**PHP ARRAY FUNCTION REFERENCES**

|  |  |  |
| --- | --- | --- |
| **FUNCTION** | **DESCRIPTION** | **SYNTAX** |
| [*array()*](https://www.w3schools.com/php/func_array.asp) | Creates an array | *array(value1, value2, value3, etc.)*  *array(key=>value,key=>value,key=>value,etc.)* |
| [*array\_change\_key\_case()*](https://www.w3schools.com/php/func_array_change_key_case.asp) | Changes all keys in an array to lowercase or uppercase | *array\_change\_key\_case(*array*,* case*)* |
| [*array\_chunk()*](https://www.w3schools.com/php/func_array_chunk.asp) | Splits an array into chunks of arrays | *array\_chunk(*array*,* size*,* preserve\_key*)* |
| [*array\_column()*](https://www.w3schools.com/php/func_array_column.asp) | Returns the values from a single column in the input array | *array\_column(*array*,* column\_key*,* index\_key*)* |
| [*array\_combine()*](https://www.w3schools.com/php/func_array_combine.asp) | Creates an array by using the elements from one "keys" array and one "values" array | *array\_combine(*keys*,* values*)* |
| [*array\_count\_values()*](https://www.w3schools.com/php/func_array_count_values.asp) | Counts all the values of an array | *array\_count\_values(*array*)* |
| [*array\_diff()*](https://www.w3schools.com/php/func_array_diff.asp) | Compare arrays, and returns the differences (compare values only) | *array\_diff(*array1, array2, array3, ...*)* |
| [*array\_diff\_assoc()*](https://www.w3schools.com/php/func_array_diff_assoc.asp) | Compare arrays, and returns the differences (compare keys and values) | *array\_diff\_assoc(*array1,array2,array3...*)* |
| [*array\_diff\_key()*](https://www.w3schools.com/php/func_array_diff_key.asp) | Compare arrays, and returns the differences (compare keys only) | *array\_diff\_key(*array1, array2, array3, ...*)* |
| [*array\_diff\_uassoc()*](https://www.w3schools.com/php/func_array_diff_uassoc.asp) | Compare arrays, and returns the differences (compare keys and values, using a user-defined key comparison function) | *array\_diff\_uassoc(*array1, array2, array3, ..., myfunction*)* |
| [*array\_diff\_ukey()*](https://www.w3schools.com/php/func_array_diff_ukey.asp) | Compare arrays, and returns the differences (compare keys only, using a user-defined key comparison function) | *array\_diff\_ukey(*array1, array2, array3, ..., myfunction*)* |
| [*array\_fill()*](https://www.w3schools.com/php/func_array_fill.asp) | Fills an array with values | *array\_fill(*index, number, value*)* |
| [*array\_fill\_keys()*](https://www.w3schools.com/php/func_array_fill_keys.asp) | Fills an array with values, specifying keys | *array\_fill\_keys(*keys, value*)* |
| [*array\_filter()*](https://www.w3schools.com/php/func_array_filter.asp) | Filters the values of an array using a callback function | *array\_filter(*array, callbackfunction, flag*)* |
| [*array\_flip()*](https://www.w3schools.com/php/func_array_flip.asp) | Flips/Exchanges all keys with their associated values in an array | *array\_flip(*array*)* |
| [*array\_intersect()*](https://www.w3schools.com/php/func_array_intersect.asp) | Compare arrays, and returns the matches (compare values only) | *array\_intersect(*array1, array2, array3, ...*)* |
| [*array\_intersect\_assoc()*](https://www.w3schools.com/php/func_array_intersect_assoc.asp) | Compare arrays and returns the matches (compare keys and values) | *array\_intersect\_assoc(*array1,array2,array3, ...*)* |
| [*array\_intersect\_key()*](https://www.w3schools.com/php/func_array_intersect_key.asp) | Compare arrays, and returns the matches (compare keys only) | *array\_intersect\_key(*array1, array2, array3, ...*)* |
| [*array\_intersect\_uassoc()*](https://www.w3schools.com/php/func_array_intersect_uassoc.asp) | Compare arrays, and returns the matches (compare keys and values, using a user-defined key comparison function) | *array\_intersect\_uassoc(*array1, array2, array3, ..., myfunction*)* |
| [*array\_intersect\_ukey()*](https://www.w3schools.com/php/func_array_intersect_ukey.asp) | Compare arrays, and returns the matches (compare keys only, using a user-defined key comparison function) | *array\_intersect\_ukey(*array1, array2, array3, ..., myfunction*)* |
| [*array\_key\_exists()*](https://www.w3schools.com/php/func_array_key_exists.asp) | Checks if the specified key exists in the array | *array\_key\_exists(*key, array*)* |
| [*array\_keys()*](https://www.w3schools.com/php/func_array_keys.asp) | Returns all the keys of an array | *array\_keys(*array, value, strict*)* |
| [*array\_map()*](https://www.w3schools.com/php/func_array_map.asp) | Sends each value of an array to a user-made function, which returns new values | *array\_map(*myfunction, array1, array2, array3, *...)* |
| [*array\_merge()*](https://www.w3schools.com/php/func_array_merge.asp) | Merges one or more arrays into one array | *array\_merge(*array1, array2, array3, ...*)* |
