5 Server-Side Scripting (PHP)

Let us learn

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

5.1 Introduction to PHP

PHP (Hypertext Preprocessor) is a widely-used open source general-purpose scripting language. It is especially suited for web development and can be embedded into HTML.

- PHP runs on various platforms (Linux, Unix, Mac OS X, Windows etc.).
- PHP is compatible with almost all servers used today (For eg. XAMMP, Apache, NGINX, lighttpd).
- PHP supports a wide range of databases.
- PHP is free and one can download it from the official PHP resource: www.php.net.
- PHP is easy to learn and runs efficiently on the server side.

5.2 Server Side Scripting

A server is a computer system that serves as a central repository of data and programs and is shared by the clients. The server-side environment that runs a scripting language is termed web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interfaces with databases or other data stores on the server.

Few Programming languages for server-side programming are :

1) PHP

2) Java and JSP

3) Python

PHP mainly focuses on server-side scripting, which is used to collect form data, generate dynamic page content, or send and receive cookies. There are three things to make this work : the PHP parser a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. Note : PHP hides the code from the user.

5.3 Features of PHP

PHP is most popular and frequently used world wide server-side scripting language and the main reason of popularity is :

- **Simple :** It is very simple and easy to use, as compared to other scripting languages.
- **Interpreted :** It is an interpreted language, i.e. there is no need for compilation.
- **Faster :** It is faster than other scripting language e.g. JSP and ASP.
- **Open Source :** Open source means you need not pay for use of PHP. You can freely download and use.
- **Platform Independent :** PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.
- **Case Sensitive :** PHP is case sensitive scripting language while declaration of variables and its use in the program. In PHP, all keywords (e.g. if, else, while, echo, etc.) classes, functions, and user-defined functions are NOT case-sensitive.
- **Error Reporting :** PHP have some predefined error reporting constants to generate a warning or error notice.
- **Real-Time Access Monitoring :** PHP provides access logging by creating the summary of recent accesses for the user.
- **Loosely Typed Language :** PHP supports variable usage without declaring its data type. It will be taken at the time of execution Based on the type, data has its value. Since the data types are not set in a strict sense, we can do things like adding a string to an integer without causing an error.

5.4 First sample code of PHP

A PHP file normally contains HTML tags, and some PHP scripting code. The PHP code is usually enclosed in special start and end processing instructions <?php and ?> that allow us to move into and out of "PHP mode." Even it allows to embed HTML code within PHP code.

All the PHP files have extension ".php". A PHP script can be placed anywhere in the html document. For e.g. : PHP program to display the text "Hello World!" on a web page. **Note that 'echo' is used to display text on web page**.

Program 5.1:

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

Output 5.1:

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

Do you know?

Popular PHP Frameworks

1. **Laravel :** It is used to build robust web application with MVC support. It simplifies task such as caching, security, authentication and many more.
2. **Symfony :** It is used for fast app development due to reusable components. It provides many app building blocks like Form management etc..

How to execute PHP program :

1. Type the above program and save it as"first.php"using any text editor (for e.g. notepad, gedit).

2. Create folder with your name in /var/www/html. (for e.g. balbharati)

Note : Create folder in servers root directory. In case of Linux the path of root directory is var/www/html. In case of windows the path is C:/XAMPP/htdocs.

Fig 5.1: Folder created on web server

3. Save the 'first.php' file in balbharati folder.

Fig 5.2: first.php in server directory

4. Go to browser and type "http://localhost/balbharati/"inURL bar. Click on 'first.php'.

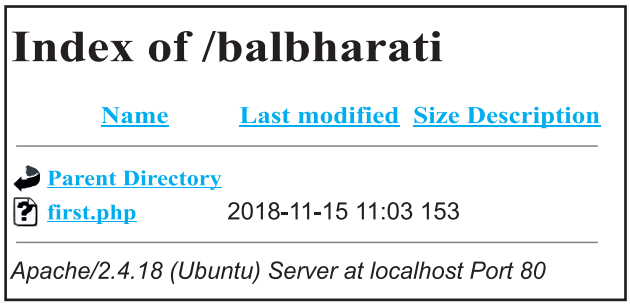


Fig 5.3: List of files on server directory

Fig 5.4 : Output of first.php program

PHP Case Sensitivity :

In PHP, the variable names are case sensitive. However keywords (e.g. if, else, break, for, while, echo etc.), function and class names are case insensitive.

