WEB DEVELOPMENT

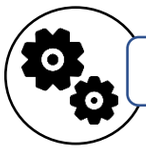
4th QUARTER MODULE

Subject Objectives

This course is designed to develop & enhance the knowledge, skills, & attitudes of a Web developer in accordance with web development industry standards. It covers the basic & common competencies in addition to the core competencies such as to develop responsive web design, utilize software methodologies, create interactive websites and develop website backend.

The Learning outcomes for the course are:

1. Identify project requirements and software methodology
2. Apply software methodologies
3. Utilize code versioning tools
4. Conduct testing
5. Perform research and analytics
6. Identify and prepare design requirements
7. Design and develop user-friendly responsive Web interface
8. Develop HTML/CSS website
9. Use/Deploy website content management system (CMS)
10. Perform search engine optimization (SEO)
11. Inspect and analyze HTML/CSS files
12. Gather and review specifications and requirements
13. Apply JavaScript to HTML/CSS
14. Configure JavaScript Efficiency
15. Develop a project plan
16. Configure a web or cloud server
17. Design databases
18. Develop server side scripts
19. Develop web application using MVC Frameworks



Subject Content

What is JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

There are following features of JavaScript:

- All popular web browsers support JavaScript as they provide built-in execution environments.
- JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
- JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
- JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
- It is a light-weighted and interpreted language.
- It is a case-sensitive language.
- JavaScript is supportable in several operating systems including, Windows, macOS, etc.
- It provides good control to the users over the web browsers.

History of JavaScript

In 1993, Mosaic, the first popular web browser, came into existence. In the year 1994, Netscape was founded by Marc Andreessen. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited Brendan Eich intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with Sun Microsystems for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascript'.

Finally, in May 1995, Marc Andreessen coined the first code of Javascript named 'Mocha'. Later, the marketing team replaced the name with 'LiveScript'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

1. Client-side validation,
2. Dynamic drop-down menus,
3. Displaying date and time,
4. Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
5. Displaying clocks etc.

Example

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

JavaScript Index

JavaScript Basics

JavaScript Comment
JavaScript Variable
JavaScript Global Variable
JavaScript Data Types
JavaScript Operators
JavaScript If Statement
JavaScript Switch
JavaScript Loop
JavaScript Function

Javascript is easy to code. JavaScript provides 3 places to put the JavaScript code: within body tag, within head tag and external JavaScript file.

3 Places to put JavaScript code

- Between the body tag of html
- Between the head tag of html
- In .js file (external javaScript)

1) JavaScript Example : code between the body tag

```
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
```

2) JavaScript Example : code between the head tag

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function_name as given below.

To call function, you need to work on event. Here we are using onclick event to call msg() function.

```
<html>
<head>
<script type="text/javascript">
function msg()
{
 alert("Hello Javatpoint");
```

```
}
</script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

External JavaScript file

We can create external JavaScript file and embed it in many html page.It provides code re usability because single JavaScript file can be used in several html pages. An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Let's create an external JavaScript file that prints Hello Javatpoint in a alert dialog box.

message.js

```
function msg(){
 alert("Hello Javatpoint");
}
```

Let's include the JavaScript file into html page. It calls the JavaScript function on button click.

index.html

```
<html>
<head>
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
```

</html>

Advantages of External JavaScript
There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.
5. The length of the code reduces as only we need to specify the location of the js file.

Disadvantages of External JavaScript
There are the following disadvantages of external files:

1. The stealer may download the coder's code using the url of the js file.
2. If two js files are dependent on one another, then a failure in one file may affect the execution of the other dependent file.
3. The web browser needs to make an additional http request to get the js code.
4. A tiny to a large change in the js code may cause unexpected results in all its dependent files.
5. We need to check each file that depends on the commonly created external javascript file.
6. If it is a few lines of code, then better to implement the internal javascript code.

JavaScript Comment

Advantage of javaScript comments

1. Single-line and Multi-line comments
2. The JavaScript comments are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

Advantages of JavaScript comments
There are mainly two advantages of JavaScript comments.

1. To make code easy to understand It can be used to elaborate the code so that end user can easily understand the code.
2. To avoid the unnecessary code It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometime, there may be need to disable the code. In such case, it is better to use comments.

There are two types of comments in JavaScript.

- Single-line Comment
- Multi-line Comment

JavaScript Single line Comment - It is represented by double forward slashes (//). It can be used before and after the statement.

Let's see the example of single-line comment i.e. added before the statement.

```
<script>
// It is single line comment
document.write("hello javascript");
</script>
<script>
var a=10;
var b=20;
var c=a+b;//It adds values of a and b variable
document.write(c);//prints sum of 10 and 20
</script>
```

JavaScript Multi line Comment - It can be used to add single as well as multi line comments. So, it is more convenient.It is represented by forward slash with asterisk then asterisk with forward slash.

For example:

```
/* your code here  */
```

It can be used before, after and middle of the statement.

```
<script>
/* It is multi line comment.
It will not be displayed */
document.write("example of javascript multiline comment");
```

</script>

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable. There are some rules while declaring a JavaScript variable (also known as identifiers).

Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign. After first letter we can use digits (0 to 9), for example value1. JavaScript variables are case sensitive, for example x and X are different variables.

Correct JavaScript variables

```
var x = 10;
var _value="sonoo";
```

Example :

```
<script>
var x = 10;
var y = 20;
var z=x+y;
document.write(z);
</script>
```

JavaScript local variable

A JavaScript local variable is declared inside block or function. It is accessible within the function or block only. For example:

```
<script>
function abc()
{
var x=10;//local variable
}
</script>
```

Or,

```
<script>
If(10<13)
{
```

```
var y=20;//JavaScript local variable
}
</script>
```

JavaScript global variable

A JavaScript global variable is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. For example:

```
<script>
var data=200;//gloabal variable
function a(){
document.writeln(data);
}
function b(){
document.writeln(data);
}
a();//calling JavaScript function
b();
</script>
```

Javascript Data Types - JavaScript provides different data types to hold different types of values. JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use var here to specify the data type. It can hold any type of values such as numbers, strings etc. There are two types of data types in JavaScript.

- Primitive data type
- Non-primitive (reference) data type

For example:

```
var a=40;//holding number
var b="Rahul";//holding string
```

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description

String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

The non-primitive data types are as follows:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands.

Example:

```
var sum=10+20;
```

Operators in Javascript

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

JavaScript Arithmetic Operators - Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

JavaScript Comparison Operators - The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false
!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

JavaScript Bitwise Operators - The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
|	Bitwise OR	(10==20 | 20==33) = false
^	Bitwise XOR	(10==20 ^ 20==33) = false
~	Bitwise NOT	(~10) = -10
<<	Bitwise Left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise Right Shift with Zero	(10>>>2) = 2

JavaScript Logical Operators - The following operators are known as JavaScript logical operators.

Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false
||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	!(10==20) = true

JavaScript Assignment Operators - The following operators are known as JavaScript assignment operators.

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

JavaScript Special Operators - The following operators are known as JavaScript special operators.

Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

JavaScript Conditional Statements

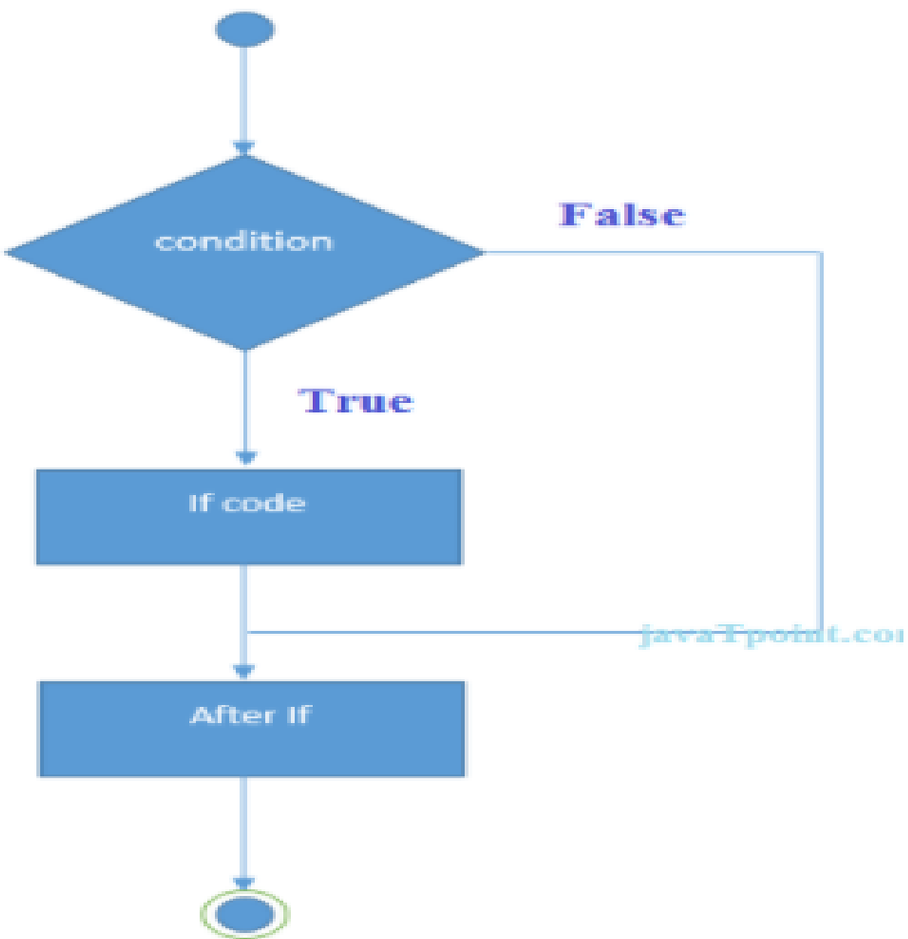
The JavaScript if-else statement is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.

- If Statement
- If else statement
- if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression){
//content to be evaluated
}
```

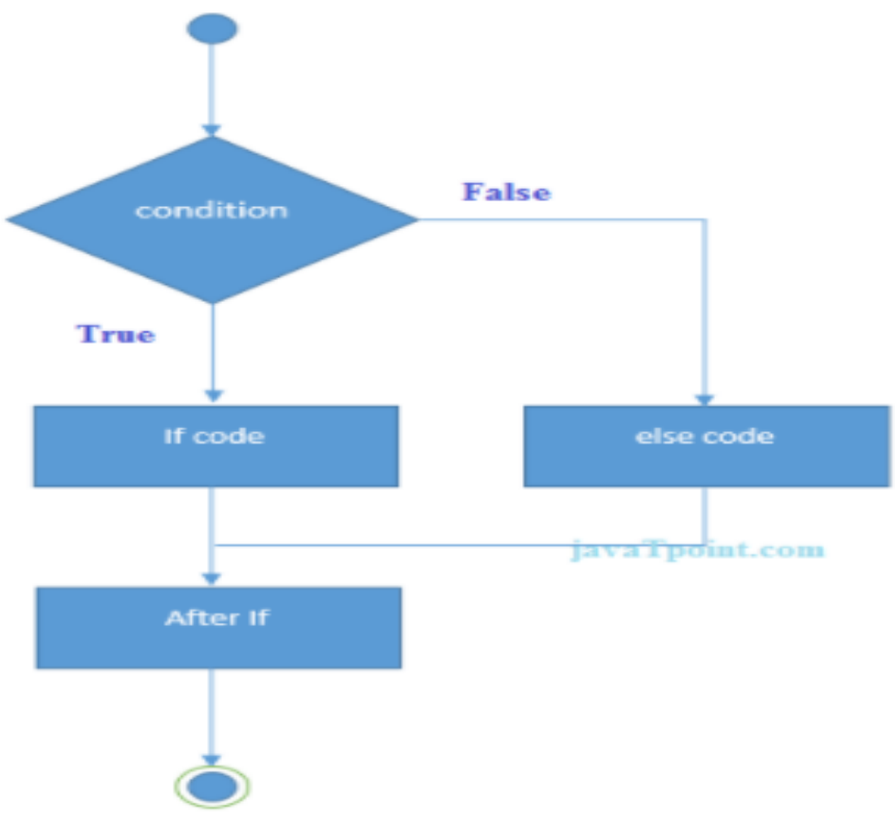


Example

```
<script>
var a=20;
if(a>10)
{
document.write("value of a is greater than 10");
}
</script>
```

JavaScript If...else Statement - It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression)
{
//content to be evaluated if condition is true
}
else
{
//content to be evaluated if condition is false
}
```



Example

```
<script>
var a=20;
if(a%2==0)
{
document.write("a is even number");
}
else
{
document.write("a is odd number");
}
</script>
```

JavaScript If...else if statement - It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1)
{
//content to be evaluated if expression1 is true
}
else if(expression2)
{
//content to be evaluated if expression2 is true
}
else if(expression3)
{
//content to be evaluated if expression3 is true
}
else
{
//content to be evaluated if no expression is true
}
```

Example

```
<script>
var a=20;
if(a==10)
{
document.write("a is equal to 10");
```

```
}
else if(a==15
){
document.write("a is equal to 15");
}
else if(a==20)
{
document.write("a is equal to 20");
}
else
{
document.write("a is not equal to 10, 15 or 20");
}
</script>
```

JavaScript Loops - The JavaScript loops are used to iterate the piece of code using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

- for loop
- while loop
- do-while loop
- for-in loop

1) JavaScript For loop

The JavaScript for loop iterates the elements for the fixed number of times. It should be used if number of iteration is known. The syntax of for loop is given below.

```
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")
}
</script>
```

