

CVWDD 102

Advance Web Development

School of Vocational Studies



उत्तराखण्ड मुक्त विश्वविद्यालय

तीनपानी बाईपास रोड, ट्रांसपोर्ट नगर के पास, हल्द्वानी- 263139

फोन न.- 05946 - 261122, 261123

टॉल फ्री न.- 18001804025

फैक्स न.- 05946-264232, ई-मेल- info@uou.ac.in

वेबसाइट- www.uou.ac.in

C. Voc. (Web Designing & Development)

CVWDD 102 Advance Web Development

INDEX

UNIT	DESCRIPTION	PAGE NO.
1	JAVASCRIPT	
	INTRODUCTION OBJECTIVES ADD JAVASCRIPT IN YOUR HTML PAGE PROGRAMMING VS. SCRIPTING TOKENS IN JAVASCRIPT BASIC PROGRAMMING CONSTRUCTS IN JAVASCRIPT ARRAYS IN JAVASCRIPT OBJECTS IN JAVASCRIPT POINTS TO REMEMBER GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	6-34
2	JAVASCRIPT EVENTS	
	INTRODUCTION OBJECTIVE JAVASCRIPT EVENTS JAVASCRIPT OBJECTS AND PROPERTIES CLIENT-SIDE VALIDATIONS COOKIES DHTML POINTS TO REMEMBER GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	35-58
3	JQUERY	
	INTRODUCTION OBJECTIVE JQUERY: BASIC STRUCTURE AND ACCESSING RULE HOW TO USE JQUERY JQUERY SYNTAX HOW TO WRITE FUNCTIONS IN A SEPARATE FILE ACCESSING JQUERY LIBRARIES HOW TO USE CUSTOM SCRIPTS POINTS TO REMEMBER GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	59-74
4	INTRODUCTION TO AJAX	
	INTRODUCTION	75-89

	<p>OBJECTIVE</p> <p>WHAT YOU NEED TO USE AJAX?</p> <p>WHAT IS AJAX?</p> <p>XML– A BRIEF INTRODUCTION</p> <p>“XMLHTTPREQUEST” OBJECT</p> <p>AJAX REQUEST</p> <p>AJAX RESPONSE</p> <p>SAMPLE WEB-PAGE USING AJAX – CALLING DATA FROM TEXT FILE</p> <p>SAMPLE WEB-PAGE USING AJAX – CALLING DATA FROM XML FILE</p> <p>AJAX SECURITY</p> <p>POINTS TO REMEMBER</p> <p>GLOSSARY</p> <p>CHECK YOUR PROGRESS</p> <p>BIBLIOGRAPHY/REFERENCES</p> <p>SUGGESTED READINGS</p>	
5	SERVER CONFIGURATION FOR SERVER-SIDE PROGRAMMING & INTRODUCTION TO PHP	
	<p>INTRODUCTION</p> <p>OBJECTIVE</p> <p>PREREQUISITE</p> <p>WHAT IS XAMPP?</p> <p>INTRODUCTION TO PHP</p> <p>BASIC PROGRAMMING COSTRUCTS IN PHP</p> <p>POINTS TO REMEMBER</p> <p>GLOSSARY</p> <p>CHECK YOUR PROGRESS</p> <p>BIBLIOGRAPHY/REFERENCES</p> <p>SUGGESTED READINGS</p>	90-107
6	INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT	
	<p>INTRODUCTION</p> <p>OBJECTIVE</p> <p>HISTORY OF MOBILE COMMUNICATION DEVELOPMENT</p> <p>WHAT IS SMARTPHONE AND MOBILE APPLICATION?</p> <p>WHY WE NEED MOBILE APPLICATIONS?</p> <p>PROGRAMMING LANGUAGES USED IN MOBILE APPLICATIONS</p> <p>OPERATING SYSTEM USED IN MOBILE PHONES</p> <p>MOBILE APPLICATION DEVELOPMENT PLATFORM</p> <p>CREATING HELLO WORLD APP USING PHONEGAP</p> <p>POINTS TO REMEMBER</p> <p>GLOSSARY</p> <p>CHECK YOUR PROGRESS</p> <p>BIBLIOGRAPHY/REFERENCES</p> <p>SUGGESTED READINGS</p>	108-125
7	PHP LANGUAGE CONSTRUCTS	
	<p>INTRODUCTION</p> <p>OBJECTIVES</p> <p>PHP CONDITIONAL STATEMENTS</p> <p>LOOPING IN PHP</p> <p>PHP FUNCTIONS</p> <p>PHP STRINGS</p> <p>PHP SECURITY</p> <p>POINTS TO REMEMBER</p>	126-139

	GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	
8	ADVANCE PHP	
	INTRODUCTION OBJECTIVES PREREQISITE OPERATIONS ON DATE & TIME IN PHP IMAGE UPLOADING USING PHP FILE HANDLING IN PHP ERROR HANDLIG IN PHP MINI PROJECT IN PHP POINTS TO REMEMBER GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	140-154
9	MYSQL BASICS- SQL QUERIES WITH PHP	
	INTRODUCTION OBJECTIVES HISTORY AND DEVELOPMENT PROCESS OF MySQL MYSQL BUILT-IN FUNCTIONS MYSQL QUERIES WORKING WITH SQL QUERIES ACCESSING DATA FROM MYSQL DATABASE USING PHP POINTS TO REMEMBER GLOSSARY CHECK YOUR PROGRESS BIBLIOGRAPHY/REFERENCES SUGGESTED READINGS	155-171

UNIT 1

JAVASCRIPT

1.1. INTRODUCTION

1.2. OBJECTIVES

1.3. ADD JAVASCRIPT IN YOUR HTML PAGE

1.4. PROGRAMMING VS. SCRIPTING

1.5. TOKENS IN JAVASCRIPT

1.6. BASIC PROGRAMMING CONSTRUCTS IN JAVASCRIPT

1.7. ARRAYS IN JAVASCRIPT

1.8. OBJECTS IN JAVASCRIPT

1.9. POINTS TO REMEMBER

1.10. GLOSSARY

1.11. CHECK YOUR PROGRESS

1.12. BIBLIOGRAPHY/REFERENCES

1.13. SUGGESTED READINGS

1.1. INTRODUCTION

JavaScript is the programming language that facilitates web pages to interact with client. It presents a page more interactive and dynamic. It was formed in 1995 for web-based programming and today it also runs on your phone applications.

This script was invented by Brendan Eich in 1995, and became an ECMA (European Computer Manufacturers Association) standard in 1997. Initially JavaScript was named as “LiveScript”, but Netscape Corporation changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript was introduced on Netscape 2.0 in 1995 with the name “LiveScript”.

Features of JavaScript

- JavaScript is a lightweight programming language.
- JavaScript is an Interpreted programming language.

- It is designed to create networking centred applications.
- It is complimentary to and integrated with Java.
- JavaScript is very easy to implement because it is integrated with HTML.
- It is cross-platform and open.

1.2. OBJECTIVE

After successful completion of this unit you will be able to-

- Define JavaScript
- Include JavaScript in your HTML page.
- Create objects and access their properties.
- Create your own interactive web page and Develop logical thinking.
- Take input from user and providing relevant response using JavaScript.

1.3. ADD JAVASCRIPT IN YOUR HTML PAGE

There is a flexibility given to include JavaScript code anywhere in an HTML Document. However the most preferred ways to include JavaScript in an HTML file are as follows:

- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body> and <head>...</head> sections.
- Script in an external file and then include in <head>...</head> section.

Script in <head>...</head> section-

Now you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows.

```
<html>
  <head>
    <script type= "text/JavaScript">
      function thismessage()
      {
        alert("Hello World");
      }
    </script>
  </head>
  <body>
    Click here for the result
    <input type= "button" onclick= "thismessage()" value= "This message" />
  </body>
</html>
```

JavaScript in <body>...</body> segment-

If you require a script to run as the page loads so that the script generates content in the page, then the script goes in the `<body>` portion of the document. In this case, you would not have any function defined using JavaScript. Take a look at the following code.

```
<html>
  <head>
  </head>
  <body>
    <script type= "text/JavaScript">
      document.write("Hello World");
    </script>
    <p>This is web page body </p>
  </body>
</html>
```

Script in `<body>...</body>` and `<head>...</head>` segment-

You can put your JavaScript code in `<head>` and `<body>` section altogether.

```
<html>
  <head>
    <script type= "text/JavaScript">
      function thismessage() {
        alert("Hello World!");
      }
    </script>
  </head>
  <body>
    <script type="text/JavaScript">
      document.write("Hello World!");
    </script>
    <input type= "button" onclick= "thismessage()" value= "This
Message" />
  </body>
</html>
```

External file script and then include in `<head>...</head>` section-

As you begin to work more comprehensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site. You are not restricted to be maintaining identical code in multiple HTML files.

The script tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

Here is an example to show how you can include an external JavaScript file in your HTML code using script tag and its *src* attribute.

```
<html>
  <head>
    <script type= "text/JavaScript" src= "filename.js" > </script>
  </head>
```

```
<body>
.....
</body>
</html>
```

To use JavaScript from an external file source, you need to write all your JavaScript source code in a simple text file with the extension “.js” and then include that file as shown above.

For example, you can keep the following content in *filename.js* file and then you can use `message()` function in your HTML file after including the *filename.js* file.

```
function message() {
    alert(“Hello World”);
}
```

1.4. PROGRAMMING VS. SCRIPTING

JavaScript and Java-

JavaScript and java are both Object Oriented Programming languages, but JavaScript is an object-oriented scripting language. They are completely different programming languages with a small number of similarities. JavaScript is mainly used in Web pages. Almost all Java expression syntax and naming conventions are followed by JavaScript. This is the main reason for calling it JavaScript. Unlike Java, JavaScript does not bother about the methods, variables and classes at all. Java is much more complex than JavaScript. Let us compare Java and JavaScript in detail.

- In Java, the source code is first converted into an intermediate code, known as the byte code. This byte code is non-readable by humans and is independent of any machine. Afterwards, this byte code is interpreted by the interpreter (Java Virtual Machine). Since JavaScript is a scripting language, it is only interpreted. So, making any modification in a Java program are more difficult than making the changes in a JavaScript program.
- Java needs the Java Development Kit. JavaScript mainly requires a text editor.
- Java is a strongly typed language instead JavaScript is a loosely typed language.
- In a Web environment, Java is a back-end language whereas JavaScript is a front-end language.
- JavaScript is considered as a part of HTML file, so it is visible in the source file. Java applets are not a part of HTML file so they are not visible in the source file.
- Java is very complex to learn due to its rules and restrictions whereas JavaScript is very easy to learn, as compared to Java.
- The User Interface of JavaScript is developed in HTML and is very easy to understand, whereas the User Interface of Java is developed in AWT or Swing, which is very complex to understand.

- The client-side is more secured in Java compared to JavaScript.
- In Java and JavaScript, routines are known as the methods and functions respectively.
- Java supports Polymorphism, but JavaScript does not support Polymorphism.
- Java uses the classes and objects to make its code reusable easily, but this service is not available in JavaScript.