For e.g. : The echo keyword is case insensitive in the program 5.2.

Program 5.2:

```
<?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
 is enclosed in echo output string.

PHP Variables :

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

Rules for declaring PHP Variable :

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the 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 _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

There are three different variable scopes in PHP:

- local
- global
- static

A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function (variable $a). And a variable declared **within** a function has a LOCAL SCOPE and can only be accessed within that function (variable $b).

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 :
```

We can access a global variable $c from within a function using

"global"keyword before the variables (inside the function).

Note : PHP also stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable

When a function is executed, then all of its variables are deleted. In some cases, if we want a local variable not to be deleted then the use of "**static**"keyword is must.

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:

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

1) String 2) Integer 3) Float 4) Boolean 5) Array 6) NULL

e.g. : One can check the data-type of variable using var_dump() method in PHP.

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)
```

var_dump() gives different output for each type of data type. It gives the length of string for 'string' data type, it gives actual digit for 'integer' data type and true/false for 'boolean' data type.

5.4.5 Comments in PHP

Comments are the statements in PHP code, which are not visible in the output of the program. It is ignored during execution. A Single-line comment is possible if one adds // or # before a statement in PHP. A multi-line comment is possible with /* and */ .

Note :

1) PHP Operators are similar to JavaScript Operators.
2) Accepting value from the user at runtime is possible through HTML Form and is covered in section 5.8

Control structures in PHP :

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

Syntax :

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

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
```

3. **Loop Structure in PHP :** Loops are used to execute the same block of code repeatedly as long as a certain condition is satisfied.
   For eg : 'For loop' in PHP

Syntax :

```
for(initialisation;condition;incrementation
or decrementation)
{
 Statement;
}
```

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
```

Use of foreach loop : This loop works only on arrays, and is used to loop through each key/value pair in an array.

Syntax :

```
foreach ($array as $value) {
   code to be executed;
}
```

Note : Use of '.' in PHP. It is used for concatenation purpose.

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.

Example : The program below shows few string manipulation functions.

5.5 PHP String Functions

A string is series of characters. The real power of PHP comes from its functions. A function is a block of statements that can be used repeatedly in a program. PHP has many built-in functions that 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.

5.6 PHP Arrays

An array is a special variable, which can hold more than one value at a time. An array stores multiple values in one single variable:

Create an Array in PHP :

In PHP, the array() function is used to create an array.

Syntax :

$a = array(values)

In PHP, there are three types of arrays:

- **Indexed arrays -** Arrays with a numeric index
- **Associative arrays -** Arrays with named keys
- **Multi-dimensional arrays -** Arrays containing one or more arrays

Indexed arrays :

The index can be assigned automatically (index always starts at 0).

Syntax :

$a = array(value1, value2, ..., value n)

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 above example, we store subject in an array at following index location.

$subjects[0] = "English";

$subjects[1] = "Hindi";

$subjects[2] = "Marathi";

The count() function is used to return the length of an array:

Note : An array can hold many values under a single name, and you can access the values by referring to an index number.

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)

Program 5.10:

```
<?php
$student_mark =
array("English"=>"75",
"Hindi" =>"64",
"Marathi"=>"88");
echo "You have scored ".$student_
mark['English']." in English .";
?>
```

Output 5.10:

```
You have scored 75 in English.
```

In above example, the values of 'student_mark' array are stored in following way:

$student_mark['English'] = "75"

$student_mark['Hindi'] = "64"

$student_mark['Marathi'] = "88"

Instead of index location, we can refer values with the help of keys. Here 'English' refers to a key & value is '75'. "=>" is used to associate key & value.

PHP Multi-dimensional Arrays :

A multidimensional array is an array containing one or more arrays. PHP can handle multiple levels of multidimensional array.

Use of foreach loop in an array

Program 5.11:

```
<?php
$subjects = array("English","Hindi",
"Marathi");
foreach ($subjects as $value) {
echo "$value <br>";
}
?>
```

Output 5.11:

```
English
Hindi
Marathi
```

In above example we have used 'foreach' loop where 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.

5.7 PHP User Defined Functions

A function is a block of statements that can be 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 always create our own functions. A user-defined function declaration starts with the word function.

Syntax :

```
function functionName() {
  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
```

PHP Function Arguments :

Information can be passed to functions through arguments. An argument is just like a variable. Arguments are specified after the function name, inside the parentheses. The following example has a function with arguments ($rollno and $name):

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.

Returning Value :

To let a function return a value, use the return statement:

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
```

5. 8 PHP Form Handling

A simple HTML form with two input fields and a submit button:

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>
```

Output 5.15.1:

When the 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:

Program 5.15.2:

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

Output 5.15.2:

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

$_POST["email"] helps us to get the data of form field whose name attribute is email. And the output is displayed as shown in output 5.15.2.

Lets understand another example with HTTP GET method.

Program 5.16.1:

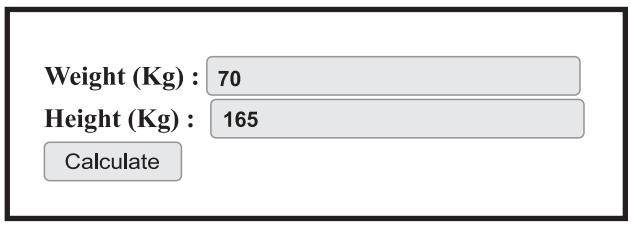
```
<html>
<head>
<title>BMI Calculator</title>
</head>
<body>
<form method="get"
action="bmioutput.php">

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:



bmioutput.php

Program 5.16.2:

```
<?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. 71162075298

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

The PHP superglobals $_GET and $_POST are used to collect form-data.

The same result could also be achieved using the HTTP GET method. Just replace POST with GET. Let us understand the difference between GET and POST Method.

GET vs POST :

Both GET and POST are treated as $_GET and $_POST superglobals, which means that they are always accessible, regardless of scope. It can be accessed from any 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.

When to use GET?

- Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL).
- GET also has limits on the amount of information to send. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.
- GET may be used for sending non-sensitive data.
- GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

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

Form connectivity with Database :

Note : Ensure you have configured 'php.ini' file before PHP & Database connectivity.

Example : Create an admission form for the registration of the college student. We have already studied HTML form creation in previous chapter and database creation in DBMS chapter. Now create"**college**"database and create table"**student**"with following fields in this Postgres SQL database.

Name	gender

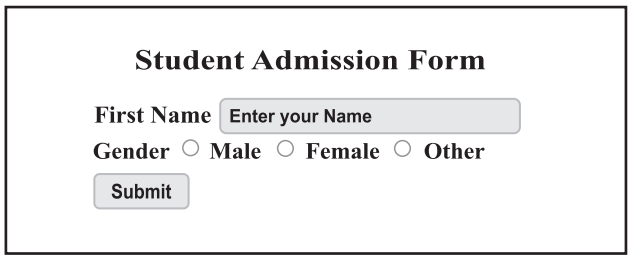
Now write following code in 'admission.php' file.

Program 5.17:

```
<!DOCTYPE html>
<head lang="en">
<title>Home</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<h1> Student Admission Form</h1>
<form  method="post" >
<label> First Name </label>
<input type="text" name="name" id="id_name" maxlength="30" required
placeholder="Enter your Name" />
</br>
<label> Gender </label>
<input type="radio" name="gender" id="id_gender" value="male"> Male
<input type="radio" name="gender" id="id_gender" value="female"> Female
<input type="radio" name="gender" id="id_gender" value="other"> Other
</br>
<input type="submit" name="submit" id="submit" value="Submit"  />
</form>
</body>
</html>
<?php
$servername = "pgsql:host=localhost;dbname=college";
$username = "postgres";
$password = "postgres";
$conn = new PDO($servername,$username , $password);

