## Function

A function is a self-contain program segment that carries out some specific well-defined tasks or a group of statements that together perform a task. In PHP, we can construct several functions according to our requirement. A function will not execute immediately when a page loads. A function will be executed by a call to the function.

## Advantages of functions:

1. **Program development made easy:** Work can be divided among project members thus implementation can be completed in parallel.
2. **Program testing becomes easy :** Easy to locate and isolate a faulty function for further investigation
3. **Code sharing becomes possible:** many other programs this means that a PHP programmer can use function written by others, instead of starting over from scratch, may use a function later.
4. **Code re-usability increases:** A function can be used to keep away from rewriting the same block of codes, which we are going, use two or more locations in a program. This is especially useful if the code involved is long or complicated.
5. **Avoid un-necessary repetition of code:** suppose you have a program that calculates square root of a number, later if you want to calculate square root of another number, you do not want to write same instructions all over again. Simply make a function that calculates square root & call it whenever need it.
6. **Increases program readability:** It makes possible top down modular programming. In this style of programming, the high-level logic of the overall problem is solved first while the details of each lower level functions is addressed later. The length of the source program can be reduced by using functions at appropriate places.
7. **Function facilitates procedural abstraction:** Once a function is written, it serves as a black box. All that a programmer would have to know to invoke a function would be to know its name, and the parameters that it expects

**Note:**

**A function name can start with a letter or underscore (not a number). Give the function a name that reflects what the function does. Function names are NOT case-sensitive.**

**Syntax:**

```
function    Name-of-function (parameters,..)
{
        function_body;
}
```

- **Function Name** − this is the actual name of the function. The function name and the parameter list together constitute the function signature.
- **Parameters** − A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional, that is, a function may contain no parameters.

- **Function Body** – The function body contains a collection of statements that define

**Types of functions:**

A function may belong to any one of the following categories:

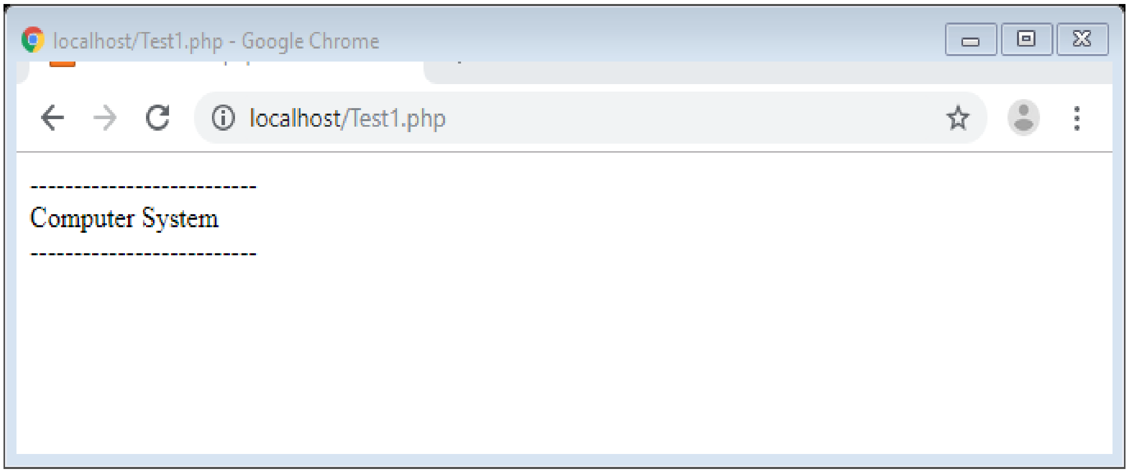
1. Functions with no arguments and no return values.
2. Functions with arguments and no return values.
3. Functions with arguments and return values.
4. Functions with no arguments and return values.

## 1. Functions with no arguments and no return value.

A PHP function without any arguments means you cannot pass data (values like integer string etc.) to the called function. Similarly, function with no return type does not pass back data to the **calling function**. It is one of the simplest types of function in PHP. This type of function, which does not return any value, cannot be used in an expression it can be used only as independent statement.