JavaScript can be implemented using JavaScript statements that are placed within the `<script>... </script>` HTML tags in a web page. The `<script>` tags may be placed anywhere in your web page, but it is recommended to keep it within `<head>` tags. The `<script>` tag alerts the browser program to start interpreting all the text between these tags as a script.

The syntax of our JavaScript should be -

```
<script> JavaScript code </script>
```

The script tag contains two attributes:

- Type: This attribute value should be set to “text/JavaScript”.
- Language: This attribute specifies the scripting language we are using. Here it is JavaScript.

So now our JavaScript syntax is:

```
<script language= “JavaScript” type= “text/JavaScript”>
    JavaScript code;
</script>
```

Example:

Let us take a sample example to print “Hello World”.

we call a function *document.write* which writes a string into our HTML document.

```
<html>
<body>
<script language= “JavaScript” type= “text/JavaScript”>
    document.write (“Hello World!");
</script>
</body>
</html>
```

This code will produce the following result:

Hello World!

Programming Vs. Scripting-

Basically, all scripting languages are programming languages. The theoretical difference between the two is that scripting languages do not require the compilation step and are rather interpreted. For example, normally, a C program needs to be compiled before running whereas normally, a scripting language like JavaScript or PHP need not be compiled.

Generally, compiled programs run faster than interpreted programs because they are first converted native machine code. Also, compilers read and analyze the code only once, and report the errors collectively that the code might have, but the interpreter will read and analyze the code statements each time it meets them and halts at that very instance if there is some error.

Scripting languages are languages that are not compiled, more like interpreted at run-time. It's like a file containing instructions for a computer to follow to carry out a task. Examples of scripting languages are JavaScript, VB script, Perl, python, php, shell script, etc.

A scripting language generally sits behind some programming language. JavaScript is extremely fast to load since most of the web browser comes with already installed components.

1.5. *TOKENS IN JAVASCRIPT*

Tokens are the smallest individual words, phrases, or characters that JavaScript can understand. When JavaScript is interpreted, the browser parses the script into these tokens while ignoring comments and white space. JavaScript tokens fit in five categories:

- Identifiers
- Keywords
- Literals
- Operators
- Separators

Identifiers-

Identifiers are simply names that represent variables, methods, or objects. They consist of a combination of characters and digits. Some names are already built into the JavaScript language and are therefore reserved these identifiers are called as "Keywords". Aside from these keywords, you can define your own creative and meaningful identifiers. Of course, you have a couple of rules to follow. As-

- You must begin all identifiers with either a letter or underscore (_).
- You can then use letters, digits, or underscores for all subsequent characters.
- Letters include all uppercase characters, "A" through "Z", and all lowercase characters, "a" through "z".

- Digits include the characters “0” through “9”.

The table below shows some examples of valid and invalid identifiers. As-

Valid	Invalid	Reason
current- WebSite	current WebSite	contains a space
NumberOfHits	#ofIslands	pound sign is prefixed

Keywords-

Keywords are predefined identifiers that make up the core of a programming language. In JavaScript, they perform unique functions such as declaring new variables and functions, making decisions based on the present state of the computer, or starting a repetitive loop inside your application. Keywords, which are built into JavaScript, are always available for use by the programmer but must follow the correct syntax.

List of keywords		
break continue else false this while	for function if in true	int new null return var

Literals-

Literals are data comprised of numbers or strings used to represent fixed values in JavaScript. They are values that do not change during the execution of your scripts. The following five sections contain descriptions and examples of the different types of literals that you can use.

Integer Literals

Integers can be expressed in either decimal (base 10), octal (base 8), or hexadecimal (base 16) format. An integer literal in decimal format can include any sequence of digits that does not begin with a 0 (zero). A zero in front of an integer literal designates octal form. The integer itself can include a sequence of the digits 0 through 7. To designate hexadecimal, Ox (or OX) is used before the integer. Hexadecimal integers can include digits 0 through 9 along with the letters. a through f or A through F. Some examples include:

Decimal (base 10)	33,2139
Octal (base 8)	071,03664
Hexadecimal (base 16)	Ox7b8, OX395

Floating-Point Literals

Floating-point literals represent decimal numbers with fractional parts. They can be expressed in either standard or scientific notation. With scientific notation, use either

e or E to designate the exponent. Both the decimal number and exponent can be either signed or unsigned as shown in the examples:

3405.673

-1.958

8.3200e+ 11

8.3200e11

9.98E-12

Boolean Literals

JavaScript implements Boolean data types and therefore supports the two literals, true and false.

They represent the Boolean values 1 and 0, respectively. The true and false keywords must appear in lowercase. As a result, the capitalized words TRUE and FALSE are

Operator	Description	Example	Result in 'y'	Result in 'x'
----------	-------------	---------	---------------	---------------

left open to define as your own identifiers, but it is not recommended.

String Literals

A string literal is zero or additional characters enclosed in double (") or single (') quotes. JavaScript gives you this option, but you must use the same type of quote to surround each string. The following are examples of string literals enclosed in quotes:

"virtual communities"

'virtual communities'

"Look, up in the sky!"

Special Characters

When writing scripts, you might sometimes need to tell the computer to use a special character or keystroke such as a tab or carriage return. To do this, use a backslash in front of one of the special characters as shown in the following list:

\b	indicates a backspace
\f	indicates a form feed
\n	indicates a new-line character
\r	indicates a carriage return
\t	indicates a tab character

Operators-

The following tokens are used in JavaScript as operators:

+	Addition	$x = y + 2$	$y = 5$	$x = 7$
-	Subtraction	$x = y - 2$	$y = 5$	$x = 3$
*	Multiplication	$x = y * 2$	$y = 5$	$x = 10$
/	Division	$x = y / 2$	$y = 5$	$x = 2.5$
%	Modulus (division remainder)	$x = y \% 2$	$y = 5$	$x = 1$
++	Increment	$x = ++y$	$y = 6$	$x = 6$
		$x = y++$	$y = 6$	$x = 5$
--	Decrement	$x = --y$	$y = 4$	$x = 4$
		$x = y--$	$y = 4$	$x = 5$

Operator	Description	Example	Result in 'x'
=	$x = y$	$x = y$	$x = 5$
+=	$x += y$	$x = x + y$	$x = 15$
-=	$x -= y$	$x = x - y$	$x = 5$
*=	$x *= y$	$x = x * y$	$x = 50$
/=	$x /= y$	$x = x / y$	$x = 2$
%=	$x \% = y$	$x = x \% y$	$x = 0$

Operator	Description	Comparing If $x=5$	Returns
==	equal to	$x == 5$	true
===	equal value and equal type	$x === "5"$	false
		$x === 5$	true
!=	not equal	$x != 8$	true
!==	not equal value or not equal type	$x !== "5"$	true
		$x !== 5$	false
>	greater than	$x > 8$	false
<	less than	$x < 8$	true
>=	greater than or equal to	$x >= 8$	false
<=	less than or equal to	$x <= 8$	true

Separators-

The following characters are used in JavaScript as separators (punctuators):

Separator: one of :- () { } [],

1.6. BASIC PROGRAMMING CONSTRUCTS IN JAVASCRIPT

JavaScript is a programming language that adds interactivity to your website (for example: games, responses when buttons are pressed or data entered in forms, dynamic styling, and animation). For doing that you should know the basics of language, which are discussed below.

Displaying output in JavaScript

JavaScript can “display” data in different ways:

- (i) Writing into an HTML element, using innerHTML
- (ii) Writing into the HTML output using document.write()
- (iii) Writing into an alert box, using window.alert()

Writing into an HTML element, using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method. The id attribute defines the HTML element. The innerHTML property defines the HTML content:

```
<html>
  <body>
    <h1>My First ever Web Page</h1>
    <p>My First ever Paragraph</p>
    <p id= "demo1"></p>
    <script>
      document.getElementById("demo1").innerHTML = 7 + 6;
    </script>
  </body>
</html>
```

Writing into the HTML output using document.write()

Writing into the HTML output using document.write(). For testing purposes, it is convenient to use document.write():

```
<html>
  <body>
    <h1>My First ever Web Page</h1>
    <p>My first ever paragraph.</p>
    <script>
      document.write(7 + 6);
    </script>
  </body>
</html>
```

Using document.write() after an HTML document is completely loaded, will delete all existing HTML.

Writing into an alert box, using window.alert()

You can use an alert box to display data:

```
<html>
  <body>
    <h1>My First ever Web Page</h1>
    <p>My first ever paragraph.</p>
    <script>
      window.alert(7 + 6);
    </script>
  </body>
</html>
```

Example: JavaScript program to calculate multiplication and division of two numbers (input from user).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8 />
    <title>Program to calculate multiplication and division of two numbers
    </title>
    <style type= "text/css">
      body {margin: 30px;}
    </style>
    <script type = "text/JavaScript" language="JavaScript">
      function multiplyBy()
      {
        num1 = document.getElementById("fstNum").value;
        num2 = document.getElementById("scndNum").value;
        document.getElementById("answer").innerHTML =
num1 * num2;
      }
      function divideBy()
      {
        num1 = document.getElementById("fstNum").value;
        num2 = document.getElementById("scndNum").value;
        document.getElementById("answer").innerHTML =
num1 / num2;
      }
    </script>
  </head>
  <body>
    <form>
      1st Number : <input type= "text" id=" fstNum" /><br>
      2nd Number: <input type= "text" id= "scndNum" /><br>
      <input type= "button" onClick= "multiplyBy()" Value=
"Multiply" />
      <input type= "button" onClick= "divideBy()" Value= "Divide"
/>
    </form>
    <p> Answer is : <br>
      <span id = "answer"></span>
    </p>
  </body>
</html>
```

Some Basics about JavaScript Statement-

A statement is a line of code.

Example:

```
var hello = "Hello";
var world = "World";
```

```
var message = hello + " " + world;
```

This code can be executed by another program called an interpreter that will read the code, and execute all the statements in the right order.

Case sensitivity-

All JavaScript identifiers are **case sensitive**. The variables **lastName** and **lastname**, are two different variables.

```
var lastname, lastName;  
lastName = "Naman";  
lastname = "Malik";
```

JavaScript does not interpret VAR or Var as the keyword var.

JavaScript is a case-sensitive language. This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters. So the identifiers Time and TIME will convey different meanings in JavaScript.

Semicolon-

Semicolons separate JavaScript statements. Add a semicolon at the end of each executable statement:

```
var x, y, z;  
x = 5;  
y = 6;  
z = x + y;
```

Comments-

Comments are statements that will not be executed by the interpreter, comments are used to mark annotations for other programmers or small descriptions of what your code does, thus making it easier for others to understand what your code does.

In JavaScript, comments can be written in 2 different ways:

A. Line starting with //:

```
// This is a comment, it will be ignored by the interpreter
```

B. Section of code starting with /*and ending with */, this method is used for multi-line comments:

```
/*  
This is a multi-line comment,  
it will be ignored by the interpreter  
*/
```

Variables-

Variables are containers for changeable values. Variables can hold all kind of values and also the results of computations. Variables have a name and a value separated by an equal sign (=). Use the var keyword only for declaration or initialization, first time you use the variable in a document. You should not re-declare same variable twice.

JavaScript is **untyped** language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

Example:

```
var value;
```

```
var result = 2;
```

You can also declare multiple variables with the same var keyword as follows –

```
var value, result;
```

Example:

This example defines two variables, after computing their addition the result is stored in third variable.

