

9. PHP Arrays

An array stores multiple values in one single variable:

Example 1

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

9.1. 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.

9.2. 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 numeric index

- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays

9.3. PHP Indexed Arrays

There are two ways to create indexed arrays:

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

```
$cars=array("Volvo", "BMW", "Toyota");
```

or the index can be assigned manually:

```
$cars[0]="Volvo";  
$cars[1]="BMW";  
$cars[2]="Toyota";
```

The following example creates an indexed array named \$cars, assigns three elements to it, and then prints a text containing the array values:

Example

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

9.4. 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);  
?>
```

9.5. Loop Through an Indexed Array

To loop through and print all the values of an indexed array, you could use a for loop, like this:

Example

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

for ($x=0;$x<$arlength;$x++)
{
    echo $cars[$x];
    echo "<br>";
}
?>
```

9.6. PHP Associative Arrays

Associative arrays are arrays that use named keys that you assign to them.

There are two ways to create an associative array:

```
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
```

or:

```
$age['Peter']="35";
$age['Ben']="37";
$age['Joe']="43";
```

The named keys can then be used in a script:

Example

```
<?php
$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");
```

```
echo "Peter is " . $age['Peter'] . " years  
old.";  
?>
```

9.7. Loop Through an Associative Array

To loop through and print all the values of an associative array, you could use a foreach loop, like this:

Example

```
<?php  
$age=array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
  
foreach($age as $x=>$x_value)  
{  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}  
?>
```

9.8. Multidimensional Arrays

Multidimensional arrays will be explained in the PHP advanced section.

9.9. 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!

