PHP Introduction

**Let's Start PHP Introduction PHP** Three letters that together constitutes the name of one of the world’s most popular programming languages for Web development, the **PHP Hypertext Preprocessor**.

### **History of PHP**

PHP was originally created by a developer named Rasmus Lerdorf

|  |  |
| --- | --- |
| **Version** | **Release Date** |
| 1.0 | 8 June 1995 |
| 2.0 | 1 November 1997 |
| 3.0 | 6 June 1998 |
| 4.0 | 22 May 2000 |
| 5.0 | 13 July 2004 |
| 6.x | cancel |
| 7 | 3 December 2015(stable) |
| 7.3 | 6 December 2018(current version) |

## Advantage of PHP

If you are familiar with other server side language like ASP.NET or JSP you might be wondering what makes PHP so special, or so different from these competing alternatives well, here are some reasons:

1. Performance
2. Portability(Platform Independent)
3. Ease Of Use
4. Open Source
5. Third-Party Application Support
6. Community Support

### **Performance**

Scripts written in PHP executives faster than those written in other scripting language, with numerous independent benchmarks, putting the language ahead of competing alternatives like JSP, ASP.NET and PERL.

The PHP 5.0 engine was completely redesigned with an optimized memory manager to improve performance, and is noticeable faster than previous versions.

In addition, third party accelerators are available to further improve performance and response time.

### **Portability**

PHP is available for UNIX, MICROSOFT WINDOWS, MAC OS, and OS/2. PHP Programs are portable between platforms.

### **Ease Of Use**

“Simplicity is the ultimate sophistication”, Said Leonardo da Vinci, and by that measure, PHP is an extremely sophisticated [programming language](https://www.phptpoint.com/).

Its syntax is clear and consistent, and it comes with exhaustive documentation for the 5000+ functions included with the core distributions.

### **Open Source**

PHP is an open source project – the language is developed by a worldwide team of volunteers who make its source code freely available on the Web, and it may be used without payment of licensing fees or investments in expensive hardware or software .

## PHP Server

The PHP Community Provides Some types of Software Server solution under The GNU (General Public License).

### **These are the following:**

1. WAMP Server
2. LAMP Server
3. MAMP Server
4. XAMPP Server

All these types of software automatic configure inside operating system after installation it having PHP, MySQL, Apache and operating system base configuration file, it doesn't need to configure manually

|  |  |
| --- | --- |
| **Server** | **Stands for** |
| WAMP | Microsoft window o/s, Apache Mysql PHP |
| LAMP | Linux Operating System Apache Mysql PHP |
| MAMP | Mac os Apache Mysql PHP |
| XAMPP | x-os(cross operating system) Apache Mysql PHP Perl |

# **PHP WAMP Installation**

## WAMP Server Installation Steps

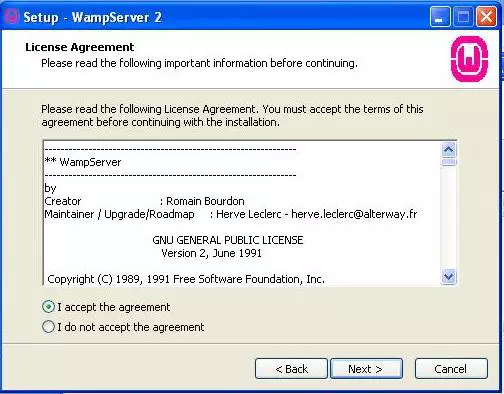
**Step:1** Download latest wamp software(64bit for 64bit os and 32bit for 32bit os)