```
var x = 5;
```

```
var y = 6;
```

```
var result = x + y;
```

Types of variable-

Computers are sophisticated and can make use of more complex variables than just numbers. This is where variable types come in. Variables come in several types and different languages support different types.

The most common types are:

a. **Numbers:**

i. **Float:** a number, like 1.21323, 4, -33.5, 100004 or 0.123

ii. **Integer:** a number like 1, 12, -33, 140 but not 1.233

b. **String:** a line of text like “boat”, “elephant” or “damn, you are tall!” (We will discuss more on strings in later section).

c. **Boolean:** either true or false, but nothing else

d. **Arrays:** a collection of values like: 1, 2, 3, 4, ‘I am bored now’

e. **Objects:** a representation of a more complex object

f. **Null:** a variable that contains null contains no valid Number, String, Boolean, Array, or Object.

g. **undefined:** the undefined value is obtained when you use an object property that does not exist, or a variable that has been declared, but has no value assigned to it.

Strings-

This is a string.

```
“I love programming”;
```

You can define strings using var keyword either using single or double quotes.

```
// 'Single quotes'
var str = 'Please Enter your name';

// "Double quotes"
var print = "Thank you for your input";
```

Concatenation of strings-

Concatenation is adding two or more strings together to creating a larger string. This is done using the + operator.

```
var message = 'wow' + 'JS is easy to learn' + 'and ' + 'easy to use';
var firstName = "Alok";
var lastName = "Kumar";
var fullName = firstName + " " + lastName;
```

You can make sure the length of your string in JavaScript (length is the number of characters in string) just use the property *.length*

```
var sizeofString = 'I Love India'.length;
```

Keywords-

There are some reserved words in every language which cannot be used as variables. JavaScript keywords are:

Abstract	arguments	boolean	Break
Byte	Case	catch	Char
Const	Continue	debugger	Default
Delete	Do	double	Else
Eval	False	Final	Finally
Float	For	function	Goto
If	implements	In	instanceof
Int	Interface	Long	Native
New	Null	package	Private
Protected	Public	return	short
Static	Switch	synchronized	this
Throw	Throws	transient	true
Try	Typeof	Var	void
Volatile	While	With	yield

Conditional Logic-

- (i) If Statements
- (ii) If...else Statements

(iii) Conditional operator

If Statements-

The 'if' statement is the basic control statement that permit JavaScript to make decisions and execute statements conditionally

Syntax

The syntax for a basic if statement is as follows:

```
if (expression){  
    Statement(s) to be executed if expression is true  
}
```

Here a JavaScript expression is evaluated. If the resulting value is true, the given statement(s) are executed. If the expression is false, then no statement would be not executed. Most of the times, you will use comparison operators while making decisions.

Example:

```
<html>  
    <body>  
        <script type= "text/JavaScript">  
            var age = 20;  
            if( age > 18 ){  
                document.write("<b>Qualifies for driving</b>");  
            }  
        </script>  
    </body>  
</html>
```

If...else Statements-

The 'if...else' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

Syntax

The syntax of an if-else statement is as follows:

```
if (/* expression */) {  
    Statement executed if expression is true  
} else {  
    Statement executed if expression is false  
}
```

Here JavaScript expression is evaluated. If the resulting value is true, the given statement(s) in the 'if' block, are executed. If the expression is false, then the

given statement(s) in the else block are executed.

Example:

```
<html>
  <body>
    <script type= "text/JavaScript">
      var age = 14;
      if( age > 18 ){
        document.write("<b>Qualifies for driving</b>");
      }else{
        document.write("<b>Does not qualify for driving</b>");
      }
    </script>
  </body>
</html>
```

Conditional operator-

In order to avoid the if-else hassle, simple logical comparisons can be utilised.

```
var adult = (age > 18) ? "YES" : "NO";
```

In the above example, ? is a logical operator. The code says that if the value of age is greater than 18, then adult = YES; otherwise adult = NO.

Basically, if the condition is true, the first argument is accessed and if the comparison/condition is false, the second argument is accessed.

Looping-

Loops are used for statements that need to be repeated. One or more variable in the loop changes so that you can check condition to stop. You can run the same code again and again, each time with a different value. There are three types of loops.

- (i) For loop
- (ii) For ... in loop
- (iii) While loop
- (iv) Do while loop

For loop-

The easiest form of a loop is *for statement*. This one has a syntax that is similar to *if statement*, but with more options:

Syntax-

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

Statement 1- is executed before the loop starts.

Statement 2- defines the end condition when the loop had to stop.

Statement 3- is executed each time after the loop has been executed.

Example: for loop to print numbers from 1 to 5

```
<html>
  <body>
    <p id="output"></p>
    <script>
      var text = "";
      var i;
      for (i = 0; i < 5; i++) {
        text += "The number is " + i + "<br>";
      }
      document.getElementById("output").innerHTML = text;
    </script>
  </body>
</html>
```

For ... in loop

A *for...in* loop is used to loop through an object's properties. As we have not discussed Objects yet, you may not feel comfortable with this loop. But once you understand how objects behave in JavaScript, you will find this loop very useful.

Syntax

```
for (variablename in object)
{
  statement or block to execute
}
```

While loop-

While Loops repetitively execute a block of code as long as a specified condition is true.

```
while(condition){
  // do it as long as condition is true
}
```

Example: the loop in this example will repetitively execute its block of code as long as the variable *i* is less than 5:

```
var i = 0, x = "";
while (i < 5) {
  x = x + "The number is " + i;
  i++;
}
```

Do while loop-

The Do/While Loop is a variant of the while loop. This loop will execute the code block once before checking if the condition is true. It then repeats the loop as long as the condition is true:

```
do {  
    // code block to be executed  
} while (condition);
```

Note: Be careful to avoid infinite looping if the condition is always true!

Example: Print numbers less than 10 using do...while loop:

```
var i = 0;  
do {  
    document.write(i + " ");  
    i++; // incrementing i by 1  
} while (i < 10);
```

Note: $i = i + 1$ can be written $i++$.

Example: using this code you can access array with loop.

```
var myArray = ["aa", "bb"];  
// for-in loop  
for (var i in myArray) {  
    document.write(myArray[i] + ", "); // output is : aa, bb,  
}
```

Functions

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions. Like any other advanced programming language, JavaScript also supports all the features necessary to write modular code using functions. You must have seen functions like `alert()` and `write()` in the earlier chapters. We were using these functions again and again, but they had been written in core JavaScript only once.

JavaScript allows us to write our own functions as well. This section explains how to write your own functions in JavaScript.

- (i) Function definition
- (ii) Function calling
- (iii) Function parameters/arguments
- (iv) Return statement

Function definition-

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

Syntax:

```
<script type= "text/JavaScript">
function functionname(parameter-list)
{
    statements
}
</script>
```

Example:

Try the following example. It defines a function called sayHello that takes no parameters:

```
<script type= "text/JavaScript">
function sayHello()
{
    alert("Hello there");
}
</script>
```

Function calling-

Calling a Function To invoke a function somewhere later in the script, you would simply need to write the name of that function as shown in the following code.

```
<html>
    <head>
        <script type= "text/JavaScript">
            function sayHello()
            {
                document.write ("Hello there!");
            }
        </script>
    </head>
    <body>
        <p>Click the following button to call the function</p>
        <form>
            <input type= "button" onclick= "sayHello()" value= "Say
Hello">
        </form>
    </body>
</html>
```

Function Parameter/Arguments-

Till now, we have seen functions without parameters. But there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by comma. Example Try the following example. We have modified our “sayHello” function here. Now it takes two parameters.

```
<html>
  <head>
    <script type= “text/JavaScript”>
      function message(name, age)
      {
        document.write (name + “ is “ + age + “ years old.”);
      }
    </script>
  </head>
  <body>
    <p>Click the following button to call the function</p>
    <form>
      <input type= “button” onclick= “message(Neha, 7)” value=
“some message”>
    </form>
  </body>
</html>
```

Return statement-

A JavaScript function can have an optional return statement. This is required if you want to return a value from a function. This statement should be the last statement in a function. For example, you can pass two numbers in a function and then you can expect the function to return their multiplication in your calling program.

Example: following function that takes two parameters and concatenates them before returning the resultant in the calling program.

```
<html>
  <head>
    <script type= “text/JavaScript”>
      function concatenate(first, last)
      {
        var full;
        full = first + last;
        return full;
      }
      function secondFunction()
      {
```



```

        var result;
        result = concatenate(Neha, Pant);
        document.write (result);
    }
</script>
</head>
<body>
    <p>Click the following button to call the function</p>
    <form>
        <input type= "button" onclick= "secondFunction()" value=
        "Call Function">
    </form>
</body>
</html>

```

1.7. ARRAYS IN JAVASCRIPT

The Array object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Syntax-

Use the following syntax to create an Array Object. As-

```
var fruits = new Array( "apple", "orange", "mango" );
```

The Array parameter is a list of strings or integers. When you specify a single numeric parameter with the Array constructor, you specify the initial length of the array. The maximum length allowed for an array is 4,294,967,295.

You can create array by simply assigning values as follows:

```
var fruits = ["apple", "orange", "mango" ];
```

You will use ordinal numbers to access and to set values inside an array as follows.

fruits[0] is the 1st element

fruits[1] is the 2nd element

fruits[2] is the 3rd element

Properties of array-

- (i) Length
- (ii) Prototype
- (iii) Methods (readymade functions) of array

Length-

Array length property returns an unsigned, 32-bit integer that specifies the number of elements in an array.

Syntax-

array.length

Return Value

Returns the length of an array.

Example:

```
<html>
<head>
    <title>JavaScript Array length Property</title>
</head>
<body>
<script type= "text/JavaScript">
    var arr = new Array( 10, 20, 30 );
    document.write("arr.length is:" + arr.length);
</script>
</body>
</html>
```

Output-

arr.length is: 3

Prototype-

The property of prototype permits you to add properties and methods to any object viz. (Number, Boolean, String, Date etc).

Syntax-

object.prototype.name = value

Example:

```
<html>
    <head>
        <title>User-defined objects</title>
    <script type= "text/JavaScript">
        function book(title, author)
        {
            this.title = title;
            this.author = author;
        }
    </script>
```

