**PHP Function: Numeric, Built in, String, Date, User Defined**

**What is a Function?**

A function is a reusable piece or block of code that performs a specific action.

Functions can either return values when called or can simply perform an operation without returning any value.

PHP has over 700 functions built in that perform different tasks.

**In this tutorial, you will learn-**

* [Why use Functions?](https://www.guru99.com/functions-in-php.html#1)
* [Built in Functions](https://www.guru99.com/functions-in-php.html#2)
* [String Functions](https://www.guru99.com/functions-in-php.html#3)
* [Numeric Functions](https://www.guru99.com/functions-in-php.html#4)
* [Date Function](https://www.guru99.com/functions-in-php.html#5)
* [Why use User Defined Functions?](https://www.guru99.com/functions-in-php.html#6)

**Why use Functions?**

* Better code organization – functions allow us to group blocks of related code that perform a specific task together.
* Reusability – once defined, a function can be called by a number of scripts in our PHP files. This saves us time of reinventing the wheel when we want to perform some routine tasks such as connecting to the database
* Easy maintenance- updates to the system only need to be made in one place.

**Built in Functions**

Built in functions are functions that exist in PHP installation package.

These built in functions are what make PHP a very efficient and productive scripting language.

The built in functions can be classified into many categories. Below is the list of the categories.

**String Functions**

These are functions that manipulate string data, refer to the article on strings for implementation examples of string functions

**Numeric Functions**

Numeric functions are function that return numeric results.

Numeric php function can be used to format numbers, return constants, perform mathematical computations etc.

**The table below shows the common PHP numeric functions**

| **Function** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- |
| is\_number | Accepts an argument and returns true if its numeric and false if it’s not | <?php  if(is\_numeric("guru"))  {  echo "true";  }  else  {  echo "false";  }  ?> | false |
| <?php  if(is\_numeric (123))  {  echo "true";  }  else  {  echo "false";  }  ?> | true |
| number\_format | Used to formats a numeric value using digit separators and decimal points | <?php  echo number\_format(2509663);  ?> | 2,509,663 |
| rand | Used to generate a random number. | <?php  echo rand();  ?> | Random number |
| round | Round off a number with decimal points to the nearest whole number. | <?php  echo round(3.49);  ?> | 3 |
| sqrt | Returns the square root of a number | <?php  echo sqrt(100);  ?> | 10 |
| cos | Returns the cosine | <?php  echo cos(45);  ?> | 0.52532198881773 |
| sin | Returns the sine | <?php  echo sin(45);  ?> | 0.85090352453412 |
| tan | Returns the tangent | <?php  echo tan(45);  ?> | 1.6197751905439 |
| pi | Constant that returns the value of PI | <?php  echo pi();  ?> | 3.1415926535898 |

**Date Function**

The date function is used to format[Unix](https://www.guru99.com/unix-linux-tutorial.html)date and time to human readable format.

Check the article on PHP date functions for more details.

Other functions

These include;

* Arrays – see the article on arrays for examples
* Files – see the article on files for examples
* Database functions – see the article on [MySQL PHP and other database access methods](https://www.guru99.com/mysql-php-and-other-database-access-methods.html)v2

**Why use User Defined Functions?**

User defined functions come in handy when;

* you have routine tasks in your application such as adding data to the database
* performing validation checks on the data
* Authenticating users in the system etc.

These activities will be spread across a number of pages.

Creating a function that all these pages can be calling is one of the features that make PHP a powerful scripting language.

Before we create our first user defined function, let’s look at the rules that we must follow when creating our own functions.

* Function names must start with a letter or an underscore but not a number
* The function name must be unique
* The function name must not contain spaces
* It is considered a good practice to use descriptive function names.
* Functions can optionally accept parameters and return values too.

Let’s now create our first function. We will create a very basic function that illustrates the major components of a function in PHP.

<?php

//define a function that displays hello function

function add\_numbers(){

echo 1 + 2;

}

add\_numbers ();

?>

**Output:**

3

  HERE,

* “function…(){…}”  is the function block that tells PHP that you are defining a custom function
* “add\_numbers” is the function name that will be called when using the function.
* “()” can be used to pass parameters to the function.
* “echo 'Hello function!';” is the function block of code that is executed. It could be any code other than the one used in the above example.

Let’s now look at a fairly complex example that accepts a parameter and display a message just like the above function.

Suppose we want to write a function that prints the user name on the screen, we can write a custom function that accepts the user name and displays it on the screen.

**The code below shows the implementation.**

<?php

function display\_name($name)

{

echo "Hello " . $name;

}

display\_name("Martin Luther King");

?>

**Output:**

Hello Martin Luther King

  HERE,

* “…($name){…” is the function parameter called name and is initialized to nameless. If no parameter is passed to the function, nameless will be displayed as the name. This comes in handy if not supplying any parameter to the function can result in unexpected errors.

Let’s now look at a function that accepts a parameter and then returns a value. We will create a function that converts kilometers to miles. The kilometers will be passed as a parameter. The function will return the miles equivalent to the passed kilometers. The code below shows the implementation.

<?php

function kilometers\_to\_miles($kilometers = 0)

{

$miles\_scale = 0.62;

return $kilometers \* $miles\_scale;

}

echo kilometers\_to\_miles(100);

?>

**Output:**

62

**Summary**

* Functions are blocks of code that perform specific tasks
* Built in functions are functions that are shipped with PHP
* PHP has over 700 built in functions
* String functions manipulate string data
* Numeric functions manipulate numeric data
* Date functions manipulate date data
* Other functions such as is\_array, fopen etc. are used to manipulate arrays and files respectively
* User defined functions are functions that you can create yourself to enhance PHP