**Step:2** click on setup

**Step:3** click on next

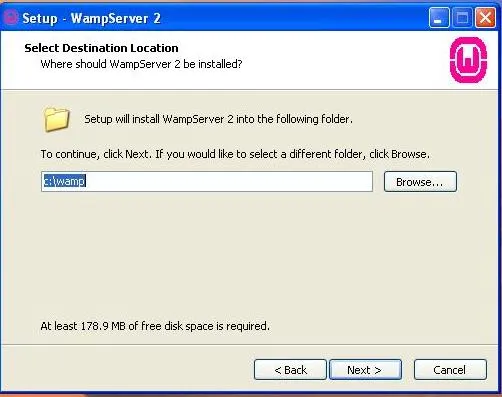


**Step:4** I accept the agreement and click on next



**Step:5** choose drive of your computer

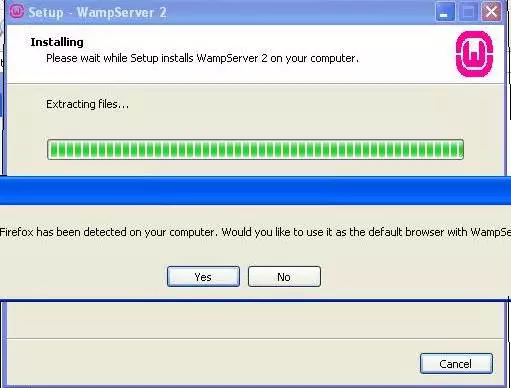
**Step:6** click on next



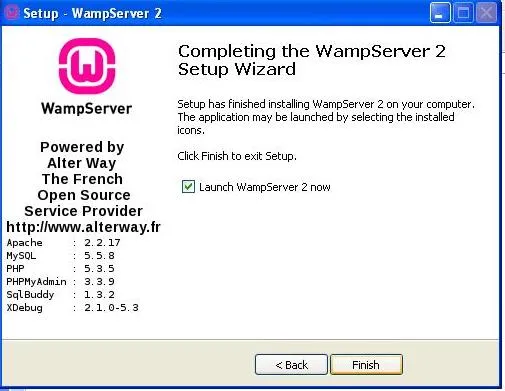
**Step:7** click on install



**Step:8** choose your browser asked if firefox found click on yes otherwise no



**Step:9** Completing the setup wizard click on finish



### **How To Execute PHP Script On WAMP Server**

**Step 1**:  First Create PHP script using any editor like notepad, notepadd++ :

<?php

echo "Welcome to the world of PHP";

?>

**Step 2**:  Save file as following... **Create a directory inside www like(myproject)** **Save it inside:**  C:wamp/www/myproject/firstProg.php **Start WAMP server** (first time only)

**Step 3**:  Run the PHP script **Open Your browser and write in url :**localhost/myproject/firstProg.php  
**Output** Welcome to the world of PHP

## XAMPP Server Installation Steps

Download the XAMPP Setup

Double click the .exe file

Choose a language from the menu,

 click OK.

Click on Next button.

## XAMPP Server Installation Steps

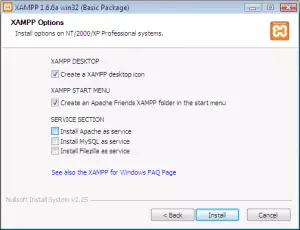
Download the XAMPP Setup

Double click the .exe file

Choose a language from the menu,

 click OK.

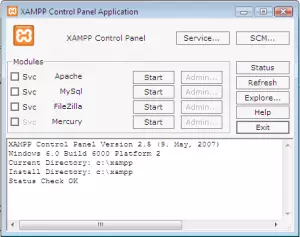
Click on Next button.



Click on Install.

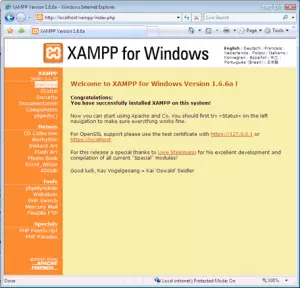
Finally, click the Finish button.

Click Yes, to open the XAMPP Control Panel (see screenshot below).



**Check Apache module and click on start for apache, check mysql module and click on start for mysql**

To Launch open your browser and type <http://localhost/>. you will got the welcome page



### **How To Execute PHP Script On XAMPP Server**

**Step 1**:  First Create PHP script using any editor like notepad, notepadd++

<?php

echo "Welcome to the world of PHP";

?>

**Step 2**:  Save file as following... **Create a directory inside xampp/htdocs/myproject/**

**Save it inside:**  C:xampp/htdocs/myproject/firstProg.php

**Start XAMPP server** (first time only)

**Step 3**:  Run the PHP script

**Open Your browser and write in url :** localhost/myproject/firstProg.php

**Output** welcome to the world of php

## PHP Environment Start - End

syntax is a way to representation of PHP script. Basically it gives the primary idea to specify the code format. It also specify the area of a written code.

**There are three ways to start PHP environment.**

### **The most commonly and effective  PHP  syntax:**

<?php

echo "Welcome to the world of php";

?>

In the given Example It begin with (< ?php) and End with(? >). echo statement is used to print the given string. Mandatory closing in("ABCD")double quotes.

Terminate the script with semicolon because semicolon is known as terminator.

### **Short or short-open tags look like this:**

<?

echo "welcome to the world of php";

?>

We use Short tags. start and end with () respectively.

Declare the statement in between (), to display the output on browser short open tags smoothly works on [**XAMPP server**](https://www.phptpoint.com/php-xampp-installation/)but Face problem on [**wamp server**](https://www.phptpoint.com/php-wamp-installation/) if you are using  WAMP Server First you have to Set the **short\_open\_tag** setting in your **php.ini** file to on

### **HTML script tags:**

<script language="PHP">

echo "welcome to the world of php";

</script>

PHP is the server side Scripting language. So HTML script is also used to start PHP environment like other scripting language.

## Comments in PHP

A comment is non-executable lines. comment is used to write description  for your own understanding.Browser doesn't read the comments.

There are two types of comments used in php

### **1. Single line comments :**

Single line comment used for short explanations. Declaration of Single line comment are two types

Either Begin with(#)

Or backslash(//)

<?php

# This is the single line comment

# This is the next line comment

// This is also a single line comment.

?>

In the above Example. \*First and second line comments begin with hash(#). \*The third one is begin with(//).