```

</head>
<body>
    <script type= "text/JavaScript">
        var myBook = new book("Programming in C", "Kanitkar");
        book.prototype.price = null;
        myBook.price = 100;
        document.write("Book title is : " + myBook.title + "<br>");
        document.write("Book author is : "+ myBook.author + "<br>");
        document.write("Book price is : " + myBook.price + "<br>");
    </script>
</body>
</html>

```

Output:

Book title is : Programming in C

Book author is : Kanitkar

Book price is : 100

Methods (readymade functions) of array-

Method	Description
concat()	Returns a new array comprised of this array joined with other array(s) and/or value(s).
every()	Returns true if every element in this array satisfies the provided testing function.
filter()	Creates a new array with all of the elements of this array for which the provided filtering function returns true.
forEach()	Calls a function for each element in the array.
indexOf()	Returns the first (least) index of an element within the array equal to the specified value, or 1 if none is found.
join()	Joins all elements of an array into a string.
lastIndexOf()	Returns the last (greatest) index of an element within the array equal to the specified value, or 1 if none is found.
map()	Creates a new array with the results of calling a provided function on every element in this array.
pop()	Removes the last element from an array and returns that element
push()	Adds one or more elements to the end of an array and returns the new length of the array.
reduce()	Apply a function simultaneously against two values of array (from left-to-right) as to reduce it to a single value.
reduceRight()	Apply a function simultaneously against two values of array (from right-to-left) as to reduce it to a single value.
reverse()	Reverses the order of the elements of an array the first becomes the last, and the last becomes the first.

shift()	Removes the first element from an array and returns that element.
slice()	Extracts a section of an array and returns a new array.
some()	Returns true if at least one element in this array satisfies the provided testing function.
toSource()	Represents the source code of an object
sort()	Sorts the elements of an array.
splice()	Adds and/or removes elements from an array.
toString()	Returns a string representing the array and its elements.
unshift()	Adds one or more elements to the front of an array and returns the new length of the array.

Example:

Here is an example of array method concat().

JavaScript array concat() method returns a new array comprised of this array joined with two or more arrays.

Syntax-

array.concat(value1, value2, ..., valueN);

Parameter Details

valueN : Arrays and/or values to concatenate to the resulting array.

Return Value

Concatenated values of different arrays in the resulting array

Example:

```
<html>
  <head>
    <title>JavaScript Array concat Method</title>
  </head>
  <body>
    <script type="text/JavaScript">
      var alpha = ["a", "b", "c"];
      var numeric = [1, 2, 3];
      var alphaNumeric = alpha.concat(numeric);
      document.write("alphaNumeric : " + alphaNumeric );
    </script>
  </body>
</html>
```

Output:

alphaNumeric: a,b,c,1,2,3

Example:

array pop() method which returns top(last in) element.

```
<html>
  <body>
    <h2>JavaScript Array Methods</h2>
```

```

<h2>pop()</h2>
<p>The pop() method removes the last element from an array.</p>
<p id= "prev"></p>
<p id= "last"></p>
<script>
    var fruits = ["banana", "apple", "orange", "mango"];
    document.getElementById("prev").innerHTML = fruits;
    fruits.pop();
    document.getElementById("last").innerHTML = fruits;
</script>
</body>
</html>

```

1.8. OBJECTS IN JAVASCRIPT

It is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers.

- (i) Encapsulation
- (ii) Aggregation
- (iii) Inheritance
- (iv) Polymorphism

Encapsulation

The capability to store related information, whether data or methods, together in an object.

Aggregation

The capability to store one object inside another object.

Inheritance

The capability of a class to rely upon another class (or number of classes) for some of its properties and methods.

Polymorphism

The capability to write one function or method that works in a variety of different ways. Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object; otherwise the attribute is considered a property.

A good Example from our real life to understand an object. As- a car

A car has “properties” like weight and color, and “methods (functions)” like start and stop:

Properties- car.name=Alto, car.model = K10, car.color = white, car.type = hatchback

Methods- car.start(), car.accelerate(), car.brake(), car.stop()

Creation of objects

There are two ways to create an object in JavaScript: as-

(i) Literal

```
var object = {};
```

(ii) object-oriented

```
var object = new Object();
```

Object Properties

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used throughout the page.

The syntax for adding a property to an object is:

```
objectName.objectProperty = propertyValue;
```

Example: The following code gets the document title using the "title" property of the document object.

```
var str = document.title;
```

Object Methods-

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

Example: Following is a simple example to show how to use the write() method of document object to write any content on the document.

```
document.write ("This is test");
```

Note: We will discuss objects in detail in next unit

1.9. POINTS TO REMEMBER

- JavaScript is a client-side as well as server side scripting language.
- JavaScript is an Interpreted programming language.
- JavaScript is also an Object based Programming language.
- JavaScript code is place inside <script></script> tags.

- JavaScript is a “*loosely typed*” language, which means that you don't have to explicitly declare what type of data the variables are. You just need to use the var keyword to indicate that you are declaring a variable, and the interpreter will work out with its associated properties.
- JavaScript is case sensitive.
- You can define strings in JavaScript by enclosing the text in single quotes or double quotes. Ex: var name='Anika'; or var name="Anika";
- For ... in statement is used to access object properties.
- You can find number of elements in array using array.length property.
- Functions can be defined using function keyword: function fn_name{}
- Object is an entity having some methods and some properties

1.10. GLOSSARY

- **ECMA Script:** It is a trademarked scripting-language specification standardized by ECMA International in ECMA-262 and ISO/IEC 16262. It was created to standardize JavaScript.
 - **Compilation:** the translation of source code into object code by a compiler
 - **Interpretation:** a process to translate source code into some efficient intermediate representation which can be immediately executed.
-

1.11. CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type questions)

- We can change an html element attribute using method.
- Variables are declared using ... keyword.
- Javascript variables can hold values of
- We can check length of string var message as:
- Syntax of logical comparison operator is:
- pop() method of array removes the element of array.
- The code inside loop block for(i=1;i<5;i++) will be executed times.
- alert() is an inbuilt of javascript.
- An object can be defined using Keyword.
- Object.color is a (property/method).
- document.write() is a (property/method)
- Identifiers are combination of and
- Literals contain values.
- Break, continue, new are some JavaScript
- Boolean literals contain value 0 and 1 (True/False).

- p) Interpretation of code is done at run time. (True/False)
- q) Lowercase and uppercase variables are treated as same variables in javascript. (True/False)
- r) By using Array.length method we can also find number of elements in array. (True/False)
- s) The condition statement in a do while loop is written at the end of block. (True/False)
- t) Object is a collection of property and methods. (True/ False)

Part- B (Descriptive type questions)

- a) Inside which HTML element do we put the JavaScript?
- b) What is the correct syntax for referring to an external script called “myscript.js”?
- c) How can one create an Object in JavaScript?
- d) How can you read properties of an Object in JavaScript?
- e) How can you build an Array in JavaScript?
- f) How to read elements of an array in JavaScript?
- g) Which built-in method returns the length of the string?
- h) Which built-in method removes the last element from an array and returns that element?
- i) What is a method in JavaScript?
- j) Define function in JavaScript.

Answers- (Part- A- Fill in the blanks and True/False type)		
(a) innerHTML	(b) var	(c) any type
(d) message.length	(e) var x = (condition)? True : false	(f) Last
(g) 4 times	(h) Function	(i) New
(j) property	(k) method	(l) Character and digits
(m) Fixed	(n) Keywords	(o) True
(p) True	(q) True	(r) True
(s) True	(t) True	

1.12. BIBLIOGRAPHY/REFERENCES

- [1] www.w3schools.com/js/default.asp, January 2018
- [2] www.tutorialspoint.com/javascript/index.htm, January 2018
- [3] en.wikipedia.org/wiki/JavaScript, January 2018
- [4] www.html-5-tutorial.com, January 2018

1.13. SUGGESTED READINGS

- [1] Daggett Mark E. (2013) Expert Javascript, Apress, XIV, 248

UNIT 2

JAVASCRIPT EVENTS

2.1. INTRODUCTION

2.2. OBJECTIVE

2.3. JAVASCRIPT EVENTS

2.4. JAVASCRIPT OBJECTS AND PROPERTIES

2.5. CLIENT-SIDE VALIDATIONS

2.6. COOKIES

2.7. DHTML

2.8. POINTS TO REMEMBER

2.9. GLOSSARY

2.10. CHECK YOUR PROGRESS

2.11. BIBLIOGRAPHY/REFERENCES

2.12. SUGGESTED READINGS

2.1 INTRODUCTION

In previous unit we have learnt about basic programming constructs of JavaScript. Now we will discuss how to handle user driven events and reply accordingly. We will also discuss objects in detail.

In this unit you will learn how DHTML can be used to develop an interactive web page and cookies can be used to store user preferences.

When users trip your web, they do things like click on text and images and given links, hover over things etc. These are examples of what JavaScript calls events.

When page loads called an event. If the user clicks on a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

2.2 OBJECTIVE

After successful completion of this unit you will be able to-

- Create JavaScript events in a webpage.
- Create and access objects with relevant properties and methods.
- Validate input by user in a web page.
- Use cookies in a web page /project.
- Design interactive web pages using DHTML.
- Develop logical thinking while using logical constructs of JavaScript.

2.3 JAVASCRIPT EVENTS

Interaction of JavaScript with HTML is handled through events that occur when the user or the browser manipulates a page.

- (i) onclick event
- (ii) onsubmit event
- (iii) onmouseover & onmouseout events
- (iv) HTML5 standard events

onclick event-

“onclick” event is frequently used event. This event occurs when a user clicks the left button of his/her mouse. You can use/activate your validation or warning or etc., against this event type.

Example

```
<html>
  <head>
    <script type= "text/JavaScript">
      <!-- function sayWelcome () {
        alert("Welcome")
      }
      //--> //comments placed to prevent execution of js if browser does not support it.
    </script>
  </head>
  <body>
    <p>Click on the button and get the result</p>
    <form>
      <input type="button" onclick= "sayWelcome()" value="Say
Welcome" />
    </form>
  </body>
</html>
```

onsubmit event-

“onsubmit” event is generated when someone submits a form. You can create form validation action against “onsubmit” event.

Example:

The example given below show how to use “onsubmit” event. Here we are calling a validate() function. The validate() function called before submitting the form data to the webserver. If this function returns true, then the form data will be submitted to the webserver, otherwise it will not be submitted to the web server.

Try the following example.

