4- PHP Function: Built in, String, Numeric with Examples

What is a Function in PHP?

A **Function in PHP** is a reusable piece or block of code that performs a specific action. It takes input from the user in the form of parameters, performs certain actions, and gives the output. 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.

Why use Functions?

- Better code organization – PHP 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.

PHP Built in Functions

Built in functions are predefined functions in PHP that exist in the installation package. These PHP inbuilt functions are what make PHP a very efficient and productive scripting language. The built in functions of PHP can be classified into many categories. Below is the list of the categories.

String Functions

These are functions that manipulate string data.

Numeric Functions

Numeric functions in PHP are the functions 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\nif(is_numeric("guru"))\n{\n echo "true";\n}\nelse\n{\n echo "false";\n}\n?>```	false
		```<?php\nif(is_numeric (123))\n{\n echo "true";\n}\nelse\n{\n echo "false";\n}\n?>```	true
number_format	Used to formats a numeric value using digit separators and decimal points	```<?php\necho\nnumber_format(2509663)\n;\n?>```	2,509,663
rand	Used to generate a random number.	```<?php\necho rand();\n?>```	Random number
round	Round off a number with decimal points to the nearest whole number.	```<?php\necho round(3.49);\n?>```	3
sqrt	Returns the square root of a number	```<?php\necho sqrt(100);\n?>```	10
cos	Returns the cosine	```<?php\necho cos(45);\n?>```	0.525321988 81773

Function	Description	Example	Output
sin	Returns the sine	<?php echo sin(45); ?>	0.850903524 53412
tan	Returns the tangent	<?php echo tan(45); ?>	1.619775190 5439
pi	Constant that returns the value of PI	<?php echo pi(); ?>	3.141592653 5898

Date Function
The date function is used to format date and time to human readable format.

Other functions

These include;

- Arrays
- Files
- Database functions

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
<?php

//define a function that displays add 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
<?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
<?php
function kilometers_to_miles($kilometers = 0)
{
$miles_scale = 0.62;
return $kilometers * $miles_scale;
}
echo kilometers_to_miles(100);
?>
```
Output:

62

Summary

- Define function in PHP: Function is a block of code that performs specific task.
- Built in function in PHP is a function that is 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