Run »Result Size: 785 x 640



<!DOCTYPE html>

<html>

<body>

​

<?php

$favcolor = "";

​

switch ($favcolor) {

case "red":

echo "Your favorite color is red!";

break;

case "blue":

echo "Your favorite color is blue!";

break;

case "green":

echo "Your favorite color is green!";

break;

default:

echo "Your favorite color is neither red, blue, nor green!";

}

?>

</body>

</html>

​

While loop condition

<?php  
$x = 1;  
  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>

otice that in a do while loop the condition is tested AFTER executing the statements within the loop. This means that the do while loop would execute its statements at least once, even if the condition is false the first time.

The example below sets the $x variable to 6, then it runs the loop, **and then the condition is checked**:

Example

<?php  
$x = 6;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>

[Try it Yourself »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_loop_do_while2)

The for loop and the foreach loop will be explained in the next chapter.

## PHP Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument ($fname). When the familyName() function is called, we also pass along a name (e.g. Jani), and the name is used inside the function, which outputs several different first names, but an equal last name:

### Example

<?php  
function familyName($fname) {  
    echo "$fname Refsnes.<br>";  
}  
  
familyName("Jani");  
familyName("Hege");  
familyName("Stale");  
familyName("Kai Jim");  
familyName("Borge");  
?>

n 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 »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_num)

## 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

## PHP Indexed Arrays

There are two ways to create indexed arrays:

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

$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] . ".";  
?>

[Try it Yourself »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_num)

## 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 »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_length)

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

[Try it Yourself »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_num_loop)

## 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.";  
?>

[Try it Yourself »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_assoc)

## 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>";  
}  
?>

[Try it Yourself »](https://www.w3schools.com/php/phptryit.asp?filename=tryphp_array_assoc_loop)

## Multidimensional Arrays

[Multidimensional arrays](https://www.w3schools.com/php/php_arrays_multi.asp) will be explained in the PHP advanced section.

## Complete PHP Array Reference