1. **array\_chunk**  — Split an array into chunks
2. **array\_combine** — Creates an array by using one array for keys and another for its values
3. **array\_merge** — Merge one or more arrays
4. **array\_count\_values** — Counts all the values of an array
5. **array\_intersect**  — Computes the intersection of arrays
6. **array\_key\_exists** — Checks if the given key or index exists in the array
7. **array\_keys** — Return all the keys or a subset of the keys of an array
8. **array\_values** — Return all the values of an array
9. **array\_unique** — Removes duplicate values from an array
10. **array\_flip** — Exchanges all keys with their associated values in an arra**y**
11. **array\_multisort** — Sort multiple or multi-dimensional arrays
12. **array\_sum**  — Calculate the sum of values in an array
13. **array\_slice**  — Extract a slice of the array
14. **array\_splice**  — Remove a portion of the array and replace it with something else
15. **array\_pad** — Pad array to the specified length with a value
16. **array\_push** — Push one or more elements onto the end of array
17. **array\_pop** — Pop the element off the end of array
18. **array\_shift**  — Shift an element off the beginning of array
19. **array\_unshift** — Prepend one or more elements to the beginning of an array
20. **array\_rand** — Pick one or more random entries out of an array
21. **array\_replace** — Replaces elements from passed arrays into the first array
22. **array\_reverse** — Return an array with elements in reverse order
23. **array\_search** — Searches the array for a given value and returns the corresponding key if successful
24. **array\_walk**  — Apply a user function to every member of an array
25. **array**  — Create an array
26. **compact**  — Create array containing variables and their values
27. **count** — Count all elements in an array, or something in an object
28. **sizeof**  — Alias of count
29. **extract**  — Import variables into the current symbol table from an array
30. **in\_array**  — Checks if a value exists in an array
31. **key**  — Fetch a key from an array
32. **list**  — Assign variables as if they were an array
33. **each**  — Return the current key and value pair from an array and advance the array cursor
34. **range**  — Create an array containing a range of elements
35. **reset**  — Set the internal pointer of an array to its first element
36. **shuffle**  — Shuffle an array
37. **sort**  — Sort an array
38. **rsort**  — Sort an array in reverse order
39. **asort**  — Sort an array and maintain index association
40. **arsort** — Sort an array in reverse order and maintain index association
41. **krsort** — Sort an array by key in reverse order
42. **ksort**  — Sort an array by key
43. **usort**  — Sort an array by values using a user-defined comparison function
44. **uasort**  — Sort an array with a user-defined comparison function and maintain index association
45. **uksort**  — Sort an array by keys using a user-defined comparison function
46. **natsort** — Sort an array using a "natural order" algorithm
47. **natcasesort**  — Sort an array using a case insensitive "natural order" algorithm