```
<html>
  <head>
    <script type= "text/JavaScript">
      <!--
        function validate() {
          all validation goes here
          .....
          return either true or false
        }
      //-->
    </script>
  </head>
  <body>
    <form method= "POST" action= "submit.html" onsubmit= "return
validate()">
      .....
      <input type= "submit" value= "Submit" />
    </form>
  </body>
</html>
```

onmouseover & onmouseout events-

The “onmouseover” and “onmouseout” events will help you to create effects with images as well as with text too. The “onmouseover” event triggers (occurs) when you move or bring mouse pointer over any element.

The “theonmouseout” event triggers (occurs) when you move or bring your mouse pointer out from that element. Example as given below-

```
<html>
<head>
<script type= "text/JavaScript">
  <!--
    function over() {
      document.write ("Mouse Over");
    }
    function out() {
      document.write ("Mouse Out");
    }
  -->
</script>
</head>
</html>
```

```

    //-->
  </script>
</head>
<body>
  <p>Bring your mouse inside the division to see the result:</p>
  <div onmouseover= "over()" onmouseout= "out()">
    <h2> This is inside the division </h2>
  </div>
</body>
</html>

```

HTML5 standard events-

The standard HTML 5 events are listed here for your reference. Here script indicates a JavaScript function to be executed against that event.

Attribute	Value	Description
Offline	script	Triggers when the document goes offline
Onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeonload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element changes
onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element is being dragged over a valid drop target

ondragover	script	Triggers at the start of a drag operation
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped
ondurationchange	script	Triggers when the length of the media is changed
onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reach the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes
onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded
onloadstart	script	Triggers when the browser starts to load the media data
onmessage	script	Triggers when the message is triggered
onmousedown	script	Triggers when a mouse button is pressed
onmousemove	script	Triggers when the mouse pointer moves
onmouseout	script	Triggers when the mouse pointer moves out of an element
onmouseover	script	Triggers when the mouse pointer moves over an element
onmouseup	script	Triggers when a mouse button is released

onmousewheel	script	Triggers when the mouse wheel is being rotated
onoffline	script	Triggers when the document goes offline
ononline	script	Triggers when the document comes online
ononline	script	Triggers when the document comes online
onpagehide	script	Triggers when the window is hidden
onpageshow	script	Triggers when the window becomes visible
onpause	script	Triggers when media data is paused
onplay	script	Triggers when media data is going to start playing
onplaying	script	Triggers when media data has start playing
onpopstate	script	Triggers when the window's history changes
onprogress	script	Triggers when the browser is fetching the media data
onratechange	script	Triggers when the media data's playing rate has changed
onreadystatechange	script	Triggers when the ready-state changes
onredo	script	Triggers when the document performs a redo
onresize	script	Triggers when the window is resized
onscroll	script	Triggers when an element's scrollbar is being scrolled
onseeked	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended
onseeking	script	Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect	script	Triggers when an element is selected
onstalled	script	Triggers when there is an error in fetching media data
onstorage	script	Triggers when a document loads
onsubmit	script	Triggers when a form is submitted
onsuspend	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate	script	Triggers when media changes its playing position

objectsonundo	script	Triggers when a document performs an undo
onunload	script	Triggers when the user leaves the document
Onvolumechange	script	Triggers when media changes the volume, also when volume is set to “mute”
Onwaiting	script	Triggers when media has stopped playing, but is expected to resume

2.4 JAVASCRIPT OBJECTS AND PROPERTIES

JavaScript is designed on a simple object-based paradigm. An object is a collection of properties, and a property is an association between a name (or *key*) and a value.

Objects are variable that contain several values. JavaScript objects are containers for named values, called properties and methods.

In JavaScript, almost “everything” is an object.

- Booleans can be objects (if defined with the new keyword)
- Numbers can be objects (if defined with the new keyword)
- Strings can be objects (if defined with the new keyword)
- Dates are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects
- All JavaScript values, except primitives, are objects.

An empty object can be created using one of two syntaxes:

- let myuser = new Object(); // “object constructor” syntax, almost like Java
- let myuser = {}; // “object literal” syntax

Object properties-

We can immediately put some properties into {...} as “key: value” pairs:

```
let user = { // an object
  name: "Meha", // key "name" contains value "Meha"
  age: 23 // key "age" contains value 23
};
```

A property has a key (also known as “name” or “identifier”) before the colon “:” and a value to the right of it. In the user object, there are two properties:

- The first property has the name “name” and the value “Meha”.
- The second one has the name “age” and the value 23.

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <p>building a JavaScript Object.</p>
    <p id="show"></p>
    <script>
      var employee = {
        firstName: "Meha",
        age: 23,
        designation: "Programmer",
        address: "Delhi"
      };
      document.getElementById("show").innerHTML =
employee.firstName + " is a " + employee.designation + " and " + employee.age + "
years old.";
    </script>
  </body>
</html>
```

Output: Meha is a Programmer and 23 years old.

Example:

```
<!DOCTYPE html>
<html>
<body>
  <p>
There are two different ways to access an object property:
  </p>
  <p>You can use person.property or person["property"].</p>
  <p id="show"></p>

  <script>
var employee = {
  firstName: "Meha",
  age: 23,
  designation: "Programmer",
  address: "Delhi"
};
document.getElementById("show").innerHTML = employee.firstName + " is a " +
employee.designation + " and " + employee.age + " years old." + "\n";
document.getElementById("show").innerHTML = employee["firstName"] + " is a "
+ employee["designation"] + " and " + employee["age"] + " years old." + "\n";
  </script>
</body>
```

```
</html>
```

Output:

Meha is a Programmer and 23 years old.

Meha is a Programmer and 23 years old

Object methods-

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the `this` keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

Example:

Following is a simple example to show how to use the `write()` method of document object to write any content on the document.

```
document.write ("This is text is displayed using JavaScript document.write method");
```

User-defined Objects-

In spite of in built objects you can build your own personalized objects in JavaScript. All user-defined objects and built-in objects are descendants of an object called `Object`.

The *new* operator is used to create an instance of an object. To create an object, the *new* operator is followed by the constructor method.

In the following example, the constructor methods are `Object()`, `Array()`, and `Date()`. These constructors are built-in JavaScript functions.

- a. `var employee = new Object();`
- c. `var cars = new Array("Maruti", "Tata", "Hyundai");`
- d. `var day = new Date("November 17, 2010");`

The `object()` constructor-

A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function called `Object()` to build the object. The return value of the `Object()` constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the `var` keyword.

Example 1: How to build an `Object`.

```
<html>
```

```
<head>
```

```

<title>Building User-defined objects</title>
<script type= "text/JavaScript">
    var book = new Object(); // Create the object
    book.title = "JavaScript"; // Assign properties to the object
    book.publisher = "Pearson";
</script>
</head>
<body>

    <script type= "text/JavaScript">
        document.write("Book Title is : " + book.title + "<br>");
        document.write("Book Publisher is : " + book.publisher + "<br>");
    </script>
</body>
</html>

```

Output:

Book Title is : JavaScript
 Book Publisher is : Pearson

Example 2:

Build an object using function and this keyword

This example demonstrates how to create an object with a User-Defined Function. Here this keyword is used to refer to the object that has been passed to a function.

```

<html>
<head>
    <title>User-defined objects using function and this keyword</title>
    <script type= "text/JavaScript">
        function book(title, publisher){
            this.title = title;
            this.publisher = author;
        }
    </script>
</head>
<body>
    <script type= "text/JavaScript">
        var myBook = new book("JavaScript", "Pearson");
        document.write("Book Title is : " + myBook.title + "<br>");
        document.write("Book Publisher is : " + myBook.publisher + "<br>");
    </script>

```

```

    </body>
</html>

```

Output:

Book Title is : JavaScript
Book Publisher is : Pearson

Example 3:

Accessing objects using for ... in loop

```

<html>

<body>

<h2>JavaScript object</h2>

<p>The for .. in statement to access properties of an object.</p>

<p id= "output"></p>

<script>

    var txt = "";

    var book = {title: "JavaScript", publisher: "Pearson", price: 350};

    var info;

    for (info in book) {

        txt = txt + book[info] + " ";

    }

    document.getElementById("output").innerHTML = txt;

</script>

</body>
</html>

```

Output:

JavaScript object
The for .. in statement to access properties of an object.
JavaScript Pearson 350

2.5 CLIENT-SIDE VALIDATIONS

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the Submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server. JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

- Basic Validation - First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.
- Data Format Validation - Secondly, the data that is entered, must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Example:

We will take an example to understand the process of validation. Here is a simple form in html format.

```
<html>
<head>
    <title>Form Validation</title>
    <script type= "text/JavaScript">
        // Form validation code will come here.
    </script>
</head>
<body>
    <form action= "/php/test.php" name= "myForm" onsubmit= "return validate();">
    <table cellpadding= "2" cellspacing= "2" border= "1">
    <tr>
    <td align="right">Name</td>
    <td><input type="text" name="txtName" /></td>
    </tr>
    <tr>
    <td align="right">EMail</td>
    <td><input type="text" name="txtEmail" /></td>
    </tr>
    <tr>
    <td align="right">Zip Code</td>
    <td><input type="text" name="Zip" /></td>
    </tr>
    <tr>
    <td align="right">Country</td>
    <td>
        <select name="cmbCountry">
            <option value="-1" selected>[choose yours]</option>
            <option value="1">USA</option>
            <option value="2">UK</option>
            <option value="3">INDIA</option>
        </select>
    </td>
    </tr>
    </table>
    </form>
</body>
</html>
```

```

<tr>
<td align="right"></td>
<td><input type="submit" value="Submit" /></td>
</tr>
</table>
</form>
</body>
</html>

```

Basic Form Validation-

First let us see how to do a basic form validation. In the above form, we are calling validate() to validate data when onsubmit event is occurring. The following code shows the implementation of this validate() function.

Example:

```

<script type="text/JavaScript">
    function validate()
    {
        if( document.myForm.Name.value == "" )
        {
            alert( "your name please!" );
            document.myForm.Name.focus() ;
            return false;
        }
        if( document.myForm.EMail.value == "" )
        {
            alert( "Email please!" );
            document.myForm.EMail.focus();
            return false;
        }
        if( document.myForm.Zip.value == "" || isNaN(
document.myForm.Zip.value ) || document.myForm.Zip.value.length != 5)
        {
            alert( "Please provide a zip in the format #####." );
            document.myForm.Zip.focus() ;
            return false;
        }
        if( document.myForm.Country.value == "-1" )
        {
            alert( "Please provide your country!" );
            return false;
        }
        return( true );
    }
</script>

```

Data Format Validation-

So we will see how you can validate your entered form data before submitting it to the web server. The following example shows how to validate an entered email address. An email address must contain at least a '@' sign and a dot (.). Also, the '@' must not be the first character of the email address, and the last dot must at least be one character after the '@' sign.

Example:

```
<script type="text/JavaScript">
    function validateEmail()
    {
        var emailID = document.myForm.EMail.value;
        atpos = emailID.indexOf("@");
        dotpos = emailID.lastIndexOf(".");
        if (atpos < 1 || ( dotpos - atpos < 2 ))
        {
            alert("Please enter correct email ID")
            document.myForm.EMail.focus() ;
            return false;
        }
        return( true );
    }
</script>
```

2.6 COOKIES

Servers and web-browsers use HTTP protocol to communicate and HTTP is a stateless protocol. But for a commercial website, it is required to maintain session information among different pages. For example, one user registration ends after completing many pages. But how to maintain users' session information across all the web pages!

In many situations, using cookies is the most efficient method of remembering and tracking preferences, purchases, commissions, and other information required for better visitor experience or site statistics.

How cookies work?