If we check the output of the given example. Browser show blank page. Because comments are always non-executable..

**another Eg of Single line Comment**

<?php

$str= "welcome ";

//$str. =" student";

echo $str;

?>

**Output** welcome

In the above Example. We declare a variable to store the String("welcome") In second line we concatenate string("student") with the Previous string("welcome") In third line we check the output.It shows Welcome Only. Because the second line statement has already specify a comment statement. So it can't take the string("student") as a declaration.

### **2. Multi-lines comments :**

Multi lines comments  used to comment multiple lines. Here we can give comments in bulk The bulk comments are enclose within (/\*.....\*/)  
<?php

/\*

This is a comment with multiline

Developer : Jagadish Babu

view : Multiline Comments Demo

\*/

?>

The all lines which is define in php evironment are Multiline comments. it is non-executable.Because it enlose with Multiline comments statement.

**another eg of Multi-line comments**

<?php

/\*

$str = "welcome ";

$str.= "users ";

\*/

echo "Hello user how are you ? ";

?>

**Output** Hello user how are you ?

## PHP Variables

**Variable** is nothing it is just name of the memory location. A Variable is simply a container i.e used to store both numeric and non-numeric information.

### **Rules for Variable declaration**

* Variables in [PHP](https://www.phptpoint.com/) starts with a **dollar($)** sign, followed by the name of the variable.
* The variable name must begin with a letter or the underscore character.
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* A variable name should not contain space

### **Assigning Values to Variables**

Assigning a value to a variable in PHP is quite east: use the equality(=) symbol, which also to the PHP's assignment operators.  
This assign value on the right side of the equation to the variable on the left.

A variable is created the moment you assign a value to it:

**Eg i**

<?php

  $myCar = "Honda";

  echo $myCar;

?>

**Output** Honda

In the above example Create a variable ($mycar)containing a string with value="Honda". To print the carname pass $mycar inside echo statement.

### **PHP Concatenation**

**Eg ii (concatenate variable with string)**

<?php

  $myCar = "Honda City";

  echo $myCar." is riding";

?>

**Output** Honda City is riding

In the above example Variable($mycar) hold value="honda city". Now we wants to concatenate variable with string. pass this variable($mycar) inside echo statement. To concatenate this with a string("is riding") use dot(.) between variable name and string. The output will be displayed : Honda City is riding

**Eg iii (Sum of two numbers)**

<?php

$first = 100;

$second = 200;

$third = $first + $second;

echo "Sum = ".$third;

?>

**Output** Sum = 300

In the above example Declare $first , $second variable with value=100, 200 respectively. Now add these two numbers using arithmetic operator ("+"). sum of these two variable result stored in a third variable($sum). Now print the sum passing ($third) with echo statement with a string.

**Eg iv (Subtraction of two numbers)**

<?php

$first = 1000;

$second = 500;

$third = $first - $second;

echo "Subtraction = ".$third;

?>

**Output** Subtraction = 500

In the above example We perform subtraction using variables( $first, $second) with vale=1000,500. Subtract second variable from first, result is hold by third variable($third) . Print this third variable passing with echo statement.

### **Destroying PHP Variables**

To destroy a variable, pass the variable to PHP's **unset( )** function. as in the following example:

**Eg v**

<?php

$name="steve";

echo $name;

//unset( ) function destroy the variable reference.

unset($name);

?>

**Output** steve

In the above example declare variable $name hold value="steve". In this program we used unset() function to delete a particular variable. first it show the output: "steve", because we pass unset function after echo statement. Now pass variable name inside unset($name) function output will show an Notice error(Variable is undefined).

**Eg vi**

<?php

$first = 100;

$second = 200;

$third = $first + $second;

echo "Sum = ".$third;

unset($third);

//after delete the variable call it again to test

echo "Sum = ".$third;

?>

**Output** Sum = 300

Sum = Notice error undefined third variable

Note : Trying to access or use a variable that's been unset( ), as in the preceding script, will result in a PHP "undefined variable" error message. This message may or may not be visible in the output page, depending on how your PHP error reporting level is configured.

### **Variable names in PHP are case-sensitive**

<?php

$name="rexx";

$NAME="rahul";

echo $name."<br/>";

echo $NAME;

?>

**Output** rexx rahul

Variable names in PHP are case-sensitive. As a result, $name refers to a different variable than does $NAME.

### **Inspecting Variable Contents(Variable Property)**

PHP offers the var\_dump( ) function, which accepts a variable and X-rays it for you. Here's an example

**Eg vii ( For String value )**

<?php

//define variables

$name = "Fiona";

$age=25;

//display variable contents

var\_dump ($name);

var\_dump($age);

?>

**Output** string 'Fiona' (length=5) int 25

In the above example We use var\_dump( ) function to check the Contents(property) of variable, $name hold a string value ="Fiona" while $age hold an integer value = 25. Now pass this variable inside var\_dump($name,$age) function . It show all information related this(data type of variable, length(only string), value)