**Example**

```
<?php
        function line()                            // function definition
        {           // function body
          for($i=0; $i<=25; $i++)
          {
              echo ("-");
           }
           echo "<br>";
        }

        line();                              // function call ended with semicolon
        echo " Computer System <br>";
        line();
?>
```



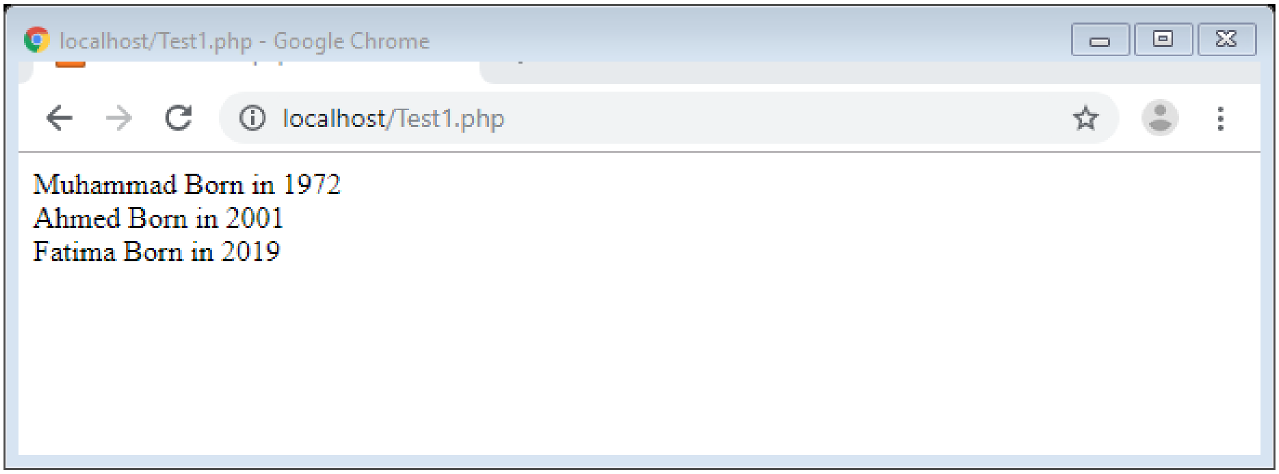
```
--------------------------
Computer System
--------------------------
```

## 2. Functions with arguments and no return value.

A PHP function with arguments can perform much better than previous function type. This type of function can accept data from calling function. In other words, you send data to the called function from calling function but you cannot send result data back to the calling function. Rather, it displays the result on the terminal. However, we can control the output of function by providing various values as arguments.

**Example-1**

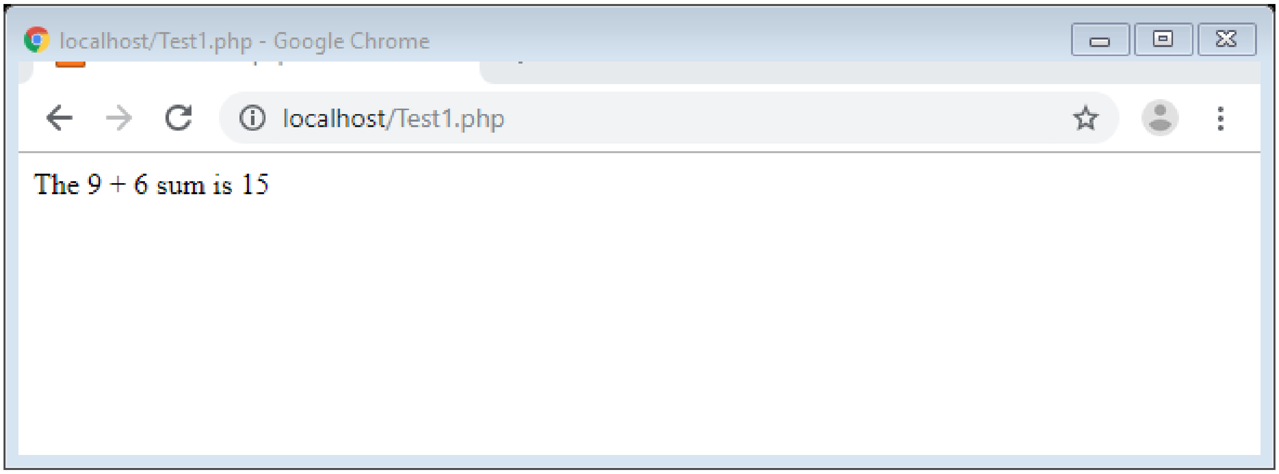
```
<?php
        function familyName($fname,$year)
        {
                echo "$fname  Born in $year <br>";
        }

         familyName("Muhammad ", 1972);
         familyName("Ahmed ", 2001);
         familyName("Fatima", 2019);
?>
```



```
Muhammad Born in 1972
Ahmed Born in 2001
Fatima Born in 2019
```