JavaScript while loop - The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known. The syntax of while loop is given below.

```
<script>
var i=11;
while (i<=15)
{
document.write(i + "<br/>");
i++;
}
</script>
```

JavaScript do while loop - iterates the elements for the infinite number of times like while loop. But, code is executed at least once whether condition is true or false. The syntax of do while loop is given below.

```
<script>
var i=21;
do{
document.write(i + "<br/>");
i++;
}while (i<=25);
</script>
```

JavaScript Functions - are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

- Code reusability: We can call a function several times so it save coding.
- Less coding: It makes our program compact. We don't need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

```
function functionName([arg1, arg2, ...argN])
{
 //code to be executed
}
```

(***Note that the JavaScript Functions can have 0 or more arguments.***)

Example

```
<script>
function msg()
{
alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
```

JavaScript Function Arguments - We can call function by passing arguments. Let's see the example of function that has one argument.

```
<script>
function getcube(number){
alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
```

Function with Return Value - We can call function that returns a value and use it in our program. Let's see the example of function that returns value.

```
<script>
function getInfo()
{
return "hello javatpoint! How r u?";
}
</script>
<script>
document.write(getInfo());
</script>
```

JavaScript Function Object - In JavaScript, the purpose of Function constructor is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

Syntax

```
new Function ([arg1[, arg2[, ....argn]],] functionBody)
```

Parameter

arg1, arg2, ... , argn - It represents the argument used by function.

functionBody - It represents the function definition.

Method	Description
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
call()	It is used to call a function contains this value and an argument list.
toString()	It returns the result in a form of a string.

JavaScript Array - JavaScript array is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

1) JavaScript array literal - The syntax of creating array using array literal is given as:

var arrayname=[value1,value2.....valueN];

As you can see, values are contained inside [ ] and separated by , (comma).

Let's see the simple example of creating and using array in JavaScript.

```
<script>
var emp=["Sonoo","Vimal","Ratan"];
for (i=0;i<emp.length;i++)
{
document.write(emp[i] + "<br/>");
}
</script>
```

2) JavaScript Array directly (new keyword)

var arrayname=new Array();

Example

```
<script>
var i;
var emp = new Array();
emp[0] = "Arun";
emp[1] = "Varun";
emp[2] = "John";

for (i=0;i<emp.length;i++)
{
document.write(emp[i] + "<br>");
}
</script>
```

3) JavaScript array constructor (new keyword) - you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Example

```
<script>
var emp=new Array("Jai","Vijay","Smith");
for (i=0;i<emp.length;i++)
{
document.write(emp[i] + "<br>");
}
</script>
```

What is PHP?

PHP (Hypertext Preprocessor) is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

PHP was created by **Rasmus Lerdorf in 1994** but appeared in the market in 1995. **PHP 7.4.0** is the latest version of PHP, which was released on **28 November**.

Some important points need to be noticed about PHP are as followed:

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.

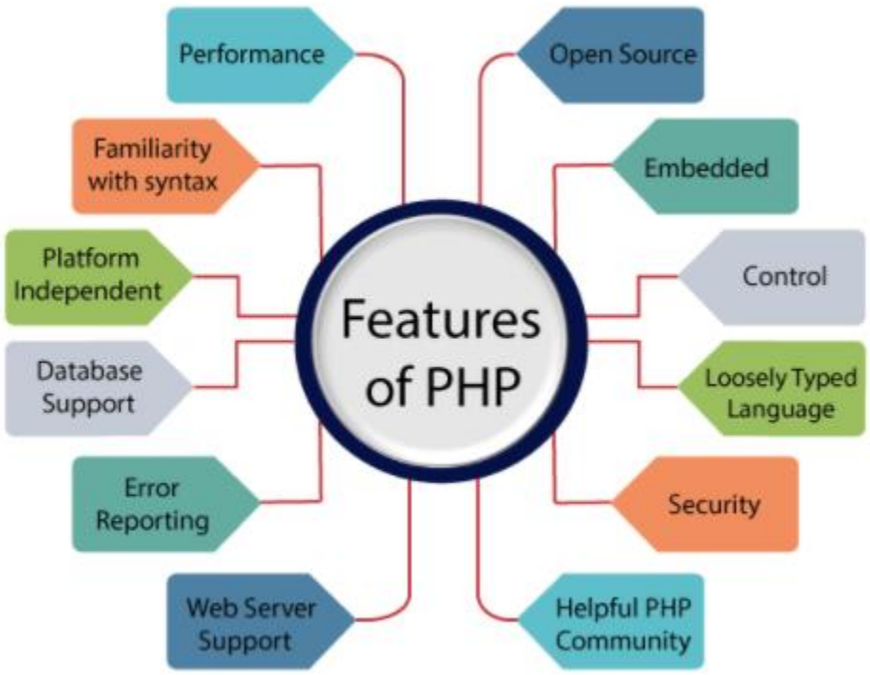
PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.

- It handles dynamic content, database as well as session tracking for the website.
- You can create sessions in PHP.
- It can access cookies variable and also set cookies.
- It helps to encrypt the data and apply validation.

- PHP supports several protocols such as HTTP, POP3, SNMP, LDAP, IMAP, and many more.
- Using PHP language, you can control the user to access some pages of your website.
- As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.
- PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. **For example** - Registration form.

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.

Web Development - PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well.