| [*array\_merge\_recursive()*](https://www.w3schools.com/php/func_array_merge_recursive.asp) | Merges one or more arrays into one array recursively | *array\_merge\_recursive(*array1, array2, array3, ...*)* |
| [*array\_multisort()*](https://www.w3schools.com/php/func_array_multisort.asp) | Sorts multiple or multi-dimensional arrays | *array\_multisort(*array1, sortorder, sorttype, array2, array3, ...*)* |
| [*array\_pad()*](https://www.w3schools.com/php/func_array_pad.asp) | Inserts a specified number of items, with a specified value, to an array | *array\_pad(*array, size, value*)* |
| [*array\_pop()*](https://www.w3schools.com/php/func_array_pop.asp) | Deletes the last element of an array | *array\_pop(*array*)* |
| [*array\_product()*](https://www.w3schools.com/php/func_array_product.asp) | Calculates the product of the values in an array | *array\_product(*array*)* |
| [*array\_push()*](https://www.w3schools.com/php/func_array_push.asp) | Inserts one or more elements to the end of an array | *array\_push(*array, value1, value2, ...*)* |
| [*array\_rand()*](https://www.w3schools.com/php/func_array_rand.asp) | Returns one or more random keys from an array | *array\_rand(*array, number*)* |
| [*array\_reduce()*](https://www.w3schools.com/php/func_array_reduce.asp) | Returns an array as a string, using a user-defined function | *array\_reduce(*array, myfunction, initial*)* |
| [*array\_replace()*](https://www.w3schools.com/php/func_array_replace.asp) | Replaces the values of the first array with the values from following arrays | *array\_replace(*array1, array2, array3, ...*)* |
| [*array\_replace\_recursive()*](https://www.w3schools.com/php/func_array_replace_recursive.asp) | Replaces the values of the first array with the values from following arrays recursively | *array\_replace\_recursive(*array1, array2, array3, ...*)* |
| [*array\_reverse()*](https://www.w3schools.com/php/func_array_reverse.asp) | Returns an array in the reverse order | *array\_reverse(*array, preserve*)* |
| [*array\_search()*](https://www.w3schools.com/php/func_array_search.asp) | Searches an array for a given value and returns the key | *array\_search(*value, array, strict*)* |
| [*array\_shift()*](https://www.w3schools.com/php/func_array_shift.asp) | Removes the first element from an array, and returns the value of the removed element | *array\_shift(*array*)* |
| [*array\_slice()*](https://www.w3schools.com/php/func_array_slice.asp) | Returns selected parts of an array | *array\_slice(*array, start, length, preserve*)* |
| [*array\_splice()*](https://www.w3schools.com/php/func_array_splice.asp) | Removes and replaces specified elements of an array | *array\_splice(*array, start, length, array*)* |
| [*array\_sum()*](https://www.w3schools.com/php/func_array_sum.asp) | Returns the sum of the values in an array | *array\_sum(*array*)* |
| [*array\_udiff()*](https://www.w3schools.com/php/func_array_udiff.asp) | Compare arrays, and returns the differences (compare values only, using a user-defined key comparison function) | *array\_udiff(*array1, array2, array3, ..., myfunction*)* |
| [*array\_udiff\_assoc()*](https://www.w3schools.com/php/func_array_udiff_assoc.asp) | Compare arrays, and returns the differences (compare keys and values, using a built-in function to compare the keys and a user-defined function to compare the values) | *array\_udiff\_assoc(*array1, array2, array3, ..., myfunction*)* |
| [*array\_udiff\_uassoc()*](https://www.w3schools.com/php/func_array_udiff_uassoc.asp) | Compare arrays, and returns the differences (compare keys and values, using two user-defined key comparison functions) | *array\_udiff\_uassoc(*array1, array2, array3, ..., myfunc\_key, myfunc\_value*)* |
| [*array\_uintersect()*](https://www.w3schools.com/php/func_array_uintersect.asp) | Compare arrays, and returns the matches (compare values only, using a user-defined key comparison function) | *array\_uintersect(*array1, array2, array3, ..., myfunction*)* |
| *[array\_uintersect\_assoc()](https://www.w3schools.com/php/func_array_uintersect_assoc.asp)* | Compare arrays, and returns the matches (compare keys and values, using a built-in function to compare the keys and a user-defined function to compare the values) | *array\_uintersect\_assoc(*array1, array2, array3, ..., myfunction*)* |
| [*array\_uintersect\_uassoc()*](https://www.w3schools.com/php/func_array_uintersect_uassoc.asp) | Compare arrays, and returns the matches (compare keys and values, using two user-defined key comparison functions) | *array\_uintersect\_uassoc(*array1, array2, array3, ..., myfunc\_key, myfunc\_value*)* |
| [*array\_unique()*](https://www.w3schools.com/php/func_array_unique.asp) | Removes duplicate values from an array | *array\_unique(*array, sorttype*)* |