if(isset($_POST['submit']))
{
$name=$_POST["name"];
$gender=$_POST["gender"];
$sql = "INSERT INTO student (name, gender) VALUES ('".$name."', '".$gender."')";
$conn->exec($sql);
echo "New record created successfully";
}
?>
```

Output 5.17:



In above example, PHP Data Object class, i.e. PDO(), helps us to connect PHP code with almost any database server. It doesn't account for database-specific syntax, and one can switch database simply by switching the connection string in many instances. It provides database access layer providing a uniform method of access to multiple databases.

1) Create database connection object

$conn = new PDO($servername,$username, $password);

In our example, we connect PHP with PostgreSQL server. The above statement creates connectivity object '$conn' with three parameters as shown below:

1. Server host name or IP address and Database name
2. Database username
3. Database password

2) SQL statement and its execution

$sql = "INSERT INTO student (name, gender) VALUES ('".$name."', '".$gender."')";

The above statement creates SQL string to insert values in the table named 'student'.

$conn -> exec($sql);

In the above statement '->' is used to execute SQL statement using $conn as connectivity object.

Note : isset() method in PHP is used to check whether variable has value or not, this help us to know if the button is clicked or not.

The above program is used to insert data from the user into the database. Now let us understand how to fetch the data from database and display as a webpage. The next program uses PDO exception. PDO exceptions are used to handle errors, which means anything you do with PDO should be wrapped in a try/catch block. The part of code which may be erroneous can be put in try{ } block and the code to handle the error can be place in catch{ } block.

PDO exception class is used to throw errors due to database connectivity. '$e' is the object of PDO and using *getMessage()*, we can print the error message for database connectivity. *prepare($sql)* is the secure way to execute SQL queries using any programming language whereas *fetchAll()* method is used to get all the records from the given table and store in $result variable.

$conn -> null is to close the database connectivity object.

Note : For datatbase connectivity you may use any other database such as mysql, .maliadb etc.

Program 5.18:

```
<!doctype html>
<html> <body>
<h1>Display data</h1>
<?php
$servername = "pgsql:host=localhost;
dbname=college";
$username = "postgres";
$password = "postgres";
try
{
$conn = new
PDO($servername,$username ,
$password);
}
catch (PDOException $e)
{
echo "connection error".
$e->getMessage();
}
try
{
$sql1 = "select * from student";
$sqlp=$conn->prepare($sql1);
$sqlp->execute();
$result=$sqlp->fetchAll();
echo "Name -- Gender";
foreach($result as $row)
{
echo $row["name"]."--";
echo $row["gender"]."  ";
echo "</br>";
}
}
catch (PDOException $e)
{
echo "connection error".
$e->getMessage();
}
$conn=null;
?>
</body></html>
```

Output 5.18 :

Display data

Name - - Gender
raj - - male
mahika - - female
meenal - - female

Cookies and session in PHP :

Cookies :

A cookie is a small text file that the server sends on the user's computer. It is used to identify user or its machine and track activities created on the user computer. When browser requests server page, cookies are sent along with the request. PHP can be used both to create and retrieve cookie values.

Eg. Cookies store visited page on browser to optimise search.

Session :

This is used to store user information on server to track user activites. It helps web application to maintain user information on all the pages.

Eg. If you login to gmail account, the session help us to access youtube account also.

Note : For database connection use username and password values which you have given at the time of postgres installation.

Program 5.17 and 5.18 uses username as 'postgres' and password as 'postgres' while programming.

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.
- 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.
- PDO- PHP Data Object helps us to connect PHP code in uniform method of access to multiple databases.
- Cookies are sent along when browser requests server pages.
- Session helps web application to maintain user information on all the pages.

For information purpose only

PHP LAMP Install

How To Install Apache, PHP and php-pgsqlstack on Ubuntu 18.04

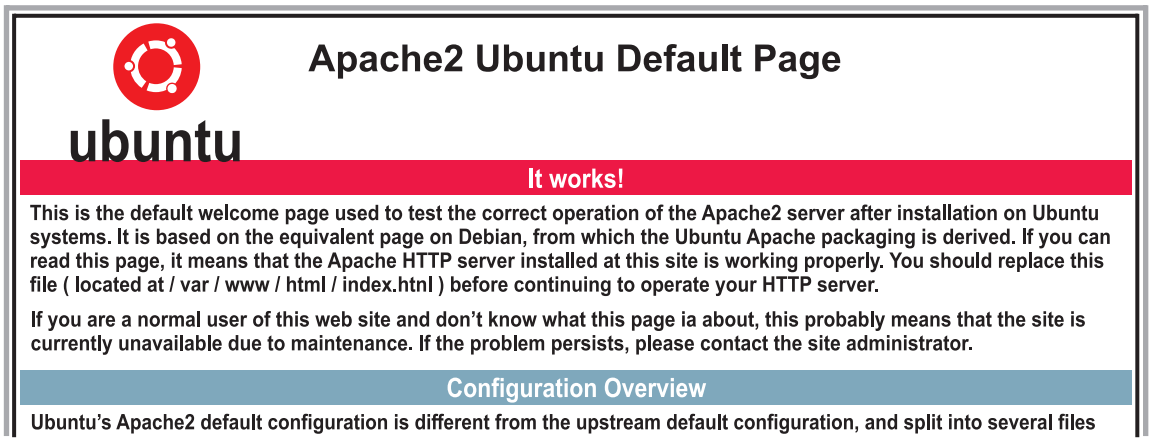
A "LAMP" stack is a group of open-source software that is typically installed together to enable a server to host dynamic websites and web apps. His term is actually an acronym which represents the Linux operating system, with the Apache web server. The site data is stored in a MySQL database, and dynamic content is processed by PHP.

Open Terminal and type following commands

Step 1 : Installing Apache

$ sudo apt update
$ sudo apt install apache2

To verify the installation, type following statement in the browser address bar.
http://your_server_ip or http://localhost/
If you get following page, its installed succesfully.



Step 2 : Installing connector for php-pgsql

$ sudo apt install php-pgsql

This command, too, will show you a list of the packages that will be installed, along with the amount of disk space they'll take up. Enter Y to continue.

Step 3 : Installing PHP

$ sudo apt install php libapache2-mod-php

After this, restart the Apache web server in order for your changes to be recognized. Do this by typing:

$sudo systemctl restart apache2

Step 4 : Testing PHP Processing on your Web Server

In order to verify PHP installation, you create a very basic PHP script called mydemo.php.

Now go to /var/www/html/. Create the file at that location. The file should contain following statement.

File Name : mydemo.php

```
<?php
echo"Hurray !!! First PHP Program is working";
?>
```

Now go to browser and type following line in the address bar.

http://your_server_ip/mydemo.php or http://localhost/mydemo.php

Step 5 : To configure Apache server for postgresql

For Ubuntu

Open 'php.ini' file located at /etc/php/x.x/apache2/php.ini and add the line below at the end of file.

extension=pgsql.so

How To Install XAMPP on Window

Step 1 : Download and install XAMPP from https://www.apachefriends.org/download.html

Step 2 : Testing PHP Processing on your Web Server

In order to verify PHP installation, you create a very basic PHP script called mydemo.php.

Now go to C:\XAMPP\htdocs\. Save the above mydemo.php file at this location. The file should contain following statement.

File Name : mydemo.php