HTML
CSS
JavaScript
Ajax
XML and JSON
jQuery

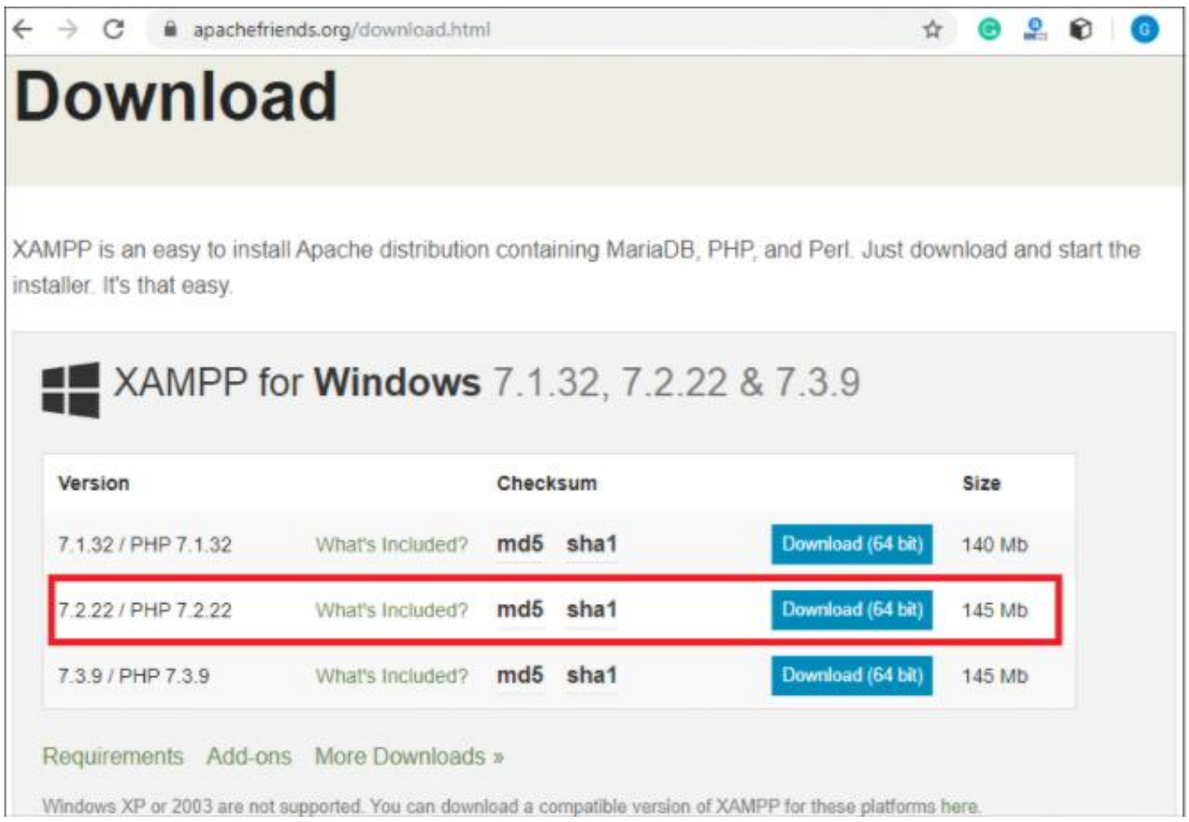
To install PHP, we will suggest you to install AMP (Apache, MySQL, PHP) software stack. It is available for all operating systems. There are many AMP options available in the market that are given below:

- WAMP for Windows
- LAMP for Linux
- MAMP for Mac
- SAMP for Solaris
- FAMP for FreeBSD
- XAMPP (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, Mercury Mail, etc.

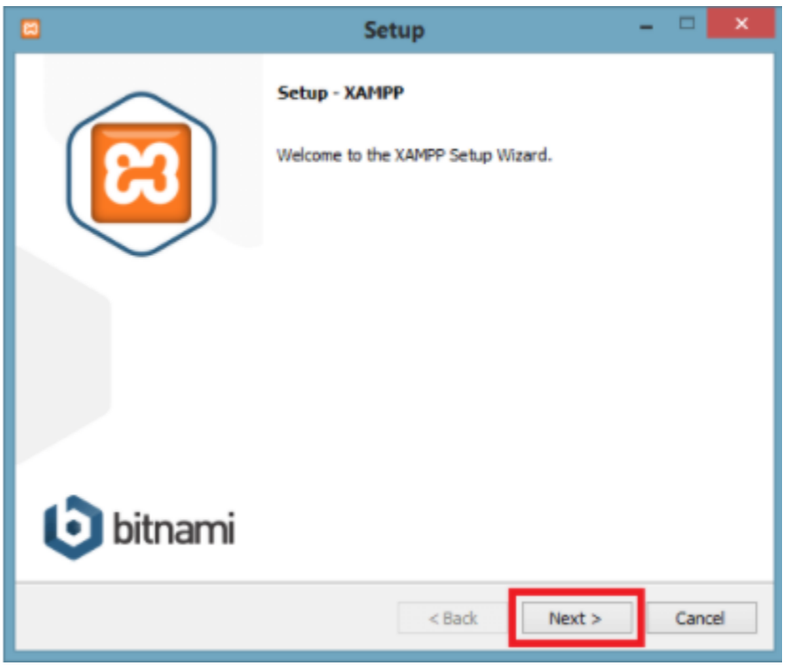
If you are on Windows and don't want Perl and other features of XAMPP, you should go for WAMP. In a similar way, you may use LAMP for Linux and MAMP for Macintosh.

How to install XAMPP server on windows
We will learn how to install the XAMPP server on windows platform step by step. Follow the below steps and install the XAMPP server on your system.

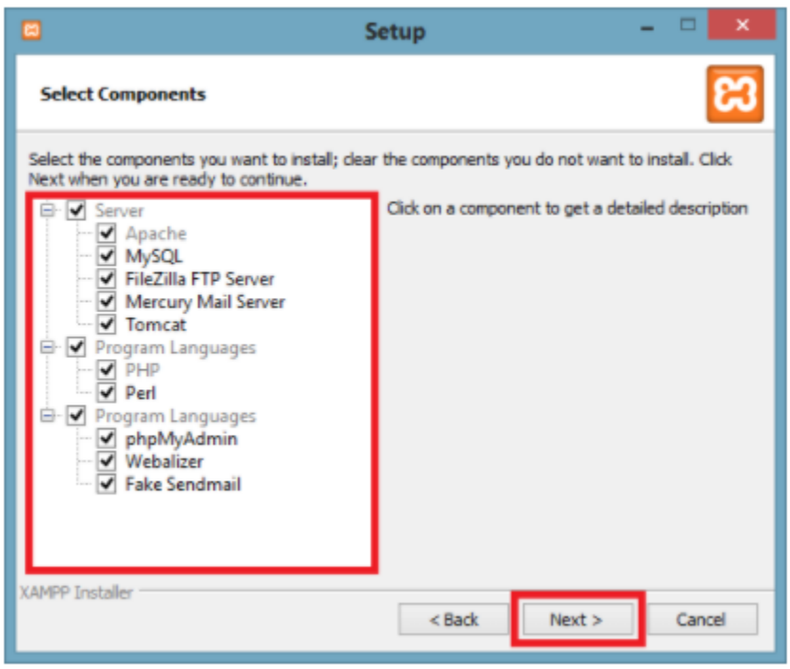
Step 1: Click on the above link provided to download the **XAMPP server** according to your window requirement.



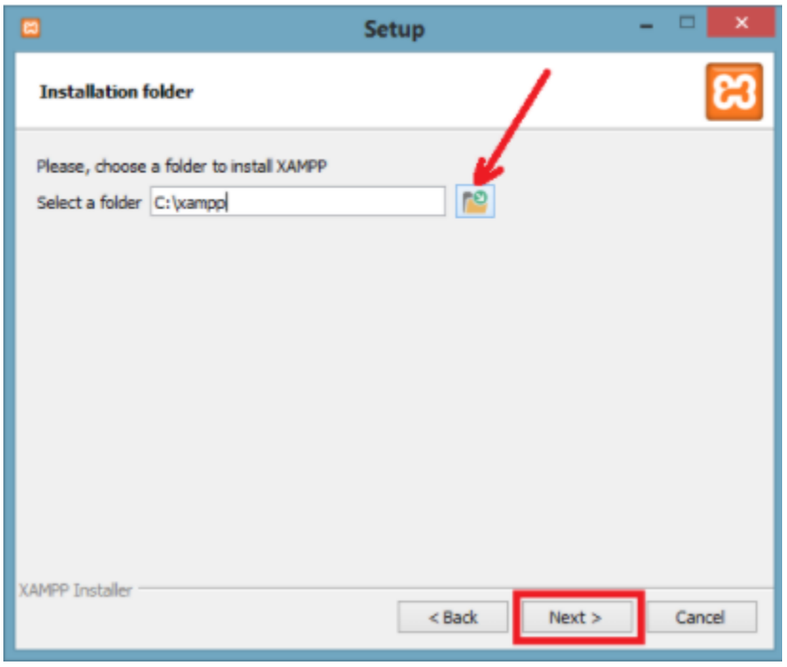
Step 2: After downloading **XAMPP**, double click on the downloaded file and allow **XAMPP** to make changes in your system. A window will pop-up, where you have to click on the Next button.



