



PHP Arrays

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 cars name, for example), storing the cars in single variables could look like:

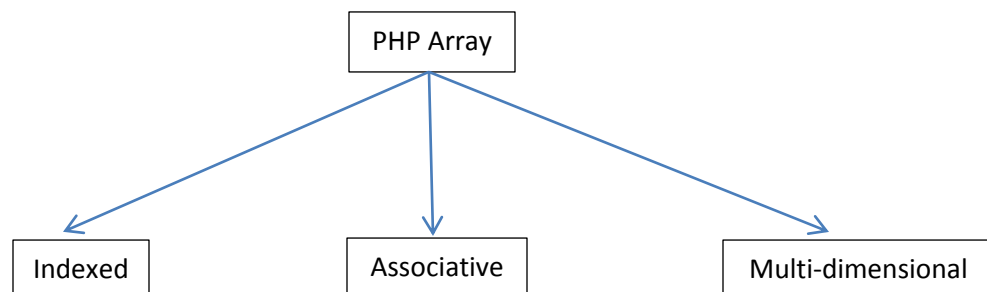
```
<?php
$car1 = "Honda";
$car2 = "Mercedes";
$car3 = "Toyota";
?>
```

However, what if you want to loop through the cars and find a specific one?

And what if you had not only three (3) cars but 500 cars?

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.



- Indexed arrays: Arrays with a numeric index.
- Associative arrays: Arrays with named keys.
- Multi-dimensional arrays: Arrays containing one or more arrays.

PHP Indexed Arrays

There are two ways to create indexed arrays:

1. The index can be assigned automatically (index always starts at 0)

```
<?php
$cars = array("Honda", "Mercedes", "Toyota");
?>
```

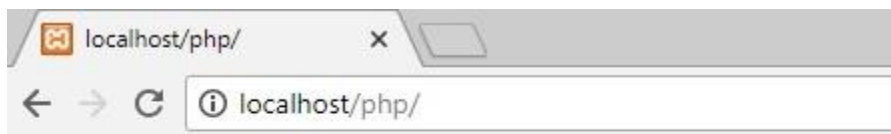
2. The index can be assigned manually.

```
<?php
$car[0] = "Honda";
$car[1] = "Mercedes";
$car[2] = "Toyota";
?>
```

How to 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:

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



3

Loop Through an Indexed Array

In our last two tutorials, we learned about **PHP Loops**. To loop through and print all the values of an indexed array, you can use **for loop**.

```
<?php
    $cars = array("Honda", "Mercedes", "Toyota");
    $arrlen = count($cars);
    for($x=0; $x<$arrlen; $x++){
        echo $cars[$x] . ' <br />';
    }
?>
```



NB: We will learn about the Associative and Multi-dimensional arrays in our next tutorial.