Your server sends some data to the visitor's browser in the form of a cookie. The browser may accept the cookie. If it does, it is stored as a plain text record on the visitor's hard drive. Now, when the visitor arrives at another page on your site, the browser sends the same cookie to the server for retrieval. Once retrieved, your server knows/remembers what was stored earlier.

Cookies are a plain text data record of 5 variable-length fields:

- Expires: The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- Domain: The domain name of your site.
- Path: The path to the directory or web page that set the cookie. This may be blank if you want to retrieve the cookie from any directory or page.
- Secure: If this field contains the word “secure”, then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists.
- Name=Value: Cookies are set and retrieved in the form of key-value pairs.

Cookies were originally designed for CGI programming. The data contained in a cookie is automatically transmitted between the web browser and the web server, so CGI scripts on the server can read and write cookie values that are stored on the client.

JavaScript can also manipulate cookies using the cookie property of the Document object. JavaScript can read, create, modify, and delete the cookies that apply to the current web page.

Storing cookies-

The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this.

document.cookie = “key1=value1; key2=value2; expires=date”;

Here the attribute named *expires* is optional. If you provide this attribute with a valid date or time, then the cookie will expire on a given date or time and thereafter, the cookies’ value will not be accessible.

Note: Cookie values may not include semicolons, commas, or whitespace. For this reason, you may want to use the JavaScript escape() function to encode the value before storing it in the cookie. If you do this, you will also have to use the corresponding unescape() function when you read the cookie value.

Example:

It sets a customer name in an input cookie.

```
<html>
  <head>
    <script type="text/JavaScript">
      <!--
        function WriteCookie()
        {
          if( document.myform.customer.value == "" )
          {
            alert ("Enter some value!");
            return;
          }
        }
      </script>
    </head>
  </html>
```



```

        cookievalue=
escape(document.myform.customer.value) + “,”;
        document.cookie=“name=“ + cookievalue;
        document.write (“Setting Cookies : “ + “name=“ +
cookievalue
        );
    }
    //-->
</script>
</head>
<body>
    <form name=“myform” action=“”>
        Enter name: <input type=“text” name=“customer”/>
        <input        type=“button”        value=“Set        Cookie”
onclick=“WriteCookie();”/>
    </form>
</body>
</html>

```

Output:

Enter name:

Now your machine has a cookie called name. You can set multiple cookies using multiple key=value pairs separated by comma.

Reading cookies-

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie. The document.cookie string will keep a list of name=value pairs separated by semicolons, where name is the name of a cookie and value is its string value.

You can use strings' split() function to break a string into key and values as follows:

Example:

Following example to gets all the cookies.

```

<html>
    <head>
        <script type=“text/JavaScript”>
            <!--
                function ReadCookie()
                {
                    var allcookies = document.cookie;
                    document.write (“All Cookies : ” + allcookies );
                    // Get all the cookies pairs in an array
                    cookiearray = allcookies.split(';');
                    // Now take key value pair out of this array
                    for(var i=0; i<cookiearray.length; i++){

```

```

        name = cookiearray[i].split('=')[0];
        value = cookiearray[i].split('=')[1];
        document.write ("Key is : " + name + " and
Value is : " + value);
    }
}
//-->
</script>
</head>
<body>
    <form name="myform" action="">
        <p> click the following button and see the result:</p>
        <input      type="button"      value="Get      Cookie"
onclick="ReadCookie()"/>
    </form>
</body>
</html>

```

Note: Here length is a method of Array class which returns the length of an array. We have discussed Arrays in unit 1.

Output:

click the following button and see the result:



Note: There may be some other cookies already set on your machine. The above code will display all the cookies set on your machine.

Setting expiry of cookies-

You can extend the life of a cookie beyond the current browser session by setting an expiration date and saving the expiry date within the cookie. This can be done by setting the 'expires' attribute to a date and time.

Example:

It illustrates how to extend the expiry date of a cookie by 1 Month.

```

<html>
    <head>
    <script type="text/JavaScript">
    <!--
        function WriteCookie()
        {
            var now = new Date();
            now.setMonth( now.getMonth() + 1 );
            cookievalue = escape(document.myform.customer.value) +
            “,”
            document.cookie="name=" + cookievalue;
            document.cookie = "expires=" + now.toUTCString() + “,”

```

```

        document.write ("Setting Cookies : " + "name=" +
cookievalue );
    }
    //-->
</script>
</head>
<body>
    <form name="formname" action="">
        Enter name: <input type="text" name="customer"/>
        <input type="button" value="Set Cookie"
onclick="WriteCookie()"/>
    </form>
</body>
</html>

```

Output:

Enter Cookie Name

Deleting cookies-

Sometimes you will want to delete a cookie so that subsequent attempts to read the cookie return nothing. To do this, you just need to set the expiry date to a time in the past.

Example:

Try the following example. It illustrates how to delete a cookie by setting its expiry date to one month behind the current date.

```

<html>
    <head>
        <script type="text/JavaScript">
            <!--
                function WriteCookie()
                {
                    var now = new Date();
                    now.setMonth( now.getMonth() - 1 );
                    cookievalue =
escape(document.myform.customer.value) + ",";
                    document.cookie="name=" + cookievalue;
                    document.cookie = "expires=" + now.toUTCString() +
                    ",";
                    document.write("Setting Cookies : " + "name=" +
cookievalue );
                }
            //-->
        </script>
    </head>
    <body>

```

```

        <form name="formname" action="">
            Enter name: <input type="text" name="customer"/>
            <input          type="button"          value="Set          Cookie"
onclick="WriteCookie()"/>
        </form>
    </body>
</html>

```

Output:

Enter Cookie Name:

2.7 DHTML

With DHTML, you can easily add effects to your web pages. Basically DHTML is a combination of HTML, JavaScript, DOM (Document Object Model), and CSS (Cascade Style Sheet). It is not a language it stands for Dynamic HTML.

- (i) HTML – it supports JavaScript, CSS and DOM.
- (ii) JavaScript - scripting standard for HTML. Manipulates elements using events.
- (iii) HTML DOM - The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate them.
- (iv) CSS – allows us to style the layout of web pages.

Note: We described HTML and CSS in CWDD 101, Unit 6 and 7 respectively.

The function of a web browser (like Internet Explorer or Firefox, etc) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret Content of the web page.

For Example:

```

<html>
    <body>
        <h1>A Heading</h1>
        <p>A paragraph.</p>
    </body>
</html>

```

Explanation:

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between <p> and </p> is displayed as a paragraph

HTML DOM (Document Object Model)-

It is the Document Object Model for HTML. It is a standard for how to get, change, add, or delete HTML elements.

HTML DOM methods are actions you can perform (on HTML Elements).

HTML DOM properties are values (of HTML Elements) that you can set or change.

Example:

The following example changes the content (the innerHTML) of the <p> element with id="show":

```
<html>
  <body>
    <p id="show"></p>
    <script>
      document.getElementById("show").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

In the example above, getElementById is a **method**, while innerHTML is a **property**.
"getElementById" method-

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id="demo" to find the element

"innerHTML" property-

- The easiest way to get the content of an element is by using the innerHTML property.
- The innerHTML property is useful for getting or replacing the content of HTML elements.
- The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

JavaScript and HTML DOM-

A JavaScript can also be used to change the content or attributes of HTML elements. To modify the content of an HTML element:

```
document.getElementById(id).innerHTML=new HTML
```

To change the attribute of an HTML element:

```
document.getElementById(id).attribute=new value
```

Example: The Below example changes the content of an h1 element:

```
<html>
  <body>
    <h1 id="header">Watch me changing</h1>
    <script type="text/JavaScript">
      document.getElementById("header").innerHTML="I am New
Text";
    </script>
  </body>
</html>
```

- The HTML document above contains an h1 element with id="header"
- We use the HTML DOM to get the element with id="header"
- A JavaScript changes the content (innerHTML) of that element

Note- You also have learnt about JavaScript in CWDD 102, Unit 1.

JavaScript and CSS-

A JavaScript can also change the style of HTML elements. To change the style of an HTML element:

`document.getElementById(id).style.property=new style`

Example: Below example changes color of heading

```
<html>
  <body>
    <h1 id="header"
onlick="document.getElementById('header').style.color='red'">
      Click here to change my color
    </h1>
  </body>
</html>
```

Explanation:

- The HTML document above contains an h1 element with id="header"
- We use the HTML DOM to get the element with id="header"
- A JavaScript changes the style.color of that element

Example: Here is an example of DHTML, using JavaScript onclick event and CSS style property.