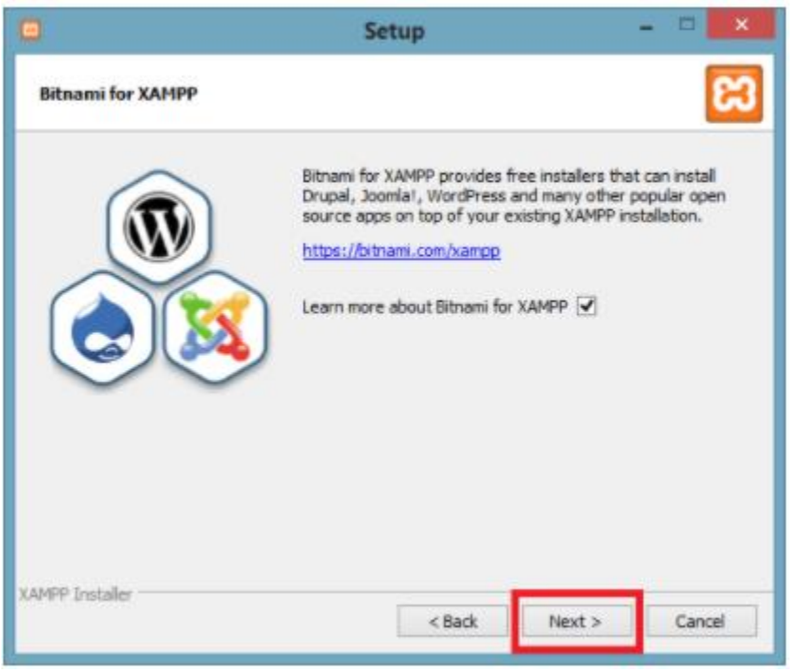
Step 3: Here, select the components, which you want to install and click **Next**.



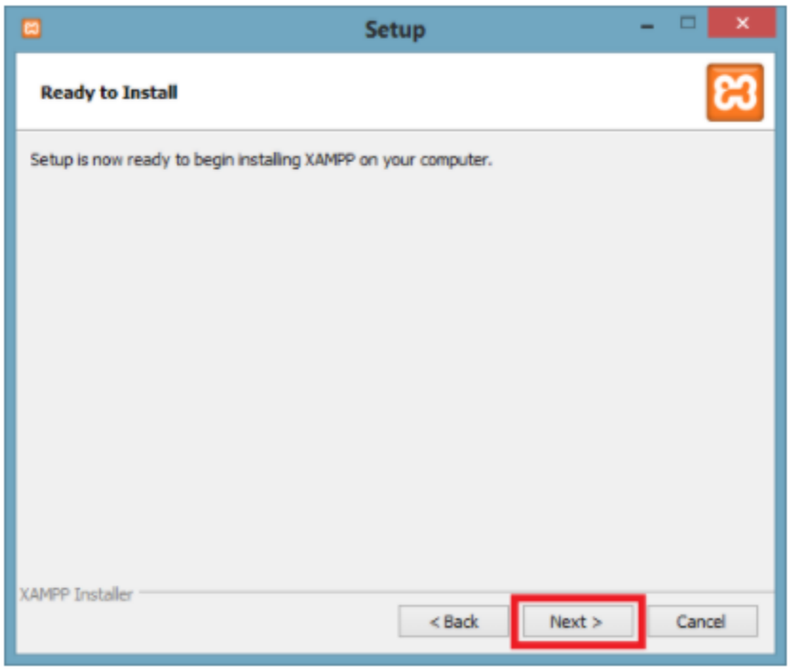
Step 4: Choose a folder where you want to install the XAMPP in your system and click **Next**.



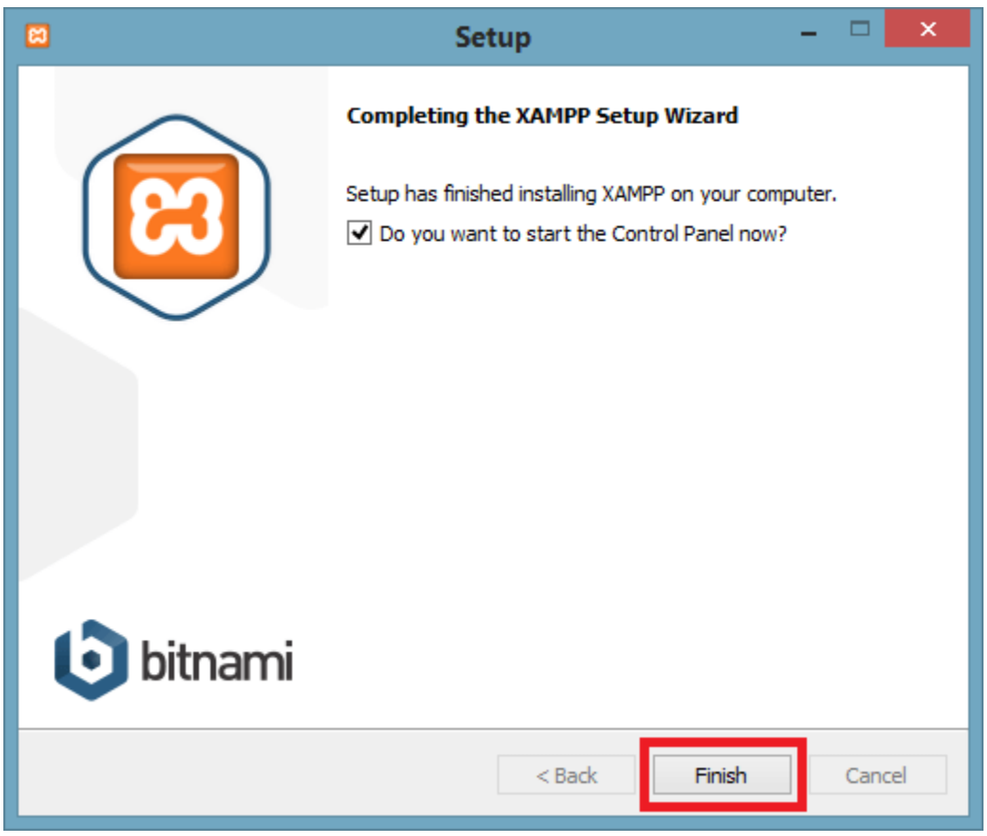
Step 5: Click **Next** and move ahead.



