

# PHP Arrays

[< Previous](#)[Next >](#)

An array stores multiple values in one single variable:

## Example

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

[Try it Yourself »](#)

## What is an Array?

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is to create an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

## Create an Array in PHP

In PHP, the `array()` function is used to create an array:

```
array();
```

In PHP, there are three types of arrays:

- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys

- **Multidimensional arrays** - Arrays containing one or more arrays
- 

## Get The Length of an Array - The count() Function

The `count()` function is used to return the length (the number of elements) of an array:

### Example

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>
```

Try it Yourself »

---

## Complete PHP Array Reference

For a complete reference of all array functions, go to our complete [PHP Array Reference](#).

The reference contains a brief description, and examples of use, for each function!