

# PHP

Arrays:

//function declaration:

```
$my_arr = array(1, 2, 3, "text")
```

shorthand:

```
$my_arr = [1, 2, 3, "text"]
```

//print arr:

1. `print_r($my_arr)`

// prints array in this format:

```
[0] =>1
```

```
[1] =>2
```

```
[2] =>3
```

```
[3] =>text
```

2. `echo implode("?", $my_arr)` // used with an array of strings

"?" separates each index with ?

// prints array in this format:

```
1? 2? 3? text?
```

change array

//add new value at the end:

1. Using `$arr[] = "new value"`

`$my_arr[] = "new index"` // adds "new index" at the end of the array.

2. Using `array_push($arr)`

`array_push($my_arr, "new value1", "new value2")`

//add new value at the front:

```
array_unshift($my_arr, "new value1", "new value2")
```

//change existing index:

```
$my_arr[0] = "new value" // change value at index 0 to "new value".
```

//remove last index:

```
array_pop($arr)
```

//remove first index:

```
array_shift($arr)
```

Associative array

```
$my_Arr = ["value0" => 124, "value1" => "Hello", "value2" => true];
```

// to print an associative array, it's better to use the print\_r(\$my\_Arr) method.

//Add values to associative array

```
$my_Arr["value3"] = 456;
```

//Change value

```
$my_Arr["value2"] = false;
```

//Delete a key => value pair

```
unset($my_Arr["value2"]);
```

```
$assignment_one = ["Alex"=> 87, "Kenny"=> 91, "Natalia"=> 91, "Lily"=> 67, "Dan"=> 81, "Kat"=> 77, "Sara" => 65];
```

```
$assignment_two = ["Alex"=> 91, "Kenny"=> 99, "Natalia"=> 100, "Lily"=> 61, "Dan"=> 88, "Kat"=> 90];
```

```
$assignment_three = ["Alex"=> 78, "Kenny"=> 92, "Natalia"=> 94, "Lily"=> 79, "Dan"=> 73, "Sara" => 61];
```

```
$student_name = "Alex";
```

```
// add Sara's grade to $assignment_two  
$assignment_two["Sara"] = 65;
```

```
// add Kat's grade to $assignment_two  
$assignment_three["Kat"] = 97;
```

```
// collecting Dan's grades in a new array  
$dans_grades =  
[$assignment_one["Dan"],  
$assignment_two["Dan"],  
$assignment_three["Dan"]];
```

```
// printing Alex's grade in assignment two using variable  
echo $assignment_two[$student_name];
```

```
// creating a 4 index array
```

```
$hybrid_array = [1, 2, true, "hello"];
```

```
// adding a value at index 8
```

```
$hybrid_array[8] = "five more";
```

```
// indexes 4 to 7 are empty
```

```
print_r($hybrid_array);
```

```
// pushing a random number into the array
```

```
array_push($hybrid_array, rand());
```

```
// new value is at index 9
```

```
echo $hybrid_array[9];
```

```
// join multiple arrays
```

```
$new_arr = ["1", "2", "3"]
```

```
$new_arr2 = ["4", "5", "6"]
```

```
$joined_arr = $new_arr + $new_arr2
```

```
// prints ["1","2","3"] because both arrays have the same  
number of indexes.
```

```
// join associative arrays
```

```
$new_arr = ["first value" => "1", "second value" => "2", "third  
value" => "3"]
```

```
$new_arr2 = ["value one" => "4", "value two" => "5", "value  
three" => "6"]
```

```
$joined_arr = $new_arr + $new_arr1
```

```
/* prints ["first value" => "1", "second value" => "2", "third value"
=> "3", "value one" => "4", "value two" => "5", "value three" =>
"6"]
```

because the keys don't have the same name, they are considered unique and are added together.

the order of the arrays matter: the values of the first array will be printed first \*/

```
// Assign by value
```

```
$my_arr = [1, 2]
```

```
$copy = $my_arr
```

```
$alias &= $my_arr
```

```
$my_arr[0] = 8;
```

```
echo $my_arr // [8,2]
```

```
echo $copy // [1,2]
```

```
echo $alias // [8,2]
```

```
// Assign by reference
```

```
function changeColor($arr) {
```

```
    $arr["color"] = "red";
```

```
}
```

```
$my_arr = ["color" => "green", "shape" => "square"]
```

```
changeColor($my_arr)
```

```
echo $my_arr["color"] // prints "green"
```

```
function changeColor(&$arr) {  
    $arr["color"] = "red";  
}  
$my_arr = ["color" => "green", "shape" => "square"]  
changeColor($my_arr)  
echo $my_arr["color"] // prints "red"
```

```
// the & operator makes the change permanent
```