



## 5. SERVER SIDE SCRIPTING (PHP)

### Index:

---

- Features of PHP
- PHP variables, datatypes
- PHP Array, String function and user- defined functions
- PHP form handling
- PHP connectivity with database server

From:

Mrs. Shailaja R. Pande (IT Teacher)  
S.B.Patil college of Science &  
Commerce,Ravet



# Introduction to php

- PHP stands for Hypertext Preprocessor.
- It is widely used open source general purpose server- side scripting language.
- It is widely used for web development and can be embedded with HTML.
- PHP scripts are executed on the server.
- PHP is free to download and use.
- PHP runs on various platform like LINUX, UNIX, Mac OS, Windows etc.
- It is compatible for almost all servers used today like XAMMP, Apache, NGINX, lighttpd.
- It supports wide range of databases.
- PHP is easy to learn and runs efficiently on the server side.
- PHP is free to download and one can download it from the official website [www.php.net](http://www.php.net)



## Server side scripting

➤ A server is a computer system that serves as a central repository of data and programs and is shared by all clients.



➤ The server side environment that runs a scripting language is termed as web server.

➤ Server-side scripting as it relates to web pages usually refers to PHP code that is executed on the web server before the data is passed to the user's browser.

➤ In the case of PHP, all PHP code is executed server-side and no PHP code ever reaches the user.

➤ After the PHP code is executed, the information it outputs is embedded in the HTML, which is sent to the viewer's web browser.

➤ It is usually used to provide interactive web sites that interfaces with databases or other data stores on the server.

➤ Few server-side scripting languages are PHP, Java & JSP, Python.

➤ PHP mainly focuses on server-side scripting , which is used to collect form data, generate dynamic page content or send and receive cookies



## FEATURES OF PHP

**Simple:** It is very simple and easy to use as compared to other scripting languages.

**Interpreted:** It is an interpreted language i.e no need of compilation.

**Faster:** It is faster than other scripting language e.g JSP & ASP

**Open Source:** It means you will not pay money to use it .You can freely download and use it.

**Platform Independent:** It will run on all platform like Linux, Unix, Mac OS, Windows.

**Case Sensitive:** It is case sensitive when you declare variables.

**All keywords(echo , if, else etc) ,classes, functions and user-defined functions are not case-sensitive.**

**Error Reporting:** PHP has some pre-defined error reporting constants to generate a warning or error notice.

**Real Time Monitoring:** PHP provides access logging by creating summary of recent accesses for the user.

**Loosely Typed Language:** PHP allows you to use variable without declaring its data type. It will be taken at the time of execution , based on the type, data has its value.



## FIRST SAMPLE CODE OF PHP

- A php file normally contains HTML tags and some PHP scripting code.
- PHP Script usually enclosed in special start and end tag processing instructions.  
`<?php .... ?>` which allows us to move into and out of php mode.
- A PHP script start with `<?php` and ends with `?>`  
Ex- `<? Php`  
        PHP program code goes here  
        `?>`
- Even it allows to embed HTML with PHP.
- Extension of php is ".php"
- PHP script can be placed anywhere in HTML document.
- `echo` is used to display text on web page.

# Sample PHP program

**Program 5.1:**

```
<!DOCTYPE html>
<html>
<body>
<h1>My First PHP Page</h1>
<?php
echo "Hello World!";
?>
</body>
</html>
```

OUTPUT

**Output 5.1:**



**Note :** The PHP code is embedded with HTML tags using `<?php` and `?>`.



lenovo

XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]



XAMPP Control Panel v3.2.4

Modules

Service	Module	PID(s)	Port(s)	Actions
<input checked="" type="checkbox"/>	Apache	4160 2856	80, 443	<div>StopAdminConfigLogs</div>
<input type="checkbox"/>	MySQL			<div>StartAdminConfigLogs</div>
<input type="checkbox"/>	FileZilla			<div>StartAdminConfigLogs</div>
<input type="checkbox"/>	Mercury			<div>StartAdminConfigLogs</div>
<input type="checkbox"/>	Tomcat			<div>StartAdminConfigLogs</div>

Config

Netstat

Shell

Explorer

Services

Help

Quit

1:07:26 PM [main] XAMPP Installation Directory: "c:\xampp\  
1:07:26 PM [main] Checking for prerequisites  
1:07:35 PM [main] All prerequisites found  
1:07:35 PM [main] Initializing Modules  
1:07:35 PM [main] Starting Check-Timer  
1:07:35 PM [main] Control Panel Ready  
1:07:45 PM [Apache] Attempting to start Apache app  
1:07:45 PM [Apache] Status change detected: running

# Learn to execute PHP program

- **Step 1 :** Create folder to save php programs in the directory as follows
  - If your are using windows operating system create a folder in `c:/xampp/htdocs/`
  - If your are using Linux operating system create a folder in `/var/www/html`
  - (for e.g. myphp)
- **Step 2 :** Type the php program using any text editor (for e.g. Notepad, gedit).
- **Step 3 :** Save it as **first.php** in myphp folder.
- **Step 4 :** Open browser and type path <http://localhost/myphp/>





## PHP CASE SENSITIVITY

- In PHP **“Variables names ”are case sensitive.**
- However keywords(ex. If,else,break,for,while,echo), functions, class name are not case sensitive.
- The “ECHO” keyword is Case insensitive.



## HOW TO EXECUTE PHP CODE

### Program 5.2: First.php

```
<?php  
ECHO "Hello World!<br>";  
echo "Hello World!<br>";  
EcHO "Hello World!<br>";  
?>
```

### Output 5.2:

**Hello World!**  
**Hello World!**  
**Hello World!**

**Note :** In above example, HTML tag `<br>` is enclosed in echo output string.

Go to the browser and type <http://localhost/first.php>



## PHP VARIABLES

- Variable is a symbol or name that stands for a value.
- Variables are used for storing values such as numeric, characters, strings or memory addresses, so that they can be used in any part of program.

---

## RULES FOR DECLARING VARIABLES

- Variable starts with \$ sign, followed by name of the variable.
- A variable name must start with a letter or underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores(A-z,0-9 and \_[underscore]) .
- Variable names are case-sensitive. (Eg. \$age and \$AGE are considered different).

## ► Program using PHP variables

```
<!DOCTYPE html>
<html>
<body>
<?php
$txt = "Hello world!";
$x = 5;
    echo $txt;
    echo "<br>";
    echo $x;
?>
</body>
</html>
```

### OUTPUT

Hello world!  
5



## SCOPE OF VARIABLES IN PHP

- **Local**: A variable declared within a function has local scope and **can be accessed within that function.**
- **Global**: A variable declared outside the function has a global scope and **can only be accessed outside the function.**
- **Static**: When a function is executed, then all of its variables are deleted. **If you want a local variable not to be deleted then you must use static keyword.**

---

## Comments

**Comments** :–are not visible in the output of the program as it is ignored during the execution.

A **single line** comment is given by **//** or **#** before the PHP statement and a **multi-line** comment is possible by **/\*...\*/**



### Program 5.3:

```
<?php
$a = 20;
$c = 15;
function myFunction() {
    $b = 10;
    global $c;
    echo "<p> value of 'a' inside function
    is : $a </p>";
    echo "<p> value of 'b' inside function
    is : $b </p>";
    echo "<p> value of 'c' inside function
    is : $c </p>";
}

myFunction();
echo "<p> value of 'a' outside function
is : $a </p>";
echo "<p> value of 'b' outside function
is : $b </p>";
?>
```

### Output 5.3:

value of 'a' inside function is :  
value of 'b' inside function is : 10  
value of 'c' inside function is : 15  
value of 'a' outside function is : 20  
value of 'b' outside function is :

## SCOPE OF VARIABLES IN PHP

### Program 5.4:

```
<?php
function myCount() {
    static $c = 0; // Static Keyword
    echo $c;
    $c++;
}
echo "Output of myCount() with use of
'static' keyword : <br>";
myCount();
echo "<br>";
myCount();
echo "<br>";
myCount();
?>
```

### Output 5.4:

Output of myCount() with use of "static" Keyword:  
0  
1  
2





## PHP DATA TYPES

PHP can store data of different types and PHP supports following data types:

- String
  - Integer
  - Float
  - Boolean
  - Array
  - NULL
- One can **check the data\_type of variable using var\_dump() method in PHP.**
  - var\_dump() gives different output for each data\_type.**

### Program 5.5:

```
<?php  
echo "<br> -- String --<br>";  
$x = "Hello World !";  
echo var_dump($x);  
echo "<br> -- Decimal --<br>";  
$x = "1234";  
echo var_dump($x);  
?>
```

### Output 5.5:

```
--String --  
string(12)" Hello world! "  
--Decimal--  
int(1234)
```

➤ It gives length of the string for “string” data\_type.

➤ It gives actual value of integer for “integer” data\_type.

➤ It gives true/false for “boolean” data\_type



## CONTROL STRUCTURES IN PHP

**If STATEMENT** : **if STATEMENT** ALLOWS PROGRAMMER to MAKE decision, BASED on one or more conditions; AND execute A piece of code CONDITIONALLY.

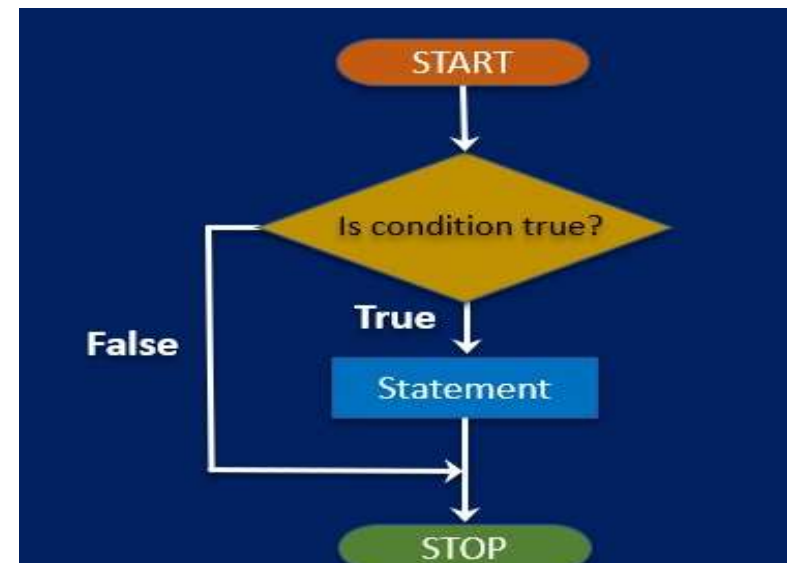
**SYNTAX :**

```
if(condition)
{
    block of statement;
}
```

**2. If-else statement:** **if-else statement** allows programmer to make decision based on either this or that conditions.

**Syntax:**

```
if(condition)
{
    statement;
}
else
{
    statement
}
```





## CONTROL STRUCTURES IN PHP

### IF – ELSE

#### Program 5.6:

```
<?php
$marks=80;
if($marks>=60)
{ echo"you passed with first class";
}
else
{ echo"you can do better";
}
?>
```

#### Output 5.6:

you passed with first class

### SIMPLE IF

```
<!DOCTYPE html>
<html>
<body>
<?php
$t=10;
if ($t < "20")
{
    echo "LESS THAN 20 !";
}
?>
</body>
</html>
```

#### OUTPUT

LESS THAN 20 !



## for.....loop

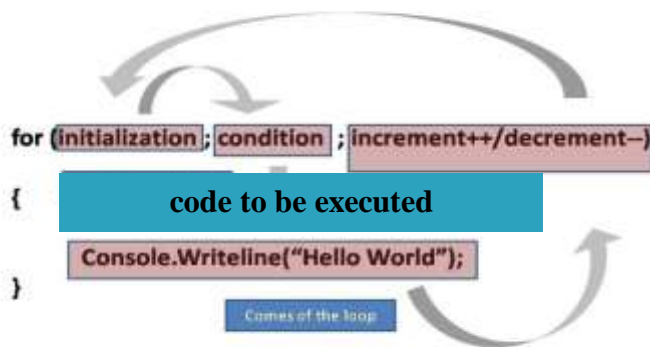
Loops are used to execute the same block of code repeatedly as long as a certain condition is satisfied.

This loop **executes statements as long as condition becomes true**, for-loop is that it combines **initialization, condition and loop iteration (increment or decrement)** in single statement.

### Syntax :

```
for(initialisation;condition;increment/decrement)
{
    Statement;
}
```

Flow of "for" Loop



### Program 5.7:

```
<?php
for($i=1;$i<=5;$i++)
{
    echo"The number is".$i."<br>";
}
?>
```

### Output 5.7:

```
The number is:1
The number is:2
The number is:3
The number is:4
The number is:5
```



## foreach.....loop

**foreach loop:** This loop work only on arrays, and is used to loop through each key/value pair in array. For every loop iteration, the value of the current array element is assigned to \$value and the array pointer is moved by one, until it reaches the last array element.

**Syntax:** `foreach($array as $value)`

```
{  
    code to be executed;  
}
```

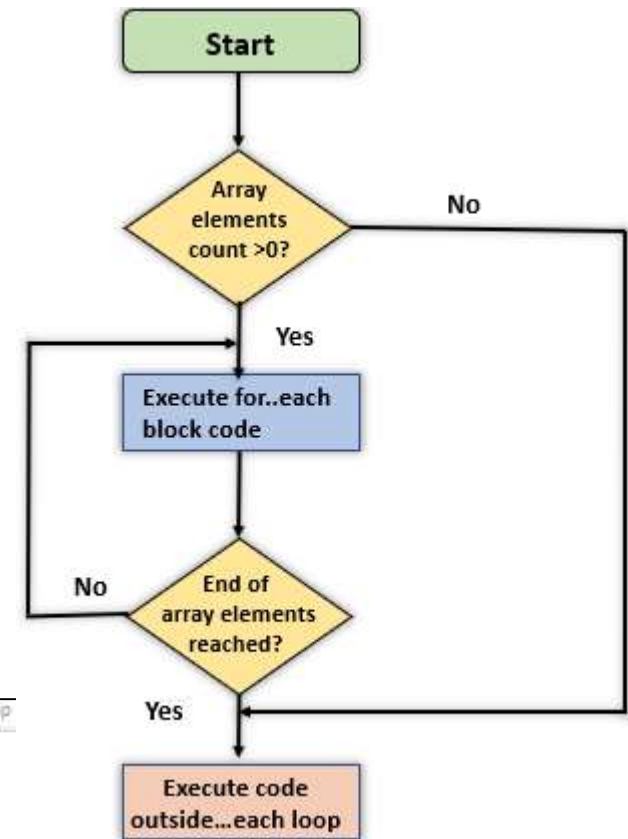
```
1 <?php  
2 $arr_numbers = array(5,10,15,20,25);  
3  
4 foreach($arr_numbers as $i){  
5     echo $i . "<br />";  
6 }  
7 ?>
```

Output

localhost/foreach.php

Php using foreach

```
$  
5  
10  
15  
20  
25
```



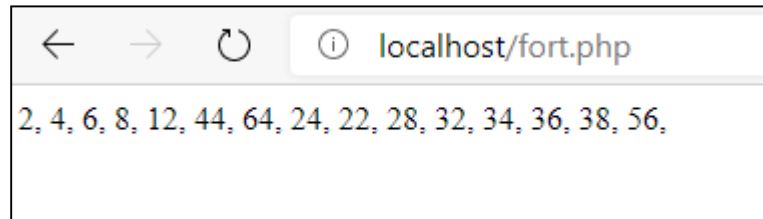


## foreach.....loop

```
<?php
$array1 = array(1, 2, 3, 4, 6, 22, 32, 12, 11, 14, 16, 17, 18, 19, 28);
foreach ($array1 as $value1){
    $value1 = $value1 * 2;
    echo $value1.", ";
}
?>
```



Output







Following are the few predefined functions in PHP to manipulate string.

Function	Description
strlen()	Returns the length of a string (i.e. total no. of characters)
str_word_count()	Counts the number of words in a string
strrev()	Reverses a string
strpos()	Searches for a specific text within a string and returns the character position of the first match and if no match is found, then it will return false
str_replace()	Replaces some characters with some other characters in a string
substr()	Returns a part of a string
strtolower()	Converts a string to lowercase
substr_count()	Counts the number of times a substring occurs in a string
ucwords()	Converts the first character of each word in a string to uppercase
trim()	Removes whitespace and other predefined characters from both sides of a string

**Table 5.1: Pre-defined functions for string manipulation.**



## PHP STRING FUNCTION

- **A string is series of character.**
- **The real power of PHP comes from its functions.**
- **A function is block of statement used repeatedly in a program.**
- **PHP has many built-in functions which can be called directly to perform a specific task.**

### Program 5.8:

```
<?php
$str="Textbooks produced by Balbharati
are also published in pdf format. ";
echo "<br>String: ".$str;
echo "<br>";
echo "<br>String Length : ".strlen($str);
echo "<br>";
echo "<br>String Word Count
: ".str_word_count($str);
echo "<br>";
echo "<br>Reverse String : ".
strrev($str);
echo "<br>";
echo "<br>Retrun position of string
search : ".strpos($str,"Balbharati");
echo "<br>";
echo "<br>Replace string :
".str_replace("Balbharati","State
Board",$str);
?>
```

### Output 5.8:

```
String : Textbooks produced by Balbharati are also published in pdf format.
String Length : 67
String Word Count : 10
Reverse String : .tamrof fdp ni dehsilbup osla era itarahblaB yb decudorp skoobtxeT
Return position of string search : 22
Replace string: Textbooks produced by State Board are also published in pdf format.
```



## PHP ARRAY

- An array is a special variable which can hold more than a value at a time.
- An array stores multiple values in one single variable.
- The index can be assigned automatically.(index always starts with 0)

**Syntax: \$array\_name=array(value1,value2,....., value n)**



```
$cars  
= array("Volvo", "BMW", "Toyota");  
$cars[0]= Volvo  
$cars[1]= BMW  
$cars[2]= Toyota
```

- **There are 3 types of arrays**
  - **Indexed arrays** - Arrays with a numeric index
  - **Associative arrays** - Arrays with named keys
  - **Multidimensional arrays** - Arrays containing one or more arrays



### Program 5.9:

```
<?php
$subjects = array("English", "Hindi",
"Marathi");
echo "I like ".$subjects[0].",
".$subjects[1]." and ".$subjects[2];
echo "<br> Count : ".count($subjects);
?>
```

### Output 5.9:

**I like English, Hindi and Marathi.**  
**Count : 3**

➤ In this example , we store subject is an array at following index location.

`$subject[0]`="English"  
`$subject[1]`="Hindi"  
`$subject[2]`="Marathi"

`count()` function is used to return the Length of array



## PHP USER DEFINED FUNCTIONS

- A function is block of statement used repeatedly in a program.
- It will not execute immediately when a page loads but will be executed by a call to the function.
- Along with built-in PHP functions we can create our own functions.
- User-defined function starts with the word function.
- Information can be passed to functions through arguments.
- An argument is just a variable.
- Arguments are specified after the function name, inside brackets.

**Syntax:** function fun\_name()  
    {  
        code to be executed  
    }

**Note :** A function name can start with a letter or underscore (not a number). Function names are NOT case-sensitive.

### Program 5.12:

```
<?php  
function writeMsg(){  
    echo "This is user-defined function";  
}  
writeMsg(); //call the function  
?>
```

### Output 5.12:

This is user-defined function



### Program 5.13:

```
<?php
function Student($rollno, $name){
echo "Roll No is $rollno and Name is
$name <br>";
}
Student(1,"Ashwini");
Student(2,"Raj");
Student(3,"Sonam");
?>
```

### Output 5.13:

**Roll No. is 1 and Name is Ashwini**  
**Roll No. is 2 and Name is Raj**  
**Roll No. is 3 and Name is Sonam**

**Note :** String is written in double quotes.





## Use return statement to return a value.

### Program 5.14:

```
<?php
function sum(int $x, int $y) {
    $z = $x + $y;
    return $z;
}
echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
```

### Output 5.14:

5 + 10 = 15
7 + 13 = 20
2 + 4 = 6

**Note :** 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.



## PHP ASSOCIATIVE ARRAYS :

ASSOCIATIVE ARRAYS ARE ARRAYS THAT use NAMED keys INSTEAD of index to identify RECORD/VALUE.

Let us see how to CREATE ASSOCIATIVE ARRAY.

### SYNTAX :

`$a = array( key1 => value1, key2=>value2, ...,key n => value n)`

```
<?php
$STUDENT_MARK =
ARRAY("ENGLISH"=>"75",
"Hindi"=>"64",
"MARATHI"=>"88");
echo "You HAVE scored ".$student_
MARK['ENGLISH']." in English .";
?>
```



You have scored 75 in English.

VALUES of 'STUDENT\_MARK' ARRAY ARE stored in following way:

`$STUDENT_MARK['English'] = "75"`

`$STUDENT_MARK['HINDI'] = "64"`

`$STUDENT MARK['MARATHI'] = "88"`



## When to use GET?

- Information send from a form using GET method is visible to everyone.(names and values of variables are displayed on URL)
- It also has limit on amount of information to send. Because the variables are displayed in the URL ,it is possible to bookmark the page.
- It may be used to send some non-sensitive data.
- It should never be used to for sending passwords or other sensitive information.



