the URI with <code>urlencode()</code> .	The function returns the read data or <code>FALSE</code> on failure. Warning This function may return Boolean <code>FALSE</code> , but may also return a non-Boolean value which evaluates to <code>FALSE</code> . Please read the section on Booleans for more information. Use the <code>===</code> operator for testing the return value of this function. An <code>E_WARNING</code> level error is generated if <code>filename</code> cannot be found, <code>maxlength</code> is less than zero, or if seeking to the specified offset in the stream fails.
int <code>file_put_contents (string \$filename , mixed \$data [, int \$flags = 0 [, resource \$context]])</code>	This function is identical to calling <code>fopen()</code> , <code>fwrite()</code> and <code>fclose()</code> successively to write data to a file. If <code>filename</code> does not exist, the file is created. Otherwise, the existing file is overwritten, unless the <code>FILE_APPEND</code> flag is set.	This function returns the number of bytes that were written to the file, or <code>FALSE</code> on failure. Warning This function may return Boolean <code>FALSE</code> , but may also return a non-Boolean value which evaluates to <code>FALSE</code> . Please read the section on Booleans for more information. Use the <code>===</code> operator for testing the return value of this function.



By **sanis**
cheatography.com/sanis/

Not published yet.
Last updated 27th August, 2016.
Page 24 of 24.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>