```
<html>
  <body>
    <p id="para">This ia an example of DHTML. You can add effects and
dynamically change your page style using JavaScript events and CSS.<br />
You are now capable to learn methods that can be used to design a dynamic and
interactive web page. Welcome to the world of Magic...
    </p>
    <input          type="button"          value="Hide          text"
onlick="document.getElementById('para').style.visibility='hidden'" />
    <input          type="button"          value="Show          text"
onlick="document.getElementById('para').style.visibility='visible'" />
  </body>
</html>
```

2.8 POINTS TO REMEMBER

- JavaScript handles user responses with the help of events.
- Onclick, onsubmit, onmouseover, onmouseout are some useful events.
- Object is a collection of property and methods.
- You can create object as: `object = {key:value, key:value.....};`
- The new operator is used to create an instance of an object.

- This operator can be used to initialize object properties.
- JavaScript events are used to validate form and data.
- A JavaScript can be used to store and retrieve information on the visitor's computer means it can also be used to create cookies.
- Cookies are a simple plain text data record of 5 variable-length fields.
- document.cookie object is used to store cookie values.
- DHTML is a combination of HTML, JavaScript, DOM, and CSS. It is not a language it stands for Dynamic HTML.
- HTML stands for Hyper Text Markup Language
- CSS stands for Cascading Style Sheets
- The Document Object Model of HTML is a standard to get, change, add, or delete HTML elements.
- Dynamically you can change style of HTML by JavaScript.

2.9 GLOSSARY

DHTML: It stands for Dynamic hypertext transfer protocol. It is a Combination of HTML, CSS and JavaScript.

HTML 5: HTML 5 is a revision of the Hypertext Markup Language (HTML), the standard programming language for describing the contents and appearance of Web pages.

Constructor: In class-based object-oriented programming, a constructor is a special type of method called to create an object.

HTTP: The Hypertext Transfer Protocol (HTTP) is an application protocol for World Wide Web.

Stateless Protocol: HTTP is a stateless protocol, which means that the connection between the browser and the server is lost once the transaction ends

Cookies: Cookies are small files which are stored on a user's computer. They are designed to hold a modest amount of data specific to a particular client and website, and can be accessed either by the web server or the client computer.

CGI programming: CGI is the abbreviation of Common Gateway Interface. ... A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language, including C, Perl, Java, or Visual Basic.

2.10 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type)

- (1) JavaScript's interaction with HTML is handled through
- (2) On submit event is called when we form to server.

- (3) event is triggered when the HTML element is changed.
- (4) Name two events that are triggered when keyboard is used.
- (5) JavaScript are containers for property and methods.
- (6) Object are also known as keys.
- (7) [function]Methods are that let the object do something.
- (8) operator is used to create an instance of object.
- (9) is done to make sure that all mandatory fields are filled.
- (10) Final validation is mostly done at Event.
- (11) is done to ensure that valid data is entered.
- (12) Cookies are data that has sent to client's computer.
- (13) Name the 5 variable length fields of cookies.
- (14) JavaScript Cookies are stored inside object.
- (15) Cookies are store as pairs separated by semicolon.
- (16) Cookies are deleted by setting the expiry date to
- (17) JavaScript is a combination of
- (18) HTML is a language
- (19) CSS stands for
- (20) HTML DOM methods are we perform on HTML elements.
- (21) is the syntax to change the style of an element.
- (22) JavaScript is used to validate data at client's computer before sending it to server. (True/False)
- (23) Page load is also an event. (True/False).
- (24) On click event can be used for page validation. (True/ False).
- (25) CSS selector is an HTML element. (True/False)
- (26) getElementById is a method. (True/False)
- (27) innerHTML property is used to change or get the contents of elements. (True/ False)
- (28) Cookies re plain text files. (True/ False)
- (29) JavaScript arrays are objects. (True/False)
- (30) We can create object as var stud = new object(); (True/False)

Part- B (Descriptive type)

- (a) Which event is fired when a document and all of its external resources are fully loaded and displayed to the user?
- (b) What are the Event Handlers in JavaScript?
- (c) Which syntax is used to describe elements in CSS?

- (d) Which of the following can be used to select HTML elements based on the value of their name attributes?
- (e) Explain JavaScript objects.
- (f) What are JavaScript standard objects?
- (g) What is InnerHTML Property in JavaScript?
- (h) What are Cookies in JavaScript?

Answers- (Part- A- Fill in the blanks and True/False type)		
1- Events	2- submit, data	3- onchange
4- Onkeypress and onkeyup	5- Objects	6- Properties
7- function	8- new	9- Basic validation
10- Form submit	11- Data Validation	12- Server
13- Expires, domain, path, secure, name=value	14- Document.cookie	15- Name=value
16- A time in past	17- HTML, JavaScript, DOM and CSS	18- markup
19- cascading style sheet	20- action	21- document.getElementById(id).style.property=new style
22- True	23- True	24- True
25- True	26- True	27- True
28- True	29- True	30- True

2.11 BIBLIOGRAPHY/REFERENCES

- www.w3schools.com/js/js_events.asp, January 2018
- www.tutorialspoint.com/javascript/javascript_events.htm, January 2018
- www.html-5-tutorial.com, January 2018

2.12 SUGGESTED READINGS

Duckett Jon (2014) Javascript & JQuery: Interactive Front-End Web Development, Wiley, 640.

UNIT 3

JQUERY

3.1. INTRODUCTION

3.2. OBJECTIVE

3.3. JQUERY: BASIC STRUCTURE AND ACCESSING RULE

3.4. HOW TO USE JQUERY

3.5. JQUERY SYNTAX

3.6. HOW TO WRITE FUNCTIONS IN A SEPARATE FILE

3.7. ACCESSING JQUERY LIBRARIES

3.8. HOW TO USE CUSTOM SCRIPTS

3.9. POINTS TO REMEMBER

3.10. GLOSSARY

3.11. CHECK YOUR PROGRESS

3.12. BIBLIOGRAPHY/REFERENCES

3.13. SUGGESTED READINGS

3.1 INTRODUCTION

In previous two units of this book (CWDD 102) you have learnt introductory JavaScript and JavaScript events. You also have studied HTML, CSS, CSS3 and HTML5, now you are able to learn JQuery and can develop an application based on JQuery.

What is JQuery?

- JQuery is a library of JavaScript.
- JQuery is a quick and short JavaScript Library created by John Resig in 2006 with a nice motto – “Write less, do more”.
- JQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

- JQuery is a JavaScript toolkit designed to simplify various tasks by writing less code.
- JQuery greatly simplifies JavaScript programming.

Features of JQuery

DOM manipulation–

The jQuery made it easy to select DOM (Document Object Model) elements, traverse them and modifying their content by using cross-browser open source selector engine called Sizzle.

Event handling–

The jQuery suggests an graceful way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.

AJAX Support–

The jQuery facilitates you a lot to develop a responsive and feature-rich site using AJAX technology.

Animations–

The jQuery comes with a lot of built-in animation effects which you can use in your websites.

Lightweight–

The jQuery is very lightweight library - about 19KB.

Cross Browser Support–

The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+.

Latest Technology –

The jQuery supports CSS3 selectors and basic XPath¹ syntax.

3.2 OBJECTIVE

After successful completion of this unit you will be able to-

- Apply JQuery library functions in your web page.
- Include JavaScript in your HTML page.
- Create your own interactive web page.
- Develop logical thinking while using logical constructs of JavaScript.

¹ XPath stands for XML Path Language. XPath uses "path like" syntax to identify and navigate nodes in an XML (eXtensible Markup Language) document.

3.3 JQUERY: BASIC STRUCTURE AND ACCESSING RULE

For being a jquery programmer, all you need to learn is some basic concepts discussed below.

DOM (Document Object Model)-

Definition of DOM as put by the W3C (World Wide Web Consortium) is– “The Document Object Model (DOM) is an application programming interface (API) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated”.

The Document Object Model is a tree structure of various elements of HTML as I discussed in previous unit.

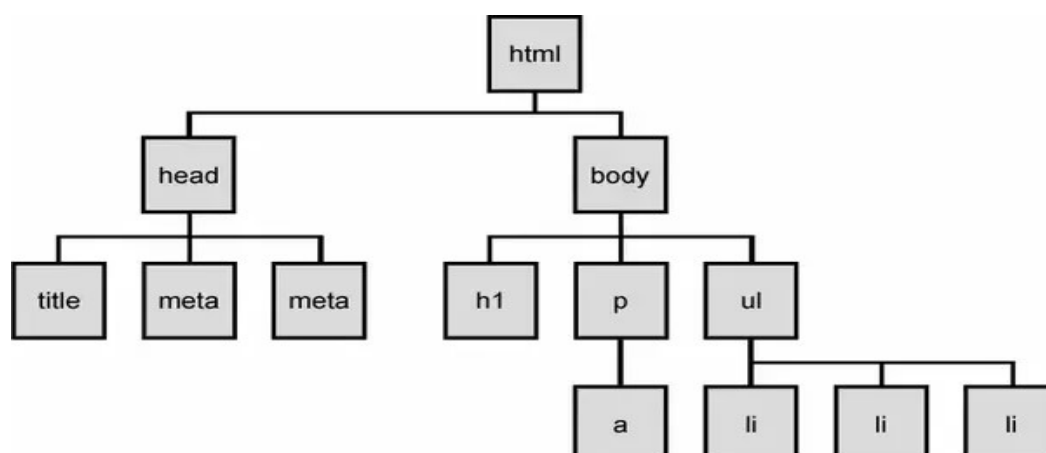
Example–

```
<html>
  <head>
    <title>The jQuery Example</title>
  </head>
  <body>
    <div>
      <p>This is a paragraph.</p>
      <p>This is second paragraph.</p>
      <p>This is third paragraph.</p>
    </div>
  </body>
</html>
```

Following are the important points about the above example given as a tree structure–

- The <html> is the ancestor (parent) of all the other elements; in other words, all the other elements are descendants (child) of <html>.
- The <head> and <body> elements are not only descendants, but children of <html>, as well.
- Likewise, in addition to being the ancestor of <head> and <body>, <html> is also their parent.
- The <p> elements are children (and descendants) of <div>, descendants of <body> and <html>, and siblings of each other <p> elements.

Following example illustrates document object hierarchy.



Let's see below example here, person and email are elements, person has two attributes name and age.

DOM creates object for person element with two attributes name and age, person element has 1 child email object with text property. A parent element shall have as many child possible, but one child object can have only one parent, hence this forms a Tree hierarchy.

```

<person name= "Neeraj" age= "50">
    <email>neeraj@gmail.com</email>
</person>
  
```

If you look closely, HTML represents similar form

```

<html>
    <head>
        <title>Welcome</title>
    </head>
    <body>
        <div id= "message">
            Welcome to JQuery
        </div>
    </body>
</html>
  
```

You can add/remove HTML elements dynamically into loaded DOM structure to have more interactive web page.

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page

- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

HTML DOM-

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements
- The HTML DOM is a standard to get, change, add, or delete HTML elements.
- HTML DOM methods are actions you can perform (on HTML Elements).
- HTML DOM properties are values (of HTML Elements) that you can set or change.

getElementById method-

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id="demo" to find the element.

innerHTML property-

The easiest way to get the content of an element is by using the innerHTML property.

- The innerHTML property is useful for getting or replacing the content of HTML elements.
- The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

Some examples to find HTML elements using different methods. As-

Method	Description
document.getElementById(id)	Find an element by element id
document.getElementsByTagName(name)	Find elements by tag name
document.getElementsByClassName(name)	Find elements by class name

Some examples to change HTML elements. As-

Method	Description
--------	-------------

element.innerHTML = new html content	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.setAttribute(attribute, value)	Change the attribute value of an HTML element
element.style.property = new style	Change the style of an HTML element

Example:

```
<html>
  <body>
    <p id="first">Hello World!</p>
    <p>This example explains getElementById method and innerHTML
    property!</p>
    <p id="new"></p>
    <script>
      var myElement = document.getElementById("first");
      document.getElementById("new").innerHTML = "The text of first
      paragraph is " + myElement.innerHTML;
    </script>
  </body>
</html>
```

Output:

Hello World!

This example explains getElementById method and innerHTML property!

The text of first paragraph is Hello World!

3.4 HOW TO USE JQUERY

Readily available are the two ways to use JQuery into your web page. As-

- (i) Local installation
- (ii) CDN (Content Delivery Network) based version

Local installation-

You can download JQuery library on your local machine and include it in your HTML code.

- Go to the <https://jquery.com/download/> to download the latest version available.
- Now put downloaded jquery-2.1.3.min.js file in a directory of your website, e.g. /jquery.

Example:

Now you can contain jquery library in your HTML file as follows –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = “text/JavaScript”      src = “/jquery/jquery-
2.1.3.min.js”></script>
    <script type = “text/JavaScript”>
      $(document).ready(function(){
        document.write(“Hello, World!”);
      });
    </script>
  </head>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```

CDN (Content Delivery Network) based version-

You can include jQuery library into your HTML code directly from Content Delivery Network. Google and Microsoft provide content deliver for the latest version. We are using Google CDN version of the library throughout this tutorial.

Example:

Now let us rewrite above example using jQuery library from Google CDN. As-

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = “text/JavaScript”
src=“https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js”>
    </script>
    <script type = “text/JavaScript”>
      $(document).ready(function(){
        document.write(“Hello, World!”);
      });
    </script>
  </head>
  <body>
    <h1>Hello</h1>
```



```

        </body>
    </html>

```

This will produce following result after loading page–

Hello, World!

