Function:

A function is a block of **code written in a program to perform some specific ask.** Functions take information/input as parameters, executes a block of statements or perform operations on these parameters and returns the result.

PHP provides two major types of functions:

- **Built-in functions:** PHP provides us with huge collection of built-in library functions. These functions are already coded and stored in form of functions. e.g. fopen(), var_dump, gettype() etc.
- **User Defined Functions:** PHP allows to create customized functions called user-defined functions. Using this we can create our own packages of code and use it wherever necessary by simply calling it.

Why functions should be used?

- **Reusability:** If we have a common code that we would like to use at various parts of a program, we can simply contain it within a function and call it whevener required. This reduces the time and effort of repetition of a single code. This can be done both within a program and also by importing the PHP file, containing the function, in some other program.
- **Easier Error Detection:** Since, our code is divided into functions, we can easily detect in which function, the error could lie and fix them fast and easily.
- **Easily Maintained:** As we have used functions in our program, so if anything or any line of code needs to be changes, we can easily change it inside the function and the change will be reflected everywhere, where the function is called. Hence, easy to maintain.

Creating a Function:

While creating a user defined function we need to keep few things in mind:

1. Any name ending with an open and closed parenthesis is a function.
2. A function name always begins with the keyword *function.*
3. To call a function we just need to write its name followed by the parenthesis.
4. A function name cannot start with a number. It can start with an alphabet or underscore.

Syntax:

function function_name()

{

executable code;

}

There are three components to every function:

- Arguments, which serve as inputs to the function
- Return values, which are the outputs returned by the function
- The function body, which contains the processing code to turn inputs to outputs.

Using Arguments:

Arguments are placeholder variables within a function definition. They are replaced at run time by values provided to the function from the main program. The processing code within the function then manipulates these values to return the desired result.

Since the input to the function will differ each time it is invoked, the output will necessarily differ too.

Example:

<?php

//function definition

//calculate perimeter of rectangle

function getPerimeter($length, $width) {

$perimeter = 2*($length+$width);

echo "The perimeter of a rectangle of length $length units and width $width units is: $perimeter units";

}

//function invocation

//with arguments

getPerimeter(4,2);

}

Using return value:

In PHP you can have a function explicitly return a value, like the result of a calculation, to the statement that called it. This is accomplished by using a return statement inside the function :

```php
<?php

function getPerimeter($length, $width)

{

$perimeter=2*($length+$width);

return $perimeter;

}

echo "The perimeter of a rectangle of length 4 units and width 2 units is: '.getPerimeter(4,2).'units';

?>
```