

PHP Create Arrays

[Back to ACP page](#)

Table of Contents

- [PHP T](#)
 - [PHP 1](#)
 - [Example 1](#)
 - [Document](#)
 - [Reference](#)

PHP 1 Create Array

```
<h4>Create_Array</h4>
<p>You can create arrays by using the array() function:</p>
<?php
$cars1 = array("Volvo", "BMW", "Toyota");
var_dump($cars1);
?>
<hr>
<p>You can also use a shorter syntax by using the [] brackets:</p>
<?php
$cars2 = ["Volvo", "BMW", "Toyota"];
var_dump($cars2);
?>
```

Create_Array

You can create arrays by using the `array()` function:

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
```

You can also use a shorter syntax by using the `[]` brackets:

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
```

Example 1

Result [View Example](#)

PHP 2 Multiple Lines

```
<pre>

<?php
$cars = [
    "Volvo",
    "BMW",
    "Toyota"
];
var_dump($cars);
?>

</pre>
```

Multiple_Lines

Line breaks are not important, so an array declaration can span multiple lines:

```
    array(3) {
  [0]=>
  string(5) "Volvo"
  [1]=>
  string(3) "BMW"
  [2]=>
  string(6) "Toyota"
}
```

Example 2

Result [View Example](#)

PHP 3 Trailing_Comma

```
<h4>Trailing_Comma</h4>
<p>A comma after the last item is allowed:</p>
<pre>

<?php
$cars = [
    "Volvo",
    "BMW",
    "Toyota",
  ];
var_dump($cars);
?>

</pre>
```

Trailing_Comma

A comma after the last item is allowed:

```
array(3) {  
  [0]=>  
  string(5) "Volvo"  
  [1]=>  
  string(3) "BMW"  
  [2]=>  
  string(6) "Toyota"  
}
```

Example 3

Result [View Example](#)

PHP 4 Array Keys

When creating indexed arrays the keys are given automatically, starting at 0 and increased by 1 for each item, so the array above could also be created with keys:

```
<h4>Array_Keys</h4>  
<pre>  
<?php  
$cars1 = [  
  0 => "Volvo",  
  1 => "BMW",  
  2 => "Toyota"  
];  
  
var_dump($cars1);  
?>  
</pre>  
<hr>  
<p>As you can see, indexed arrays are the same as associative arrays, but associative arrays have names instead of numbers:</p>  
<pre>  
<?php  
$myCar2 = [  
  "brand" => "Ford",  
  "model" => "Mustang",  
  "year" => 1964  
];
```

```
var_dump($myCar2);  
?>  
</pre>
```

Array_Keys

```
array(3) {  
  [0]=>  
  string(5) "Volvo"  
  [1]=>  
  string(3) "BMW"  
  [2]=>  
  string(6) "Toyota"  
}
```

As you can see, indexed arrays are the same as associative arrays, but associative arrays have names instead of numbers:

```
array(3) {  
  ["brand"]=>  
  string(4) "Ford"  
  ["model"]=>  
  string(7) "Mustang"  
  ["year"]=>  
  int(1964)  
}
```

Example 4

Result [View Example](#)

PHP 5 Declare Empty Array

You can declare an empty array first, and add items to it later:

```
<h4>Declare_Empty_Array</h4>  
<pre>  
<?php  
$cars = [];  
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

```
var_dump($cars);
?>
</pre>
<hr>
<p>The same goes for associative arrays, you can declare the array first,
and then add items to it:</p>
<pre>
<?php
$myCar = [];
$myCar["brand"] = "Ford";
$myCar["model"] = "Mustang";
$myCar["year"] = 1964;
var_dump($myCar);
?>
</pre>
```

Declare_Empty_Array

```
array(3) {
  [0]=>
  string(5) "Volvo"
  [1]=>
  string(3) "BMW"
  [2]=>
  string(6) "Toyota"
}
```

The same goes for associative arrays, you can declare the array first, and then add items to it:

```
array(3) {
  ["brand"]=>
  string(4) "Ford"
  ["model"]=>
  string(7) "Mustang"
  ["year"]=>
  int(1964)
}
```

Example 5

Result [View Example](#)

PHP 6 Mixing Array Keys

```
<h4>Mixing_Array_Keys</h4>
<p>You can have arrays with both indexed and named keys:</p>
<pre>
<?php
$myArr = [];
$myArr[0] = "apples";
$myArr[1] = "bananas";
$myArr["fruit"] = "cherries";

var_dump($myArr);
?>
</pre>
```

Mixing_Array_Keys

You can have arrays with both indexed and named keys:

```
array(3) {
  [0]=>
  string(6) "apples"
  [1]=>
  string(7) "bananas"
  ["fruit"]=>
  string(8) "cherries"
}
```

Example 6

Result [View Example](#)

Document

Document in project

You can [Download PDF](#) file.

Reference