```
<?php
echo"Hurray !!! First PHP Program is working";
?>
```

Now go to browser and type following line in the address bar.

http://your_server_ip/mydemo.php or http://localhost/mydemo.php

Note : your_server_ip is the actual IP address of your server machine which is in digits.

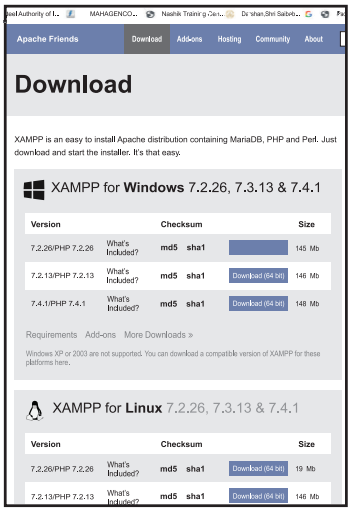
Step 3 : To configure Apache server for postgresql

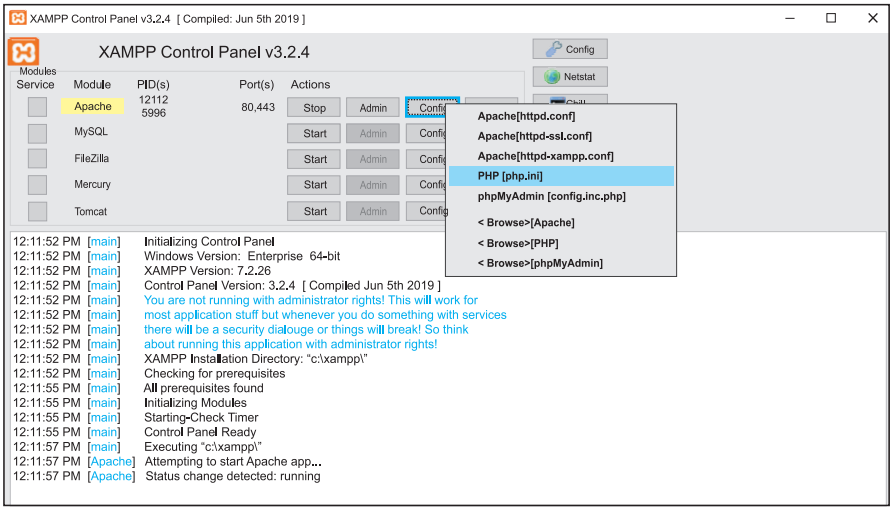
Open 'php.ini' file located at C:\xampp\php\php.ini or **click on 'config' button on the XAMPP Apache control panel.**

Now add the line below at the end of file.

extension=php_pgsql.dll

extension=php_pdo_pgsql.dll





Exercise

Q.1 Fill in the blanks.

1. PHP is _____ side scripting language.
2. PHP is _____ language i.e. there is no need for compilation.
3. A variable starts with _____ sign followed by variable name.
4. An _____ is a variable, which can hold more than one value at a time.
5. Information can be passed to functions through _____.

Q2. State True/False.

1. PHP is platform dependent scripting language.
2. $_POST is an array of variables passed via the URL parameters.
3. A Function is a block of statements that can be used repeatedly in a program.
4. PHP cannot be embeded along with HTML tags.
5. GET should NEVER be used for sending sensitive information.

Q3. Multiple Choice Question. (1 correct)

1. The program file of PHP have _____ extension.

   a) .asp b) .php
   c) .js d) .txt

2. A variable declared _____ a function has global scope.

   a) outside b) anywhere
   c) inside d) none

3. The _____ function returns a part of a string.

   a) trim() b) ucwords()
   c) substr() d) strpos()

Q4. Multiple Choice Question. (2 correct)

1. The _____ & _____ are valid datatype in PHP.

   a) Double b) Varchar
   c) Integer d) Array
   e) BigInt

2. Single line comment in PHP is possible using _____, _____.

   a) // b) /* */
   c) # d) <!-- -->
   e) $

Q5. Multiple Choice Question. (3 correct)

1. In PHP, three types of arrays are _____, _____, _____ .

   a) Indexed b) Simple
   c) Associative
   d) Multidimensional
   e) Complex f) General

2. The scope of variable can be _____.

   a) local b) global
   c) universal d) static
   e) final f) outside

Q6. Brief Questions.

1) Explain any two features of PHP?
2) What are the rules to declare variable in PHP?
3) What is server side scripting?
4) List the supported datatypes in PHP.
5) Explain any two string manipulation function.

Q7. Write Programs for the following.

1) Write a PHP code which calculates square of any number using form.
2) Write a PHP code to count no. of words in the given string.
3) Create a website with two PHP webpage in which each webpage is connected.

   The first page of the website contains two form fields for taking 'name' and 'password' from users. On onclick event, details of forms should be displayed on second webpage.

❐❐❐