Step 6: XAMPP is ready to install, so click on the **Next** button and install the XAMPP.

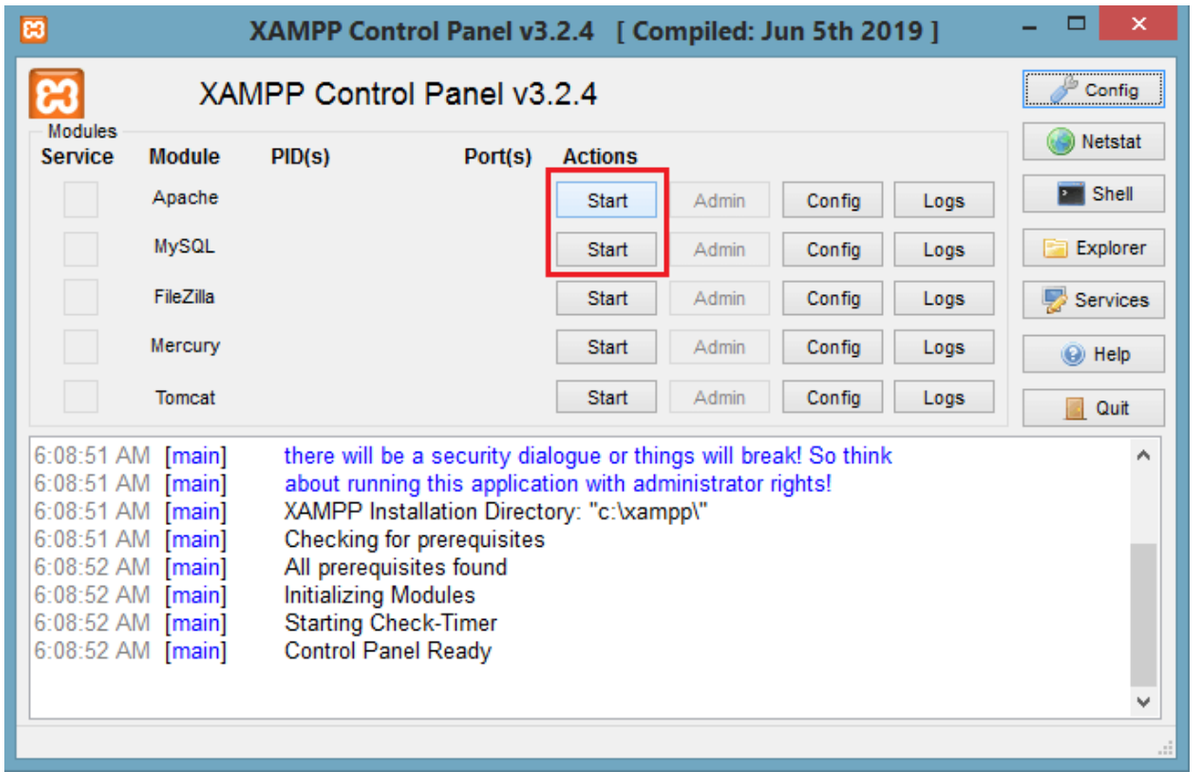


Step 7: A finish window will display after successful installation. Click on the **Finish** button.

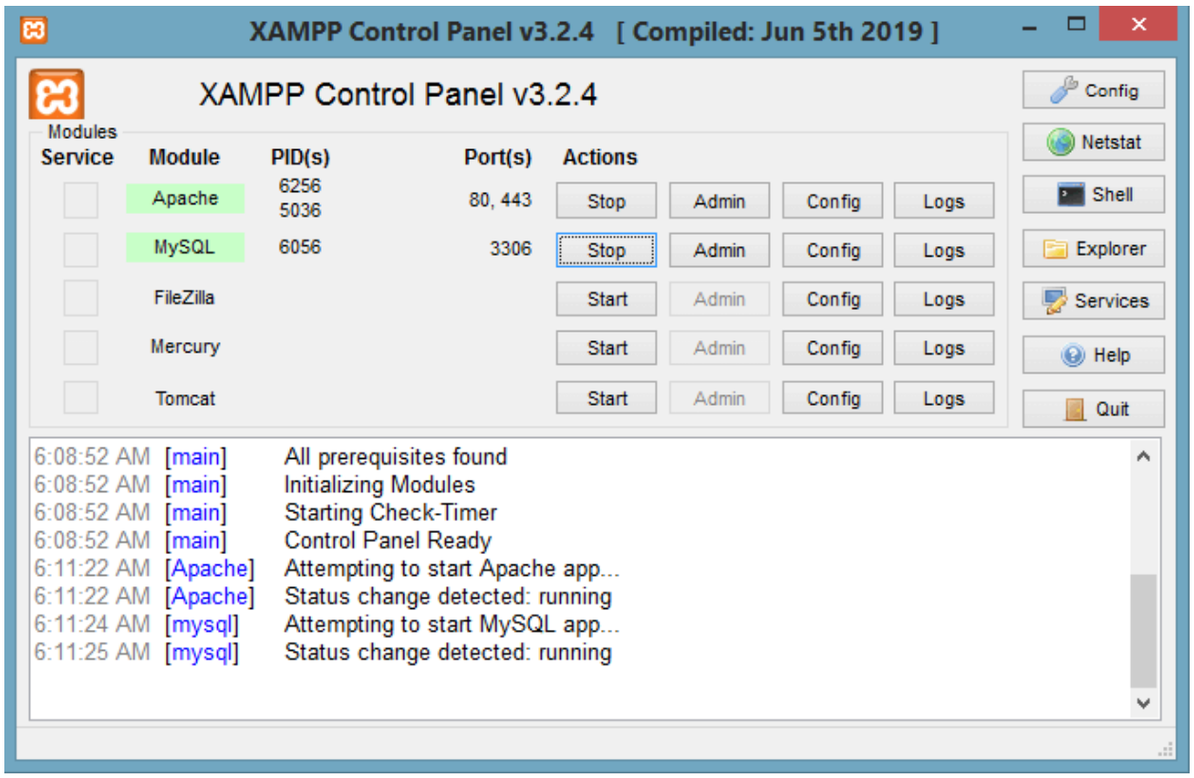


Step 8: Choose your preferred language.

Step 9: XAMPP is ready to use. Start the Apache server and MySQL and run the php program on the localhost.



Step 10: If no error is shown, then XAMPP is running successfully.



How to run PHP code in XAMPP

Generally, a PHP file contains HTML tags and some PHP scripting code. It is very easy to create a simple PHP example. To do so, create a file and write HTML tags + PHP code and save this file with .php extension.