| [*array\_unshift()*](https://www.w3schools.com/php/func_array_unshift.asp) | Adds one or more elements to the beginning of an array | *array\_unshift(*array, value1, value2, value3, ...*)* |
| [*array\_values()*](https://www.w3schools.com/php/func_array_values.asp) | Returns all the values of an array | *array\_values(*array*)* |
| [*array\_walk()*](https://www.w3schools.com/php/func_array_walk.asp) | Applies a user function to every member of an array | *array\_walk(*array, myfunction, parameter...*)* |
| [*array\_walk\_recursive()*](https://www.w3schools.com/php/func_array_walk_recursive.asp) | Applies a user function recursively to every member of an array | *array\_walk\_recursive(*array, myfunction, parameter...*)* |
| [*arsort()*](https://www.w3schools.com/php/func_array_arsort.asp) | Sorts an associative array in descending order, according to the value | *arsort(*array, sorttype*)* |
| [*asort()*](https://www.w3schools.com/php/func_array_asort.asp) | Sorts an associative array in ascending order, according to the value | *asort(*array, sorttype) |
| [*compact()*](https://www.w3schools.com/php/func_array_compact.asp) | Create array containing variables and their values | *compact(*var1, var2...*)* |
| [*count()*](https://www.w3schools.com/php/func_array_count.asp) | Returns the number of elements in an array | *count(*array, mode*)* |
| [*current()*](https://www.w3schools.com/php/func_array_current.asp) | Returns the current element in an array | *current(*array*)* |
| [*each()*](https://www.w3schools.com/php/func_array_each.asp) | Deprecated from PHP 7.2. Returns the current key and value pair from an array | *Deprecated from PHP 7.2.* |
| [*end()*](https://www.w3schools.com/php/func_array_end.asp) | Sets the internal pointer of an array to its last element | *end(*array*)* |
| [*extract()*](https://www.w3schools.com/php/func_array_extract.asp) | Imports variables into the current symbol table from an array | *extract(*array, extract\_rules, prefix*)* |
| [*in\_array()*](https://www.w3schools.com/php/func_array_in_array.asp) | Checks if a specified value exists in an array | *in\_array(*search, array, type*)* |
| [*key()*](https://www.w3schools.com/php/func_array_key.asp) | Fetches a key from an array | *key(*array*)* |
| [*krsort()*](https://www.w3schools.com/php/func_array_krsort.asp) | Sorts an associative array in descending order, according to the key | *krsort(*array, sorttype*)* |
| [*ksort()*](https://www.w3schools.com/php/func_array_ksort.asp) | Sorts an associative array in ascending order, according to the key | *ksort(*array, sorttype*)* |
| [*list()*](https://www.w3schools.com/php/func_array_list.asp) | Assigns variables as if they were an array | *list(*var1, var2, ...*)* |
| [*natcasesort()*](https://www.w3schools.com/php/func_array_natcasesort.asp) | Sorts an array using a case insensitive "natural order" algorithm | *natcasesort(array)* |
| [*natsort()*](https://www.w3schools.com/php/func_array_natsort.asp) | Sorts an array using a "natural order" algorithm | *natsort(*array*)* |
| [*next()*](https://www.w3schools.com/php/func_array_next.asp) | Advance the internal array pointer of an array | *next(*array*)* |
| [*pos()*](https://www.w3schools.com/php/func_array_pos.asp) | Alias of [current()](https://www.w3schools.com/php/func_array_current.asp) | *pos(*array*)* |
| [*prev()*](https://www.w3schools.com/php/func_array_prev.asp) | Rewinds the internal array pointer | *prev(*array*)* |
| [*range()*](https://www.w3schools.com/php/func_array_range.asp) | Creates an array containing a range of elements | *range(*low, high, step*)* |
| [*reset()*](https://www.w3schools.com/php/func_array_reset.asp) | Sets the internal pointer of an array to its first element | *reset(*array*)* |
| [*rsort()*](https://www.w3schools.com/php/func_array_rsort.asp) | Sorts an indexed array in descending order | *rsort(*array, sorttype*)* |
| [*shuffle()*](https://www.w3schools.com/php/func_array_shuffle.asp) | Shuffles an array | *shuffle(*array*)* |
| [*sizeof()*](https://www.w3schools.com/php/func_array_sizeof.asp) | Alias of [count()](https://www.w3schools.com/php/func_array_count.asp) | *sizeof(*array, mode*)* |
| [*sort()*](https://www.w3schools.com/php/func_array_sort.asp) | Sorts an indexed array in ascending order | *sort(*array, sorttype*)* |
| [*uasort()*](https://www.w3schools.com/php/func_array_uasort.asp) | Sorts an array by values using a user-defined comparison function and maintains the index association | *uasort(*array, callback*)* |
| [*uksort()*](https://www.w3schools.com/php/func_array_uksort.asp) | Sorts an array by keys using a user-defined comparison function | *uksort(*array, callback*)* |
| [*usort()*](https://www.w3schools.com/php/func_array_usort.asp) | Sorts an array by values using a user-defined comparison function | *usort(*array, callback*)* |