Arrays

http://overapi.com/php/

Array

Array Functions

- array()
- array_change_key_case()
- array_chunk()
- array_combine()
- array_count_values()
- array_diff()
- array_diff_assoc()
- array_diff_key()
- array_diff_uassoc()
- array_diff_ukey()
- array_fill()
- array_filter()
- array_flip()
- array_intersect()
- array_intersect_assoc()
- array_intersect_key()
- array_intersect_uassoc()
- array_intersect_ukey()
- array_key_exists()
- array_keys()
- array_map()
- array_merge()
- array_merge_recursive()
- array_multisort()
- array_pad()
- array_pop()
- array_product()
- array_push()
- array_rand()
- array_reduce()
- array_reverse()
- array_search()
- array_shift()
- array_slice()
- array_splice()
- array_sum()
- array_udiff()
- array_udiff_assoc()
- array_udiff_uassoc()
- array_uintersect()
- array_uintersect_assoc()
- array_uintersect_uassoc()
- array_unique()
- array_unshift()
- array_values()

- array_walk()
- array_walk_recursive()
- arsort()
- asort()
- compact()
- count()
- current()
- each()
- end()
- extract()
- in_array()
- key()
- krsort()
- ksort()
- list()
- natcasesort()
- natsort()
- next()
- pos()
- prev()
- range()
- reset()
- rsort()
- shuffle()
- sizeof()
- sort()
- uasort()
- uksort()
- usort()

http://php.net/manual/en/ref.array.php

Table of Contents ¶

- array_change_key_case — Changes the case of all keys in an array
- array_chunk — Split an array into chunks
- array_column — Return the values from a single column in the input array
- array_combine — Creates an array by using one array for keys and another for its values
- array_count_values — Counts all the values of an array
- array_diff_assoc — Computes the difference of arrays with additional index check
- array_diff_key — Computes the difference of arrays using keys for comparison
- array_diff_uassoc — Computes the difference of arrays with additional index check which is performed by a user supplied callback function
- array_diff_ukey — Computes the difference of arrays using a callback function on the keys for comparison

- array_diff — Computes the difference of arrays
- array_fill_keys — Fill an array with values, specifying keys
- array_fill — Fill an array with values
- array_filter — Filters elements of an array using a callback function
- array_flip — Exchanges all keys with their associated values in an array
- array_intersect_assoc — Computes the intersection of arrays with additional index check
- array_intersect_key — Computes the intersection of arrays using keys for comparison
- array_intersect_uassoc — Computes the intersection of arrays with additional index check, compares indexes by a callback function
- array_intersect_ukey — Computes the intersection of arrays using a callback function on the keys for comparison
- array_intersect — Computes the intersection of arrays
- array_key_exists — Checks if the given key or index exists in the array
- array_keys — Return all the keys or a subset of the keys of an array
- array_map — Applies the callback to the elements of the given arrays
- array_merge_recursive — Merge two or more arrays recursively
- array_merge — Merge one or more arrays
- array_multisort — Sort multiple or multi-dimensional arrays
- array_pad — Pad array to the specified length with a value
- array_pop — Pop the element off the end of array
- array_product — Calculate the product of values in an array
- array_push — Push one or more elements onto the end of array
- array_rand — Pick one or more random entries out of an array
- array_reduce — Iteratively reduce the array to a single value using a callback function
- array_replace_recursive — Replaces elements from passed arrays into the first array recursively
- array_replace — Replaces elements from passed arrays into the first array
- array_reverse — Return an array with elements in reverse order
- array_search — Searches the array for a given value and returns the corresponding key if successful
- array_shift — Shift an element off the beginning of array
- array_slice — Extract a slice of the array
- array_splice — Remove a portion of the array and replace it with something else
- array_sum — Calculate the sum of values in an array
- array_udiff_assoc — Computes the difference of arrays with additional index check, compares data by a callback function

- array_udiff_uassoc — Computes the difference of arrays with additional index check, compares data and indexes by a callback function
- array_udiff — Computes the difference of arrays by using a callback function for data comparison
- array_uintersect_assoc — Computes the intersection of arrays with additional index check, compares data by a callback function
- array_uintersect_uassoc — Computes the intersection of arrays with additional index check, compares data and indexes by separate callback functions
- array_uintersect — Computes the intersection of arrays, compares data by a callback function
- array_unique — Removes duplicate values from an array
- array_unshift — Prepend one or more elements to the beginning of an array
- array_values — Return all the values of an array
- array_walk_recursive — Apply a user function recursively to every member of an array
- array_walk — Apply a user supplied function to every member of an array
- array — Create an array
- arsort — Sort an array in reverse order and maintain index association
- asort — Sort an array and maintain index association
- compact — Create array containing variables and their values
- count — Count all elements in an array, or something in an object
- current — Return the current element in an array
- each — Return the current key and value pair from an array and advance the array cursor
- end — Set the internal pointer of an array to its last element
- extract — Import variables into the current symbol table from an array
- in_array — Checks if a value exists in an array
- key_exists — Alias of array_key_exists
- key — Fetch a key from an array
- krsort — Sort an array by key in reverse order
- ksort — Sort an array by key
- list — Assign variables as if they were an array
- natcasesort — Sort an array using a case insensitive "natural order" algorithm
- natsort — Sort an array using a "natural order" algorithm
- next — Advance the internal array pointer of an array
- pos — Alias of current
- prev — Rewind the internal array pointer
- range — Create an array containing a range of elements
- reset — Set the internal pointer of an array to its first element

- rsort — Sort an array in reverse order
- shuffle — Shuffle an array
- sizeof — Alias of count
- sort — Sort an array
- uasort — Sort an array with a user-defined comparison function and maintain index association
- uksort — Sort an array by keys using a user-defined comparison function
- usort — Sort an array by values using a user-defined comparison function