**Example-2**

```
<?php
        function   add( $x ,   $y)    //function definition
        {
                $sum = $x + $y;
                echo " The $x + $y  sum is $sum";
        }
         add(9,6);      // call the function
?>
```



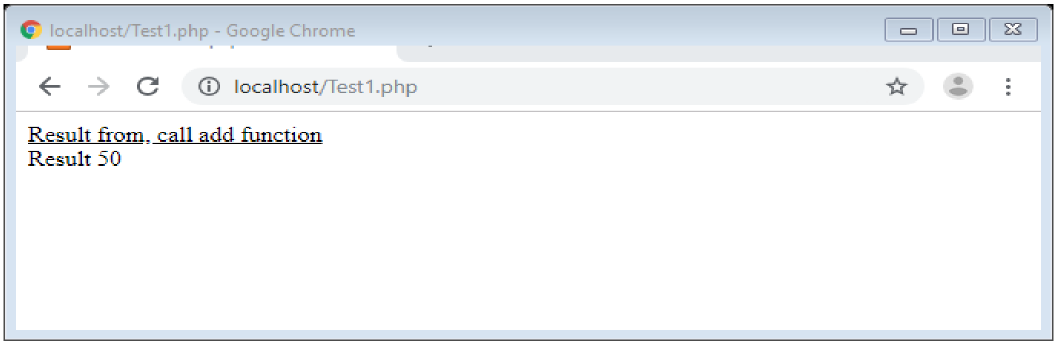
```
The 9 + 6 sum is 15
```

## 3. Functions with arguments and return value.

This type of function can send arguments (data) from the calling function to the called function and wait for the result to be returned back from the called function back to the calling function. And this type of function is mostly used in programming world because it can do two way communications; it can accept data as arguments as well as can send back data as return value. The data returned by the function can be used later in our program for further calculations.

### Example

```
<?php
     $x=30;
      $y=20;
     echo " <u>Result from, call add function</u>  <br>";
          $z = add( $x, $y);      //function call ,the return value of function will stored to z
           echo "Result  $z <br>";       // result of function will display here
          function   add( $x,  $y)       //  function has 2 arguments and return an integer value
          {
                $result;
                $result = $x + $y;
                 return ( $result);
           }
     ?>
```



```
Result from, call add function
Result 50
```

