Advantages of PHP Function

* Reusability of Code: Unlike other [programming languages](https://www.simplilearn.com/best-programming-languages-start-learning-today-article), PHP Functions are specified only once and can be called multiple times.
* Less Code: It saves a lot of code because the logic doesn't have to be written several times. You can write the logic only once and reuse it by using functions.
* Simple to Comprehend: The [programming](https://www.simplilearn.com/how-to-learn-programming-article) logic is separated using PHP Functions. Since every logic is divided into functions, it is easier to understand the application's flow.

Creating and Calling Function

In PHP, the function name is any name that ends in an open and closed parenthesis.

* The keyword function is often used to start a function name.
* To invoke a function, simply type its name followed by the parenthesis.
* A number cannot be the first character in a feature name. It can begin with a letter or an underscore.
* The case of a feature name is unimportant.

Syntax:

function function\_name()

{

//Statement to be executed

}

Code:

<!DOCTYPE html>

<html>

<body>

<?php

functionwelcMsg()

{

echo "Hello welcome!";

}

welcMsg();

?>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<?php

functionwelcMsg()

{

echo "Hello welcome!";

}

welcMsg();

?>

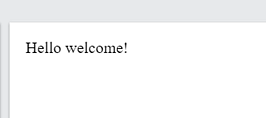
</body>

</html>

Explanation:

In the above example, a function called "welcMsg()" is created. The beginning of the function code is indicated by the opening curly bracket {, and the end of the function is indicated by the closing curly bracket }. The function says "Hello welcome!"; write the function's name in brackets () to invoke it.

Output:



Get the Must-Have Skills of a Web Developer

PHP Function Arguments

In PHP Function, arguments may be used to transfer information to functions. A variable is the same as an argument. Arguments are listed within parentheses after the function name. You can add as many arguments as you want; just use a comma to divide them.

Parameters are the information or variables contained within the function's parenthesis. These are used to store the values that can be executed at runtime. A user can enter as many parameters as he desires, separated by the comma (,) operator. During runtime, these parameters are used to accept inputs. Arguments are used while exchanging values, such as during a function call.

Code:

<!DOCTYPE html>

<html>

<body>

<?php

functionStudentsName($firstname)

{

echo "$firstname<br>";

}

StudentsName("Janani");

StudentsName("Helen");

StudentsName("Stella");

StudentsName("Kamal");

StudentsName("Babu");

?>

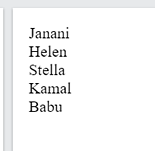
</body>

</html>

Explanation:

A function with only one argument ($fname) is used in the following example. When we call the StudentsName() function, we also transfer a name (for example, Janani) used within the function to generate several different first names but the same last name.

Output:



Passing Arguments By Reference

Arguments are generally passed by value in PHP, which ensures that the function uses a copy of the value and the variable passed into the function cannot be modified. Changes to the argument modify the variable passed in when a function argument is passed by reference. The & operator is used to convert a function argument into a reference.

Code:

<!DOCTYPE html>

<html>

<body>

<?php

functionaddition(&$val) {

  $val += 10;

}

$number =5;

addition($number);

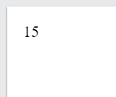
echo $number;

?>

</body>

</html>

Output:



PHP Is a Loosely Typed Language

Depending on the variable’s value, PHP automatically assigns it a data form. You can do things like adding a string to an integer without making an error because the [data types](https://www.simplilearn.com/what-is-data-article) strictly aren't set. Kind declarations were introduced in PHP 7. This allows one to define the intended data type when declaring a method, and the strict declaration ensures that if the data type mismatches, a "Fatal Error" is thrown.

Without using strict, we try to submit both a number and a string to the function in the following example:

Example:

<?php

function add(int $s, int $t) {

return $s + $t;

}

echo add(5, "8 days");

// since strict is NOT enabled, "8 days" is changed to int(8), and it will return 13.

?>

Output:



Get the Coding Skills You Need to Succeed

PHP Functions - Returning Value

This means the PHP Function can be called by its name, and when executing the function, it will return some value.

Example:

<?php

function circle($r){

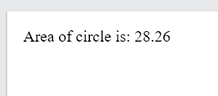
return 3.14\*$r\*$r;

}

echo "Area of circle is: ".circle(3);

?>

Output:



Setting Default Values for Passing Arguments

It is possible to specify a default argument value in the function. While calling the Function in PHP, it will take the default value if no arguments are passed.

Example:

<?php declare(strict\_types=1); // strict requirement ?>

<!DOCTYPE html>

<html>

<body>

<?php

functionsetval(int $a = 5) {

echo "The value is : $a <br>";

}

setval(50);

setval();

setval(13);

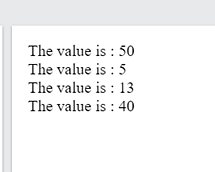
setval(40);

?>

</body>

</html>

Output:



Dynamic Function Calls

In Dynamic Function Calls, it is possible to assign function names to variables as strings and then handle them as function names.

Code:

<html>

<head>

<title>Listing 6.5</title>

</head>

<body>

<?php

function say() {

print "hello<br>";

   }

  $function\_holder = "Hello";

  $function\_holder();

  ?>

</body>

</html>

Output:

