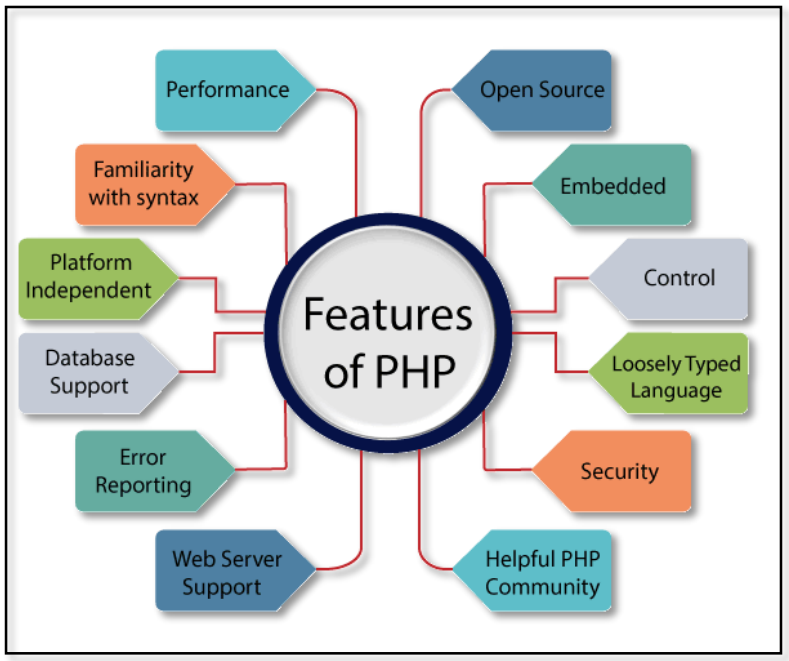
Unit-II PHP Configurations and Basic

Introduction to PHP

- PHP is an acronym for "PHP: Hypertext Pre-processor or Personal Home Page"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use.
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ". php"

PHP Features

- PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:



- Performance:
  - PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.
- Open Source:
  - PHP source code and software are freely available on the web.

  - You can develop all the versions of PHP according to your requirement without paying any cost. All its components are free to download and use.

- Familiarity with syntax:
  - PHP has easily understandable syntax. Programmers are comfortable coding with it.
- Embedded:
  - PHP code can be easily embedded within HTML tags and script.
- Platform Independent:
  - PHP is available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
- Database Support:
  - PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.
- Error Reporting :
  - PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E_ERROR, E_WARNING, E_STRICT, E_PARSE.
- Loosely Typed Language:
  - PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value.
- Web servers Support:
  - PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc.
- Security:
  - PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threads and malicious attacks.
- Control:
  - Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.
- A Helpful PHP Community:
  - It has a large community of developers who regularly updates documentation, tutorials, online help, and FAQs. Learning PHP from the communities is one of the significant benefits.

Installation of Apache, MySQL and PHP

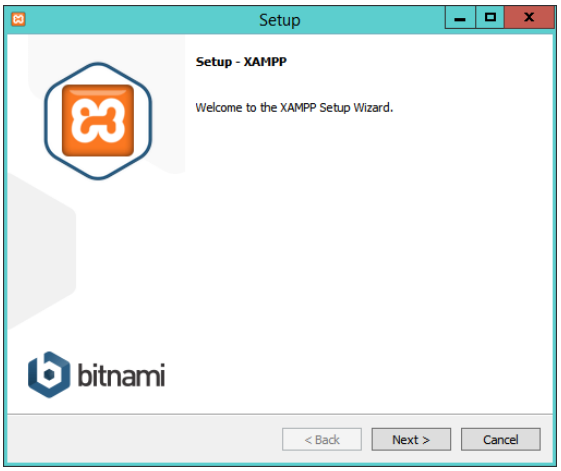
- If you wish to develop Internet applications but don't have your own development server, you can develop one such on your local computer.
- So, testing can be as easy as saving an update (usually just a matter of clicking once on an icon) and then hitting the Refresh button in your browser.
- Following are the fully functioning setup used for developing dynamic Internet web pages.
    1. WAMP - Windows, Apache, MySQL, and PHP
    2. MAMP - Mac, Apache, MySQL, and PHP
    3. LAMP - Linux, Apache, MySQL, and PHP
- WAMPs, MAMPs, and LAMPs come in the form of a package that binds the bundled programs together so that you don't have to install and set them up separately.
- This means you can simply download and install a single program, and follow a few easy prompts, to get your web development server up and running in the quickest time with a minimum hassle.

Installing Xampp on windows

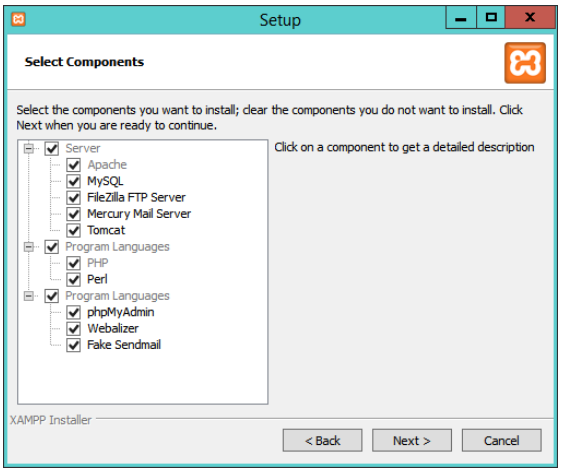
- There are several available WAMP servers, each offering slightly different configurations, but out of the various open source and free options, the best is probably XAMPP.

Link: *http://apachefriends.org*

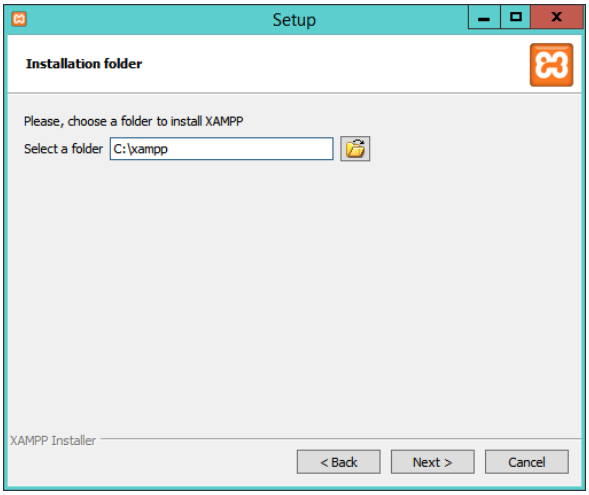
Installation Steps



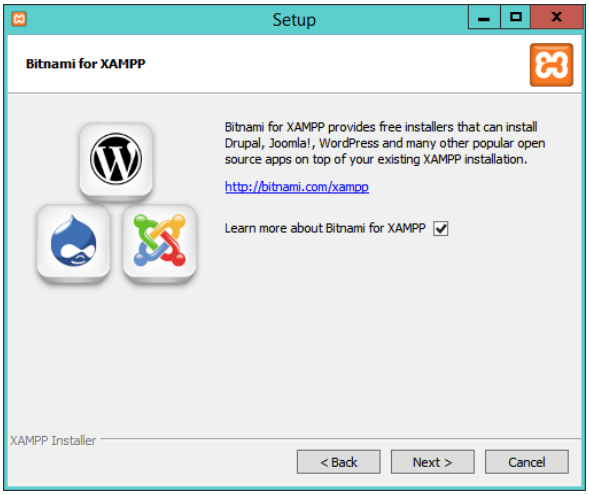
Once you've downloaded the installer, run it. It will show the following window.



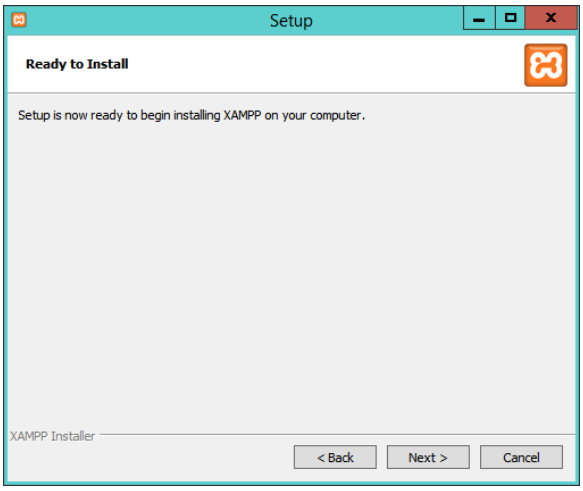
Uncheck any of the components that you don't need. For example, at the minimum for this book you will want to keep Apache, MySQL, PHP, and PHPMyAdmin checked.



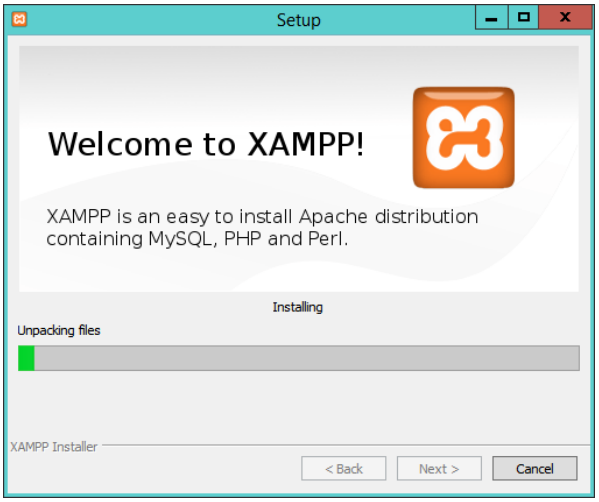
Choose a folder for the installation. You are recommended to accept the default provided unless you have a good reason to choose a different one. If the folder you choose exists and is not empty, you will be unable to use it.



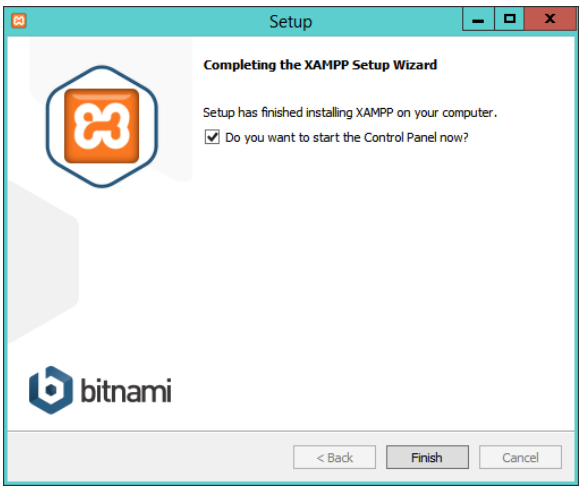
The screen is for supplying information on adding free installers for related products in a new web browser window or tab.



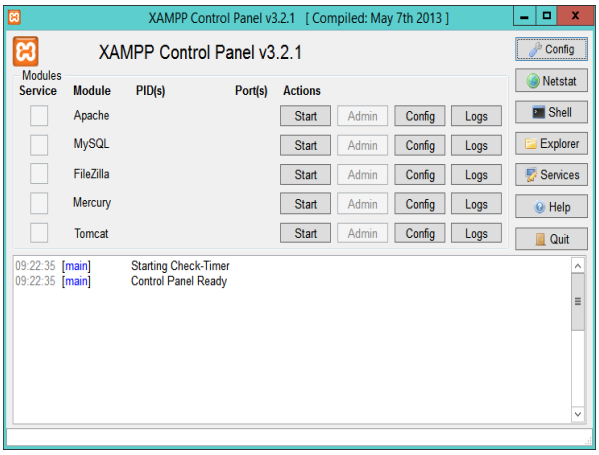
Setup is ready to start. Click next



Installation screen. During installation, you can click any of the icons to open up a web browser window displaying information on the product shown. The whole process should take only a few minutes on most computers.

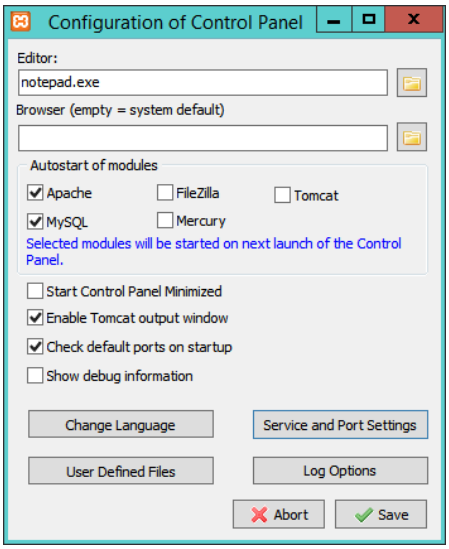


Once installation is complete, the screen will be displayed with a check box already checked for starting the XAMPP control panel. Click Finish.



Now you are ready to start using XAMPP by configuring it from the control panel.

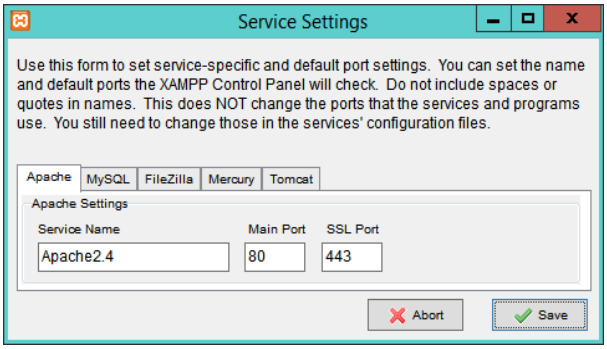
Click the Config button at the top-right corner



Check the Apache and MySQL boxes to ensure they auto start.

OR

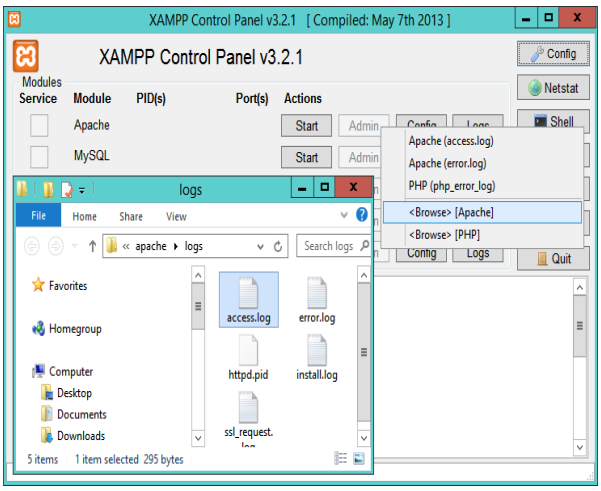
You can simply click the Start buttons for Apache and MySQL to have them run for this session only.



Alter the port settings used by clicking Service and Port Settings to call up the window

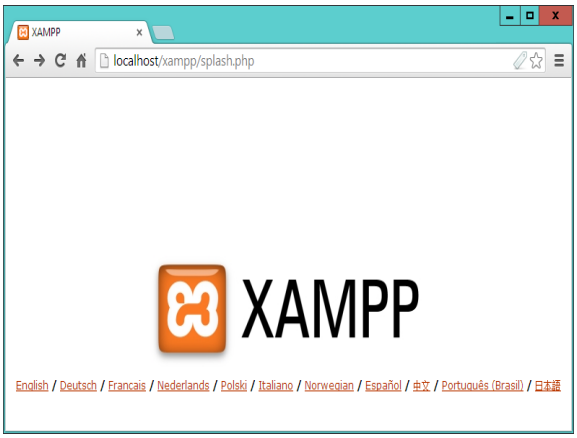
The default ports assigned will normally be port 80 for the Apache web server, 443 for

SSL, and 3306 for MySQL.



If the Apache Log button has been clicked, and the folder of log files

opened.



Browse in your default browser

Localhost Or 127.0.0.1

Port number if you changed

localhost:<port>

127.0.0.1:<port>

How PHP Code is parsed?

- So you have a file, and in that file you have some HTML and some PHP code. This is how it all works, assuming a PHP document with an extension of ". php".
- The web browser requests a document with ". php" extension.
- The web server says, "Hey! Someone wants a PHP file, which means this is a file that needs to be parsed," and sends the request on to the PHP parser.
- The PHP parser finds the requested file and scans it for PHP code.
- When the PHP parser finds PHP code, it executes that code and places the resulting output (if any) into the place in the file formerly occupied by the code.
- This new output file is sent back to the web server.
- The web server sends it along to the web browser.
- The web browser displays the output.

Embedding PHP Code and HTML?

- You can incorporate this into an HTML document by simply adding HTML outside the PHP start tag <?php and end tag ?>.
- When PHP parses a file, it looks for opening and closing tags, which tell PHP to start and stop interpreting the code between them.
- Parsing in this manner allows PHP to be embedded in different documents, as the PHP parser ignores everything outside of a pair of PHP opening and closing tags.
- If you were to view the document source, it would look exactly like a normal HTML document.
- A PHP code block can be placed anywhere in an HTML document.
- Although you can have multiple blocks of code in a single document, they combine to form a single script.

- Any variables defined in the first block will usually be available to subsequent blocks.

Example

```
<?php $var = "Hello Good Morning students."; ?>
<html>
<body>
        <h1>Multiple PHP blocks in a document</h1><hr>
        <?php echo "<h3>Message from php variable</h3>"; ?>
        <h2 align="center">
                <?php echo $var; ?>
        </h2>
</body>
</html>
```

Data Types

- Some programming languages demand that the programmer declares in advance which type of data a variable will contain.
- PHP is **loosely typed**, meaning that it will calculate data types as data is assigned to each variable.
- **Advantage:** The variable can be used flexibly, holding a string at one point and an integer at another.
- **Disadvantage:** This can lead to problems in larger scripts if you expect a variable to hold one data type when in fact it holds something completely different.

Standard Data Types

Type	Example	Description
Integer	5	A whole number
Double	3.234	A floating-point number
String	“hello”	A collection of characters
Boolean	True	One of the special values true or false
Object	$con=new DBconnection();	An instance of a class
Array	$arr=array(“rno”=>21,”name”=> ”Ram”);	An ordered set of keys and values

Special Data Types

Type	Description
Resource	Reference to a third-party resource (a database, for example)
NULL	An uninitialized variable

- Resource types are often returned by functions that deal with external applications or files.
- The type NULL is reserved (default) for variables that have not been initialized.

Operators

- Operators are symbols that make it possible to use one or more values to produce a new value.
- A value that is operated on by an operator is referred to as an operand.
- For example – "5 + 2"; "5" and "2" are operands and are operated by the operator "+" to produce new value "7"
- The combination of operands with an operator to produce a result is called an expression.
- An expression is any combination of functions, values, and operators that resolve to a value.
- There are three types of operators.
- Firstly, there is the unary operator which operates on only one value, for example (the negation operator) or ++ (the increment operator).
- The second group is termed binary operators; this group contains most of the operators that php supports. For example, logical, conditional operator etc…
- The third group is the ternary operator ?: it should be used to select between two expressions depending on a third one.

Assignment operators

Operator	Example	Is equivalent to	Description
=	$x=$y	$x=$y	Assignment
+=	$x += $y	$x = $x + $y	Combined Assignment
-=	$x -= $y	$x= $x - $y	Combined Assignment
*=	$x *= $y	$x=$x * $y	Combined Assignment
/=	$x /= $y	$x=$x / $y	Combined Assignment
%=	$x%=$y	$x=$x % $y	Combined Assignment
.=	$x .= "test"	$x = $x . "test"	Combined Concatenation (string)

Arithmetic operators

Example	Name	Result
-$a	Negation	Opposite of $a. (Unary)
$a + $b	Addition	Sum of $a and $b.
$a - $b	Subtraction	Difference of $a and $b.
$a * $b	Multiplication	Product of $a and $b.
$a / $b	Division	Quotient of $a and $b.
$a % $b	Modulus	Remainder of $a divided by $b.

Increment/decrement operators

Example	Name	Effect
++$a	Pre-increment	Increments $a by one, then returns $a.
$a++	Post-increment	Returns $a, then increments $a by one.
--$a	Pre-decrement	Decrements $a by one, then returns $a.
$a--	Post-decrement	Returns $a, then decrements $a by one.

Comparison or relational operators

Example	Name	Result
$a == $b	Equal	True if $a is equal to $b otherwise false.
$a=== $b	Identical	True if $a is equal to $b, and they are of the same type. (introduced in php 4) otherwise false.
$a != $b	Not equal	True if $a is not equal to $b otherwise false.
$a <> $b	Not equal	True if $a is not equal to $b otherwise false.
$a !== $b	Not identical	True if $a is not equal to $b, or they are not of the same type. (introduced in php 4) otherwise false.
$a < $b	Less than	True if $a is strictly less than $b otherwise false.
$a > $b	Greater than	True if $a is strictly greater than $b otherwise false.
$a <= $b	Less than or equal to	True if $a is less than or equal to $b otherwise false.
$a >= $b	Greater than or equal to	True if $a is greater than or equal to $b otherwise false.

Logical operators

Example	Name	Result
$a and $b	and	True if both $a and $b are true otherwise false.
$a or $b	or	True if either $a or $b is true otherwise false.
$a xor $b	xor	True if either $a or $b is true, but not both.

! $a	not	True if $a is not true. (Unary) otherwise false
$a && $b	and	True if both $a and $b are true otherwise false.
$a || $b	or	True if either $a or $b is true otherwise false.

Ternary (conditional) operator

- The ? : or ternary operator is similar to the if statement, but returns a value derived from one of two expressions separated by a colon.
- This construct will provide you with three parts of the whole, hence the name ternary.
- The expression used to generate the returned value depends on the result of a test expression.

Syntax

- (expression) ? returned_if_expression_is_true : returned_if_expression_is_false;
- If the test expression evaluates to true, the result of the second expression is returned, otherwise, the value of the third expression is returned.

Example

```
<? php
        $a = 30;
        $b = 20;
        $x = ($a>$b) ? $a : $b;
        echo "Max = $x";
?>
```

String operators

- There are two string operators.
- The first is the concatenation operator ('.'), which returns the concatenation of its right and left arguments.
- The second is the concatenating assignment operator ('.='), which appends the argument on the right side to the argument on the left side.

Example

```
<? php
    $a = "hello ";
    $b = $a . "world!";      // now $b contains "hello world!"
    $a = "hello ";
    $a .= "world!";          // now $a contains "hello world!"
?>
```

PHP Variable

- A variable is a special container, to hold a value.
- Such value can be a number, a string, an object, an array, or a Boolean.
- Variable is a fundamental unit of any programming language.
- It is used as a placeholder which occupies specific amount of memory to store and access a value.

RULES FOR DEFINING VARIABLE IN PHP

- Any name preceded by a dollar sign ($) followed by a letter or the underscore character
- A variable name cannot start with a number
- Can include letters, numbers, and the underscore.
- Cannot include spaces or special symbols.
- Variable names are case-sensitive ($age and $AGE are two different variables)

Example

```
$a, $b;                 // Variable declaration
$a = 10;                // Assignment
$rollno = 21;           // Initialization of variable
$b = 2 * sum($a,20);    // Assignment
```

Variable Scopes

- The scope of a variable is the portion of the program within which it is defined and can be accessed.
- PHP has three types of variable scopes:
    - Local variable
    - Global variable
    - Static variable

Local Variable

- The variables that are declared within a function are called local variables for that function.
- These local variables have their scope only in that particular function in which they are declared.
- This means that these variables cannot be accessed outside the function, as they have local scope.

Global variable

- The global variables are the variables that are declared outside the function.

- These variables can be accessed anywhere in the program.
- To access the global variable within a function, use the "global" keyword before the variable.
- However, these variables can be directly accessed or used outside the function without any keyword. Therefore, there is no need to use any keyword to access a global variable outside the function.

Static variable

- It is a feature of PHP to delete the variable, once it completes its execution and memory is freed.
- Sometimes we need to store a variable even after completion of function execution. Therefore, another important feature of variable scoping is static variable.
- We use the static keyword before the variable to define a variable, and this variable is called as static variable.
- Static variables exist only in a local function, but it does not free its memory after the program execution leaves the scope. Understand it with the help of an example:

Example

```
<?php
        $gvar = 10;
        function scope()
        {
                static $svar=0;
                global $gvar;
                echo "<h4> Global Variable value = $gvar </h4>";
                $lvar = 100;
                echo "<h4> Local variable value = $lvar </h4>";
                $svar++;
                echo "<h4> static variable value = $svar </h4>";
        }
        scope();
        scope();
        scope();
        echo "<h4> Outside Global Variable value = $gvar </h4>";
?>
```

Comments in PHP

- A comment is text in a script that is ignored by the PHP engine. Comments can be used to make code more readable or to annotate a script.

Single-line comment

- Begin with two forward slashes (//) (C style)
- A single hash sign (#) (Shell style)
- The PHP engine ignores all text between these marks and either the end of the line or the PHP close tag

```
// this is a comment
# this is another comment
```

Multiline comment

- Begin with a forward slash followed by an asterisk (/*) and end with an asterisk followed by a forward slash (*/)

```
/* This is a comment
none of these lines will
be parsed by the
PHP engine.  */
```