## Let us see one program on “get” method

### Program 5.16.1:

```
<html>
<head>
<title>BMI Calculator</title>
</head>
<body>
<form method="get"
action="bmioutput.php">

Weight (kg): <input name="weight"
id="weight" type="text" /> <br/>
Height (cm): <input name="height"
id="height" type="text" /> <br/>

<input name="submit" id="submit"
value="Calculate" type="submit" />

</form>
</body>
</html>
```

### Output 5.16.1:

Weight (Kg) :	<input type="text" value="70"/>
Height (Kg) :	<input type="text" value="165"/>
<input type="button" value="Calculate"/>	



```
<?php
$height = $_GET["height"];
$weight = $_GET["weight"];
$heightInMs = $height/100;
$bmi = $weight/
($heightInMs*$heightInMs);

if($bmi < 18.5)
{
    $message = "You are underweight.";
}
else if($bmi >=18.5 && $bmi <= 24.9)
{
    $message = "Congrats!!! You have
normal weight.";
}
else if($bmi >24.9 && $bmi <=29.9)
{
    $message = "You are overweight.";
}
else
{
    $message = "Be careful!!! You are
obese.";
}

echo $message;
echo "</br> BMI : ".$bmi;
?>
```

**Output 5.16.2:**

**You are over weight.**  
**BMI : 25. 711662075298**

Once you click on 'Calculate' button,  
the output is displayed as shown above.



## When to use POST?

- Information send from a form using POST method is invisible to everyone.(all names and values are embedded within the body of the HTTP request.)
- It has no limit on amount of information to send.
- Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to the server.
- The variables are not passed in the URL so it is possible to bookmark the page.





A simple HTML form with 2 input fields and submit button code is as follows:

**Program 5.15.1:**

```
<html>
<body>
<form action="welcome.php"
method="post">
Name: <input type="text"
name="name"><br>
E-mail: <input type="text"
name="email"><br>
<input type="submit">
</form>
</body>
</html>
```

**Program 5.15.2:**

```
<html>
<body>
Welcome
<?php echo $_POST["name"]; ?> <br>
Your email address is:
<?php echo $_POST["email"]; ?>
</body>
</html>
```

**Output 5.15.1:**

Name:	<input type="text" value="Balbharati"/>
E-mail:	<input type="text" value="Balbharati@balbharati.in"/>
<input type="submit" value="Submit Query"/>	

**Output 5.15.2:**

*Welcome balbharti Your email address is: **balbharti@balbharati.in***

**Note :** 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.



### Program 5.15.2:

```
<html>
<body>
Welcome
<?php echo $_POST["name"]; ?> <br>
Your email address is:
<?php echo $_POST["email"]; ?>
</body>
</html>
```

- When user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named "welcome.php".
- The form data is sent with the HTTP POST method.
- The code for "welcome.php" looks like this

### Output 5.15.2:

*Welcome balbharti Your email address is: [balbharti@balbharati.in](mailto:balbharti@balbharati.in)*



## GET vs POST

- Both GET and POST are treated as \$\_GET and \$\_POST super globals, which means they are accessible ,regardless of scope.
- It can be accessed from function ,class or file without having to do anything special.
- \$\_GET is an array of variables passed via the URL parameters.
- \$\_POST is an array of variables passed via the HTTP POST method.



## Summary

- PHP is widely-used open source server-side programming language which runs on various platforms.
- PHP is a script executed on server which generate dynamic HTML pages.
- The PHP code can also be embedded with HTML tags using `<?php` and `?>`.
- PHP is case sensitive only at time of variable declaration and not-case sensitive for other keywords.
- PHP variable start with \$ sign followed by name of variable which must start with a alpha-numeric characters or underscore character.
- PHP variable has three different scopes namely : local, global and static.
- PHP supports String, Integer, Float, Boolean, Array and NULL data types.
- Three types of Arrays are Indexed array, Associative array and Multi-dimensional array.
- PHP supports 'foreach' loop to iterate easily.
- String functions are used to manipulate strings.



## Summary

- A function is a block of statements that can be used repeatedly in a program.
- Information can be passed to functions through arguments.
- Form is used to collect information from user and process or store in database.
- Form data can be submitted by GET or POST method.
- The PHP superglobals `$_GET` and `$_POST` are used to collect form-data.
- `$_GET` is an array of variables passed via the URL parameters and are visible to everyone.
- `$_POST` is an array of variables passed via the HTTP POST method and are invisible to others.
- GET has limits on the amount of information to send whereas POST has no limits on the amount of information to send.
- GET should NEVER be used for sending sensitive information.
- Cookies are sent along when browser requests server pages.
- Session helps web application to maintain user information on all the pages.



# Thank you !