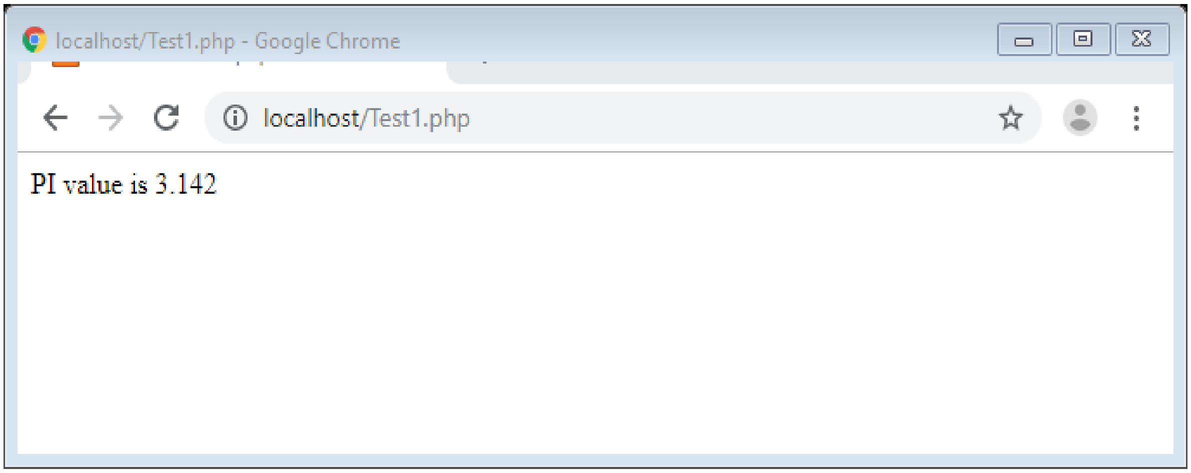
### 4. Functions with no arguments but returns value.

This type of function, which does not take any argument but only returns values to the calling functions.

**Example**

```
<?php
  $z;
      $z = pivalue();      //function call ,the return value of function will stored to z

    echo "PI value is $z <br>";// result of function will display here

    function pivalue()          // function definition
    {
            return 3.142;
    }
?>
```

All functions and classes in PHP have the global scope - they can be called outside a function even if they were defined inside and vice versa. PHP does not support function overloading, or is it possible to un-define or redefine previously-declared functions.



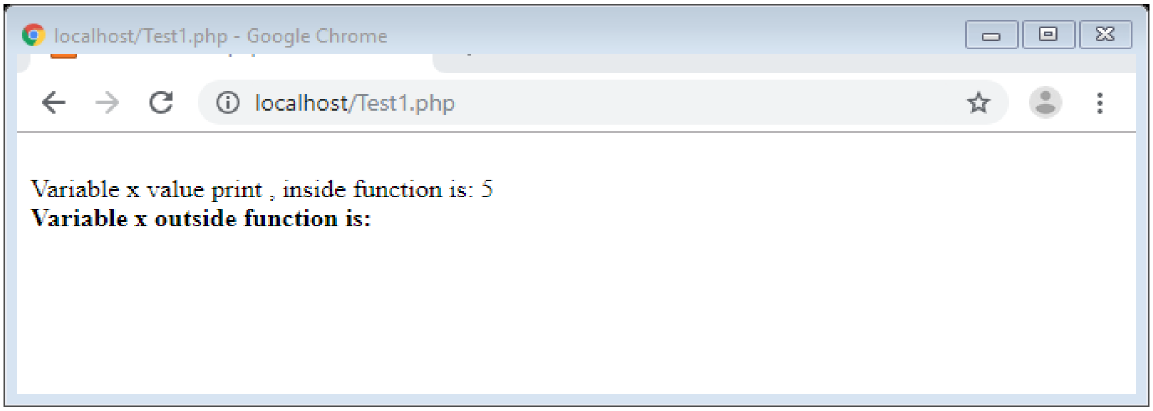
```
PI value is 3.142
```

### Local variables scope:

The variables declared within a function are called local variables to that function and has its scope only in that particular function. In simple words it cannot be accessed outside that function. Any declaration of a variable outside the function with same name as that of the one within the function is a complete different variable.

### Example

```
<?php
   function myFunction()
  {
     $x = 5;      // local scope or local variable
     echo "<br>Variable x value print , inside function is: $x ";
 }
  myFunction();

   // using $x variable outside the function will display nothing value or may be error.
   echo "<b>Variable x outside function is: </b>". $x;
?>
```

