

# Introduction to Functions in PHP

## return statement in PHP

The return statement is used to return the final output of a function. Once the first return statement is executed, the function is exited.

```
function returnMe() {  
    return 6;  
    //this value is returned  
    return 8;  
    //this statement is not executed  
}  
  
echo returnMe();
```

## Invoking a function in PHP

Functions are invoked by using the function name followed by parentheses.

```
//First the function needs to be defined:  
function callFunc() {  
    //code goes here  
}  
  
//Then the function can be invoked:  
callFunc();
```

## Define PHP Function

A function contains a set of instructions to be executed. It is defined by using the keyword `function` followed by the name of the function, the parentheses which contain the parameters, and finally the curly braces which contain the code block.

```
function plus($x,$y) {  
    return $x + $y;  
}  
  
echo plus(10, 5);  
//prints "15"
```

## Camel Case Function

When writing a function in PHP, the convention is to use camel case. This means that we start with a lowercase letter and then capitalize the first letter of every new word.

Additionally, the function name should typically start with a verb.

```
// This is a camel case function
function calculateSum() {}

// This is not a camel case function; it
// is a snake_case function
function calculate_sum() {}

// This is not a camel case function
function CalculateSum() {}
```

## PHP Variable Scope

A variable with local scope can only be accessed within the function it is declared. A variable with global scope can be accessed from multiple functions in the PHP script.

```
<?php

$x = 6;

function scope(){
    $y = 7;

    echo $x;
    // prints 'undefined variable'

    global $x;
    echo $x;
    // prints 6

    echo $y;
    // prints 7
}

scope();
```



# PHP Built-in Functions

## PHP built-in functions

PHP comes standard with many built-in functions. They give us an easier way to implement and repeat popular tasks throughout a program. A popular example is the `echo` function.

Many more are documented at

<https://www.php.net/docs.php>. In addition to a

description of what the function does, the

documentation for each function denotes:

- The name of the function
- Required parameters and their types
- Optional parameters (in square brackets, []), their types, and default values
- The return type of the function (after the final colon, :)

```
echo "I am built in";
```

 **Print**    **Share** ▼