Note: PHP statements ends with semicolon (;).
All PHP code goes between the php tag. It starts with <?php and ends with ?>. The syntax of PHP tag is given below:

```
<?php
//your code here
?>
```

Let's see a simple PHP example where we are writing some text using PHP echo command.

File: first.php

```
<!DOCTYPE>
<html>
<body>
```

```
<?php
echo "<h2>Hello First PHP</h2>";
?>
</body>
</html>
```

Output:

Hello First PHP

How to run PHP programs in XAMPP

How to run PHP programs in XAMPP PHP is a popular backend programming language. PHP programs can be written on any editor, such as - Notepad, Notepad++, Dreamweaver, etc. These programs save with **.php** extension, i.e., filename.php inside the htdocs folder.

For example - p1.php.

As I'm using window, and my XAMPP server is installed in D drive. So, the path for the htdocs directory will be "D:\xampp\htdocs".

PHP program runs on a web browser such as - Chrome, Internet Explorer, Firefox, etc. Below some steps are given to run the PHP programs.

Step 1: Create a simple PHP program like hello world.

```
1. <?php
2.     echo "Hello World!";
3. ?>
```

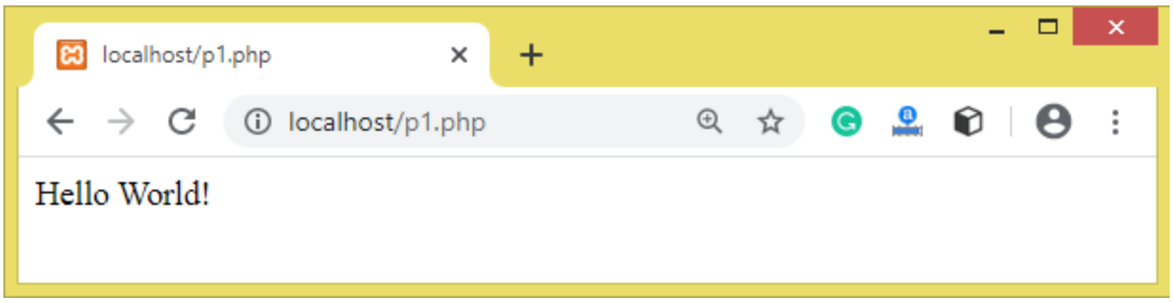
Step 2: Save the file with **hello.php** name in the htdocs folder, which resides inside the xampp folder.

Note: PHP program must be saved in the htdocs folder, which resides inside the xampp folder, where you installed the XAMPP. Otherwise it will generate an error - Object not found.

Step 3: Run the XAMPP server and start the Apache and MySQL.

Step 4: Now, open the web browser and type localhost *http://localhost/hello.php* on your browser window.

Step 5: The output for the above **hello.php** program will be shown as the screenshot below:



Most of the time, PHP programs run as a web server module. However, PHP can also be run on CLI (Command Line Interface).

PHP Case Sensitivity

In PHP, keyword (e.g., echo, if, else, while), functions, user-defined functions, classes are not case-sensitive. However, all variable names are case-sensitive.

In the below example, you can see that all three echo statements are equal and valid:

```
<!DOCTYPE>
<html>
   <body>
      <?php
         echo "Hello world using echo </br>";
         ECHO "Hello world using ECHO </br>";
         EcHo "Hello world using EcHo </br>";
      ?>
   </body>
</html>
```

Output:

Hello world using echo
Hello world using ECHO
Hello world using EcHo

Look at the below example that the variable names are case sensitive. You can see the example below that only the second statement will display the value of the $color variable. Because it treats $color, $ColoR, and $COLOR as three different variables:

```
<html>
   <body>
      <?php
         $color = "black";
```

```
        echo "My car is ". $ColoR ."</br>";
        echo "My dog is ". $color ."</br>";
        echo "My Phone is ". $COLOR ."</br>";
    ?>
  </body>
</html>
```

Output:

Notice: Undefined variable: ColoR in D:\xampp\htdocs\program\p2.php on line 8

My car is
My dog is black

Notice: Undefined variable: COLOR in D:\xampp\htdocs\program\p2.php on line 10
My Phone is

Only $color variable has printed its value, and other variables $ColoR and $COLOR are declared as undefined variables. An error has occurred in line 5 and line 7.

PHP echo is a language construct, not a function. Therefore, you don't need to use parenthesis with it. But if you want to use more than one parameter, it is required to use parenthesis.

The syntax of PHP echo is given below:

```
void echo ( string $arg1 [, string $... ] )
```

PHP echo statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

- echo is a statement, which is used to display the output.
- echo can be used with or without parentheses: echo(), and echo.
- echo does not return any value.
- We can pass multiple strings separated by a comma (,) in echo.
- echo is faster than the print statement.

PHP echo: printing string

File: echo1.php

```
<?php
echo "Hello by PHP echo";
?>
```

Output

```
Hello by PHP echo
```

File: echo2.php

```
<?php
echo "Hello by PHP echo
this is multi line
text printed by
PHP echo statement
";
?>
```

Output:

```
Hello by PHP echo this is multi line text printed by PHP echo statement
```

PHP echo: printing escaping characters

File: echo3.php

```
<?php
echo "Hello escape \"sequence\" characters";
?>
```

Output:

```
Hello escape "sequence" characters
```

PHP echo: printing variable value
File: echo4.php

```
<?php
$msg="Hello JavaTpoint PHP";
echo "Message is: $msg";
?>
```

Output:

```
Message is: Hello JavaTpoint PHP
```

PHP Print - Like PHP echo, PHP print is a language construct, so you don't need to use parenthesis with the argument list. Print statement can be used with or without parentheses: print and print(). Unlike echo, it always returns 1.

The syntax of PHP print is given below:

```
int print(string $arg)
```

PHP print statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

- print is a statement, used as an alternative to echo at many times to display the output.
- print can be used with or without parentheses.
- print always returns an integer value, which is 1.
- Using print, we cannot pass multiple arguments.
- print is slower than the echo statement.

PHP print: printing string

File: print1.php

```
<?php
print "Hello by PHP print ";
print ("Hello by PHP print()");
?>
```

Output:

```
Hello by PHP print Hello by PHP print()
```

PHP print: printing multi line string

File: print2.php

```
<?php
print "Hello by PHP print
this is multi line
text printed by
PHP print statement
";
?>
```

Output:

```
Hello by PHP print this is multi line text printed by PHP print
statement
```

PHP print: printing escaping characters

File: print3.php

```
<?php
print "Hello escape \"sequence\" characters by PHP print";
?>
```

Output:

```
Hello escape "sequence" characters by PHP print
```

PHP print: printing variable value

File: print4.php

```
<?php
$msg="Hello print() in PHP";
print "Message is: $msg";
```

```
?>
```

Output:

```
Message is: Hello print() in PHP
```

PHP echo and print Statements - We frequently use the echo statement to display the output. There are two basic ways to get the output in PHP:

- echo
- print

echo and print are language constructs, and they never behave like a function. Therefore, there is no requirement for parentheses. However, both the statements can be used with or without parentheses. We can use these statements to output variables or strings.