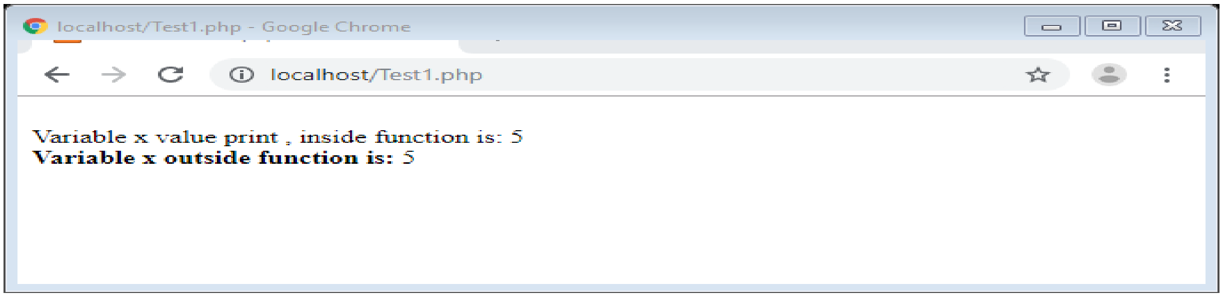


```
Variable x value print , inside function is: 5
Variable x outside function is:
```

### Global variables Scope:

The variables declared outside a function are called global variables. These variables can be accessed directly outside a function. To get access within a function we need to use the “global” keyword before the variable to refer to the global variable.

```
<?php
    $x=5;
   function myFunction()
  {
      global  $x;      // local scope or local variable
      echo "<br>Variable x value print , inside function is: $x  <br> ";
  }
   myFunction();     // call myFunction

    // using $x variable outside the function will be display 5 also.
    echo "<b>Variable x outside function is: </b>". $x;
?>
```



```
Variable x value print , inside function is: 5
Variable x outside function is: 5
```

## Recursion:

A function that calls itself again and again until some test pattern remain true , is called "**Recursive** " function and this process is known as "**Recursion**".

## Example

```
<?php
  function add ( $i )
 {
  $n;
 if( $i ==0 )
       return 0;
 else
      $n = $i+ add($i-1);
        return $n;
 }
 $x=add(10);
 echo "Sum : $x  ";
?>
```

**Output**

Set number 10 in $x with add (10) function

Sum: 55

## Explanation:

User set "10" passes to "add" , Function will check either zero or not , if yes , then it will return "0" ,otherwise it will add integer values until it becomes zero.

## Advantages:

- It is easy to use.
- It represents compact programming structures.

## Disadvantage:

- It is slower than that of looping statements because each time function called.

## Points to Remember

- There should be at least one if statement used to terminate recursion.
- It does not contain any looping statements.

## Exercise

### Theory Questions

1. Describe the purpose of the return statement in a function.
2. Explain why some functions do not have a return statement.
3. Explain the different between a local variable and global variable.
4. Describe the recursion?
5. What is difference between the users-define and pre-define functions in PHP.

### Practical Questions

1. Write a program to display your complete profile, using at least 3 functions.

   Hint: If user press 1: your bio-data will display onto the screen

   If user press 2: your academic profile will display onto screen
2. Write a program to find;

   a. Surface area (A=4pi r 2)

   b. volume(v=4/3 pi r3 ) of a sphere using functions make at least two functions.
3. Write a function to calculate if a number is prime, Return 1 if it is prime and 0 if it is not a prime.
4. Write a program to calculate factorial of a number input by the user.

   (Hint: use Recursion)
5. Make a calculator using functions sum( ), sub( ), mul( ), div( ) for calculations.
6. Write a function ***pow($x,$y)*** to calculate the value of $x raised to $y.

### Objective MCQ's

1. A variable that is declared outside a function is called a ____ variable.
   a) Local
   b) Global
   c) Program
   d) Class

2. A local variable must be declared __________.
   a) Before a function
   b) After a function
   c) Within the braces of a function definition.
   d) With the local keyword.

3. A(n) ______ allows you to treat a related group of PHP commands as a single unit.
   a) Statement
   b) Variable
   c) Event
   d) Function

4. When create user-define function should be use which keyword with function name.
   a) Function
   b) func
   c) function
   d) Procedure

5. The function calls itself this process is called_________.
   a) Repetitive
   b) Decision
   c) Recursion
   d) loop