Array BuiltIn

1. array_chunk

Splits an array into chunks.

```
$friends = ["Osama", "Ahmed", "Sameh", "Mahmoud"];
print_r(array_chunk($friends, 2));  // Splits into chunks of 2
```

2. array_change_key_case

Changes the case of all array keys.

```
$countries = ["EG" => "Egypt", "KSA" => "Saudi Arabia"];
print_r(array_change_key_case($countries, CASE_UPPER));  // Changes keys to uppercase
```

3. array_combine

Combines two arrays where one is keys, and the other is values.

```
$keys = ["A", "B"];
$values = ["Apple", "Banana"];
print_r(array_combine($keys, $values));  // ["A" => "Apple", "B" => "Banana"]
```

4. array_count_values

Counts occurrences of values in an array.

```
$fruits = ["Apple", "Banana", "Apple"];
print_r(array_count_values($fruits));  // ["Apple" => 2, "Banana" => 1]
```

5. array_reverse

Reverses the order of array elements.

```
$numbers = [1, 2, 3];
print_r(array_reverse($numbers));  // [3, 2, 1]
```

6. array_flip

Swaps keys and values.

```
$countries = ["EG" => "Egypt", "KSA" => "Saudi Arabia"];
print_r(array_flip($countries));  // ["Egypt" => "EG", "Saudi Arabia" => "KSA"]
```

7. count

Counts elements in an array.

```
$items = ["Apple", "Banana", ["Orange", "Grape"]];
echo count($items);  // 3
```

8. in_array

Checks if a value exists in an array.

```
$fruits = ["Apple", "Banana"];
echo in_array("Apple", $fruits) ? "Found" : "Not Found";  // Found
```

9. array_key_exists

Checks if a key exists in an array.

```
$courses = ["PHP" => 100, "JavaScript" => 95];
echo array_key_exists("PHP", $courses) ? "Exists" : "Not Found";  // Exists
```

10. array_merge

Merges multiple arrays.

```


$array1 = ["A" => "Apple"];
$array2 = ["B" => "Banana"];
print_r(array_merge($array1, $array2));  // ["A" => "Apple", "B" => "Banana"]
```

11. array_push / array_pop

Adds or removes elements at the end of an array.

```
$colors = ["Red", "Green"];
array_push($colors, "Blue");  // Adds "Blue"
print_r($colors);  // ["Red", "Green", "Blue"]
array_pop($colors);  // Removes last element "Blue"
print_r($colors);  // ["Red", "Green"]
```

12. array_shift / array_unshift

Removes or adds elements at the start of an array.

```
$colors = ["Red", "Green"];
array_unshift($colors, "Blue");  // Adds "Blue" at the start
print_r($colors);  // ["Blue", "Red", "Green"]
array_shift($colors);  // Removes the first element "Blue"
print_r($colors);  // ["Red", "Green"]
```

13. array_slice

Extracts a portion of an array.

```
$letters = ["A", "B", "C", "D"];
print_r(array_slice($letters, 1, 2));  // ["B", "C"]
```

14. array_replace

Replaces values in the first array with values from subsequent arrays.

```
$array1 = ["A" => "Apple", "B" => "Banana"];
$array2 = ["A" => "Apricot"];
print_r(array_replace($array1, $array2));  // ["A" => "Apricot", "B" => "Banana"]
```

15. array_map

Applies a callback function to each element in an array.

```
$numbers = [1, 2, 3, 4];
print_r(array_map(function($n) { return $n * 2; }, $numbers));
// [2, 4, 6, 8]
```

16. array_filter

Filters elements of an array using a callback function.

```
$numbers = [1, 2, 3, 4];
print_r(array_filter($numbers, function($n) { return $n % 2 == 0; }));
// [2, 4]
```

17. array_walk

Applies a user-defined function to every element of the array.

```
$numbers = [1, 2, 3];
array_walk($numbers, function(&$item) { $item = $item * 2;
});
print_r($numbers);  // [2, 4, 6]
```

18. array_reduce

Iteratively reduces an array to a single value using a callback function.

```
$numbers = [1, 2, 3, 4];
echo array_reduce($numbers, function($carry, $item) { return
$carry + $item; });
// 10
```

19. array_sum

Calculates the sum of the values in an array.

```
$numbers = [1, 2, 3];
echo array_sum($numbers);  // 6
```

20. array_diff

Compares two arrays and returns the values from the first array that are not in the second.

```
$array1 = [1, 2, 3];
$array2 = [2, 3, 4];
print_r(array_diff($array1, $array2));  // [1]
```

21. array_intersect

Compares two arrays and returns the common values.

```
$array1 = [1, 2, 3];
$array2 = [2, 3, 4];
```

```
print_r(array_intersect($array1, $array2));  // [2, 3]
```

22. array_merge_recursive

Merges multiple arrays recursively.

```
$array1 = ["a" => "Apple", "b" => "Banana"];
$array2 = ["a" => "Apricot", "b" => "Blueberry"];
print_r(array_merge_recursive($array1, $array2));
// ["a" => ["Apple", "Apricot"], "b" => ["Banana", "Blueberr
y"]]
```

23. array_unique

Removes duplicate values from an array.

```
$numbers = [1, 2, 2, 3, 3, 4];
print_r(array_unique($numbers));  // [1, 2, 3, 4]
```

24. array_values

Returns all the values from an array without the keys.

```
$person = ["name" => "John", "age" => 30];
print_r(array_values($person));  // ["John", 30]
```

25. array_splice

Removes a portion of an array and optionally replaces it with new elements.

```
$fruits = ["Apple", "Banana", "Cherry", "Date"];
array_splice($fruits, 1, 2, ["Grapes", "Kiwi"]);
print_r($fruits);  // ["Apple", "Grapes", "Kiwi", "Date"]
```

26. array_rand

Selects one or more random keys from an array.

```
$colors = ["Red", "Green", "Blue", "Yellow"];
$randomKey = array_rand($colors);
echo $colors[$randomKey];  // Random color
```

27. array_slice

Extracts a portion of an array without modifying the original array.

```
$fruits = ["Apple", "Banana", "Cherry", "Date"];
print_r(array_slice($fruits, 1, 2));  // ["Banana", "Cherry"]
```

28. array_key_first / array_key_last

Gets the first or last key of an array.

```
$array = ["first" => "Apple", "second" => "Banana"];
echo array_key_first($array);  // "first"
echo array_key_last($array);   // "second"
```

29. array_flip

Flips all keys and values of an array.

```
$array = ["a" => "apple", "b" => "banana"];
print_r(array_flip($array));  // ["apple" => "a", "banana" => "b"]
```

30. array_search

Searches for a value in an array and returns the key if found.

```
$fruits = ["Apple", "Banana", "Cherry"];
$key = array_search("Banana", $fruits);
echo $key;  // 1
```