Difference between echo and print

echo

- echo is a statement, which is used to display the output.
- echo can be used with or without parentheses.
- echo does not return any value.
- We can pass multiple strings separated by comma (,) in echo.
- echo is faster than print statement.

print

- print is also a statement, used as an alternative to echo at many times to display the output.
- print can be used with or without parentheses.
- print always returns an integer value, which is 1.
- Using print, we cannot pass multiple arguments.
- print is slower than echo statement.

You can see the difference between echo and print statements with the help of the following programs.

PHP echo and print Statements - We frequently use the echo statement to display the output. There are two basic ways to get the output in PHP:

- echo
- print
-

echo and print are language constructs, and they never behave like a function. Therefore, there is no requirement for parentheses. However, both the statements can be used with or without parentheses. We can use these statements to output variables or strings.

Difference between echo and print

echo

- echo is a statement, which is used to display the output.
- echo can be used with or without parentheses.
- echo does not return any value.
- We can pass multiple strings separated by comma (,) in echo.
- echo is faster than print statement.

print

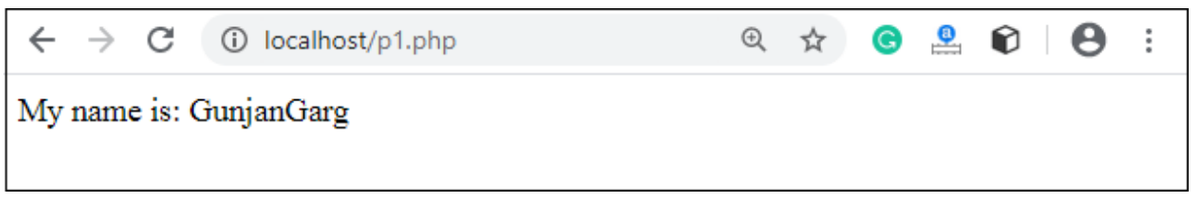
- print is also a statement, used as an alternative to echo at many times to display the output.
- print can be used with or without parentheses.
- print always returns an integer value, which is 1.
- Using print, we cannot pass multiple arguments.
- print is slower than echo statement.

You can see the difference between echo and print statements with the help of the following programs.

Example (Check multiple arguments)

```
<?php
$fname = "Gunjan";
$lname = "Garg";
echo "My name is: ".$fname,$lname;
?>
```

Output:



PHP echo and print Statements

It will generate a syntax error because of multiple arguments in a print statement.

```
<?php
$fname = "Gunjan";
$lname = "Garg";
print "My name is: ".$fname,$lname;
?>
```

Output:

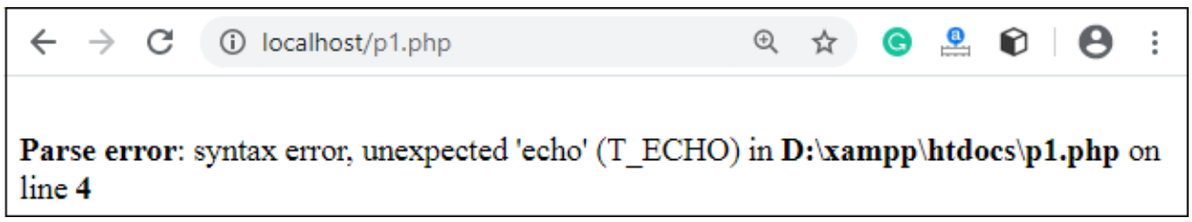
PHP echo and print Statements

Example (Check Return Value)

echo statement does not return any value. It will generate an error if you try to display its return value.

```
<?php
$lang = "PHP";
$ret = echo $lang." is a web development language.";
echo "</br>";
echo "Value return by print statement: ".$ret;
?>
```

Output:

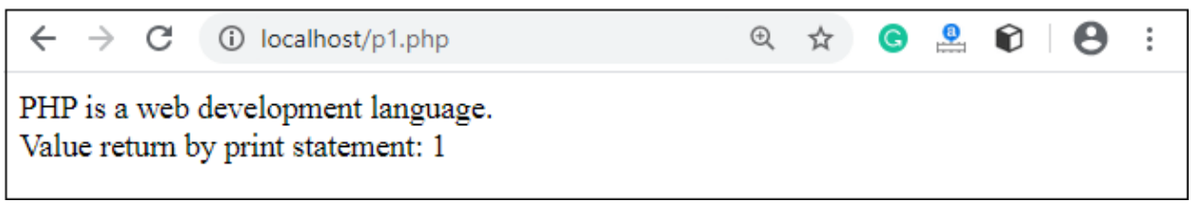


PHP echo and print Statements

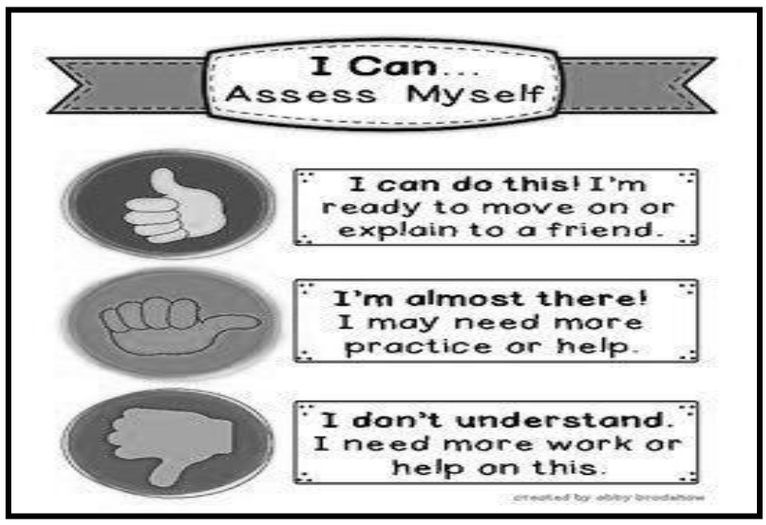
As we already discussed that print returns a value, which is always 1.

```
<?php
$lang = "PHP";
$ret = print $lang." is a web development language.";
print "</br>";
print "Value return by print statement: ".$ret;
?>
```

Output:



Self-Reflection

Encircle your answer	

FORM

Read each statement and check (✓) the box that reflects your work today.

Name:		Date:		
Section:				
	Strongly Agree	Agree	Disagree	Strongly Disagree
1. I found this work interesting.				
2. I make a strong effort.				
3. I am proud of the results.				
4. I understood all the instructions.				
5. I followed all the steps.				
6. I learned something new.				
7. I feel ready for the next assignment.				

www.ldatschool.ca/executive-function/self-assessment/