3.5 JQUERY SYNTAX

The jQuery syntax is tailor-made for selecting HTML elements and performing some action on the element(s). As-

Basic syntax is:

```
$(selector).action()
```

Explanation of the above syntax-

- A \$ sign to define/access jQuery
- A (selector) to “query (or find)” HTML elements
- A jQuery action() to be performed on the element(s)

Example:

```

<html>
    <head>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script>
            $(document).ready(function(){
                $("p").click(function(){
                    $(this).hide();
                });
            });
        </script>
    </head>
    <body>
        <p>If you click on me, I will disappear.</p>
        <p>Also Click me to hide!</p>
    </body>
</html>

```

The Document ready function-

All jQuery methods are put inside a document ready event- As:

```
$(document).ready(function(){
```

```
// jQuery methods go here...

});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

Execute jQuery only when the complete DOM objects (the complete page has been loaded). You will have to wrap your code in .ready function.

```
<script language="JavaScript" type="text/JavaScript">

    $(document).ready(function () {

        $("#div1").css("border", "2px solid green");

    });

</script>
```

This is the better and safer way to execute jQuery. This makes sure that jQuery code will execute only if complete page has been loaded in the browser so you are rest assured that user will not see any undesired behaviour on the page.

The \$() factory event-

jQuery selectors start with the dollar sign and parentheses– \$(). The factory function \$() makes use of following three building blocks while selecting elements in a given document–

- (i) **Tag Name:** Represents a tag name available in the DOM. For example \$('p') selects all paragraphs <p> in the document.
- (ii) **Tag ID:** Represents a tag available with the given ID in the DOM. For example \$('#some-id') selects the single element in the document that has an ID of some-id.
- (iii) **Tag Class:** Represents a tag available with the given class in the DOM. For example \$('.some-class') selects all elements in the document that have a class of some-class.

Note– The factory function \$() is a synonym of jQuery() function. So in case you are using any other JavaScript library where \$ sign is conflicting with something else then you can replace \$ sign by jQuery name and you can use function jQuery() instead of \$().

Example:

Following is a simple example which makes use of Tag Selector. This would select all the elements with a tag name p and will set their background to “yellow”.

```
<html>

    <head>

        <title>The jQuery Example</title>
```

```

        <script src =
        “https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js”> </script>
        <script language = “JavaScript”>
            $(document).ready(function() {
                $(<“p”>).css(<“background-color”, “yellow”>);
            });
        </script>
    </head>
    <body>
        <div>
            <p class = “myclass”>This is a paragraph.</p>
            <p id = “myid”>This is second paragraph.</p>
            <p>This is third paragraph.</p>
        </div>
    </body>
</html>

```

JQuery Selectors-

Following table lists down few basic selectors and explains them with examples.

- (i) Name Selector
- (ii) Id Selector
- (iii) Class Selector

Name Selector-

It selects all elements which match with the given *element Name*. The jQuery element selector selects elements based on the element name. You can select all <p> elements on a page like this:

```
$(<“p”>)
```

Example:

When a user clicks on a button, all <p> elements will be hidden:

```

$(document).ready(function(){
    $(<“button”>).click(function(){
        $(<“p”>).hide();
    });
});

```

Id Selector-

The jQuery `#id` selector uses the `id` attribute of an HTML tag to find the specific element. An `id` should be unique within a page, so you should use the `#id` selector when you want to find a single, unique element. To find an element with a specific `id`, write a hash character, followed by the `id` of the HTML element.

```
$("#test")
```

Example:

When a user clicks on a button, the element with `id="test"` will be hidden:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide();  
    });  
});
```

Class Selector-

The jQuery class selector finds elements with specific class. To find elements with a specific class, write a period character (`.`), followed by the name of the class: As-

```
$(".test")
```

Example:

When a user clicks on a button, the elements with `class="test"` will be hidden:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide();  
    });  
});
```

Note:

- `$("*")` Selects all elements
- `$(this)` Selects the current HTML element
- `$("#p.intro")` Selects all `<p>` elements with `class="intro"`
- `$("#p:first")` Selects the first `<p>` element
- `$("#ul li:first")` Selects the first `` element of the first ``
- `$("#ul li:first-child")` Selects the first `` element of every ``
- `$("#:button")` selects all `<button>` elements and `<input>` elements of type `"button"`

3.6 HOW TO WRITE JQUERY FUNCTIONS IN A SEPARATE FILE?

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file. When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):

Example:

```
<head>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"> </script>
    <script src="my_jquery_functions.js"> </script>

</head>
```

3.7 ACCESSING JQUERY LIBRARIES

As almost everything we do when using jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

If you want an event to work on your page, you should call it inside the \$(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

To do this, we register a ready event for the document as follows–

```
$(document).ready(function() {
    // do stuff when DOM is ready
});
```

To call upon any jQuery library function, use HTML script tags as shown as under–

```
<html>

    <head>

        <title>The jQuery Example</title>
        <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
        <script language = "JavaScript">
            $(document).ready(function() {
                $("div").click(function() {alert("Hello, world!");});
            });
        </script>
    </head>
```

```

<body>
    <div id = "mydiv">
        Click on this to see a dialogue box.
    </div>
</body>
</html>

```

3.8 HOW TO USE CUSTOM SCRIPTS?

It is better to write our custom code in the custom JavaScript file: custom.js, as follows–

```

/* Filename: custom.js */
$(document).ready(function() {
    $("div").click(function() {
        alert("Hello, world!");
    });
});

```

Now we can include custom.js file in our HTML file as follows –

```

<html>
    <head>
        <title>The jQuery Example</title>
        <script src = "https://ajax.googleapis.com
        /ajax/libs/jquery/2.1.3/ jquery.min.js"></script>
        <script src = "/jquery/custom.js"></script>
    </head>
    <body>
        <div id = "mydiv">
            Click on this to see a dialogue box.
        </div>
    </body>
</html>

```

We can use multiple libraries here in a single document.

Note – Conflict possibility

Many JavaScript libraries use \$ as a function or variable name, just as jQuery does. In jQuery's case, \$ is just an alias for jQuery, so all functionality is available without using \$.

Run \$.noConflict() method to give control of the \$ variable back to whichever library first implemented it. This helps to make sure that jQuery doesn't conflict with the \$ object of other libraries.

Here is simple way of avoiding any conflict:

```
// Import other library  
// Import jQuery  
$.noConflict();  
jQuery(document).ready(function($) {  
    // Code that uses jQuery's $ can follow here.  
});  
// Code that uses other library's $ can follow here.
```

3.9 POINTS TO REMEMBER

- jQuery is a fast and short JavaScript Library created by John Resig in 2006 with a nice motto – Write less, do more.
- jQuery allows us to manipulate DOM.
- *getElementById* method allows access to an HTML element by the *id* of the element.
- JavaScript can change all the HTML elements, their attributes and CSS styles in the page.
- Basic JQuery syntax is: \$(selector).action()
- If you want an event to work on your page, you should call it inside the \$(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

3.10 GLOSSARY

- **AJAX:** AJAX = Asynchronous JavaScript And XML.
- **Sizzle:** A pure-JavaScript CSS selector engine designed to be easily dropped in to a host library.
- **CSS3:** CSS is a language that describes the style of an HTML document.
- **XPATH:** XPath stands for XML Path Language. XPath uses “path like” syntax to identify and navigate nodes in an XML document

3.11 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type)

- (a) JQuery is a javaScript
- (b) JQuery was created by in

- (c) JQuery allows us to select, traverse and modify
- (d) DOM is a tree structure of elements of
- (e) Using we can add and remove HTML elements dynamically.
- (f) What is the syntax to find element in DOM using its Tag name?
- (g) What is the basic syntax of JQuery?
- (h) We write our JQuery code inside to prevent it from running before the document is finished loading.
- (i) \$('<h1>') is used to select in the document.
- (j) How will you select <p class = "myclass"> using JQuery?
- (k) JQuery selectors are used to select elements using..... , and
- (l) jQuery functions are written in a separate file with extension.
- (m) Write the syntax to include external file in JavaScript.
- (n) \$ is an alias for
- (o) method allows us to make sure that no two libraries are conflicted
- (p) jQuery does not provide cross-browser support. (True/False)
- (q) <p>,<h1>, elements are descendents of <body>.(True/False)
- (r) You can download JQuery into your local machine. (True/False)
- (s) CDN based version allows us to always get updated version of JQuery. (True/False) name?

Part- B (Descriptive type)

- (a) What is JQuery?
- (b) How will you make sure that DOM is ready using jQuery?
- (c) What is a JQuery Selector?
- (d) How to select elements using jQuery with the given element tag-name?
- (e) How to set the style property of an element using jQuery?
- (f) How to select all elements using jQuery?
- (g) How to get the direct parent of an element using jQuery?
- (h) How to set the width of an element using jQuery?

Answers- (Part- A- Fill in the blanks and True/False type)			
(a) Library	(b) John Resig in 2006	(c) Document	Object

		Model
(d) HTML	(e) JavaScript	(f) document.getElements ByTagName(name)
(g) \$(selector).action()	(h) \$(document).ready()	(i) [All the headings of level 1
(j) \$('myclass')	(k) Name, Id and class	(l) .js
(m) <script src="my_jquery_ functions.js"></script>	(n) JQuery	(o) \$.noConflict()
(p) False	(q) True	(r) True
(s) True		

3.12 BIBLIOGRAPHY/REFERENCES

- www.w3schools.com/js/js_events.asp, January 2018
- www.tutorialspoint.com/javascript/javascript_events.htm, January 2018
- www.html-5-tutorial.com, January 2018
- www.jquery.com, January 2018

3.13 SUGGESTED READINGS

Duckett Jon (2014) Javascript & JQuery: Interactive Front-End Web Development, Wiley, 640.

UNIT 4

INTRODUCTION TO AJAX

- 4.1. INTRODUCTION
- 4.2. OBJECTIVE
- 4.3. WHAT YOU NEED TO USE AJAX?
- 4.4. WHAT IS AJAX?
- 4.5. XML – A BRIEF INTRODUCTION
- 4.6. “XMLHTTPREQUEST” OBJECT
- 4.7. AJAX REQUEST
- 4.8. AJAX RESPONSE
- 4.9. SAMPLE WEB-PAGE USING AJAX – CALLING DATA FROM TEXT FILE
- 4.10. SAMPLE WEB-PAGE USING AJAX – CALLING DATA FROM XML FILE
- 4.11. AJAX SECURITY
- 4.12. POINTS TO REMEMBER
- 4.13. GLOSSARY
- 4.14. CHECK YOUR PROGRESS
- 4.15. BIBLIOGRAPHY/REFERENCES
- 4.16. SUGGESTED READINGS

4.1 INTRODUCTION

Imagine a website always reloading its content when clicked on a hyperlink. You usually have to wait for the whole page to refresh before you see any result. Ajax gives rid of all the reloading pages that give poor web experience. With Ajax you can display data in the browser, and download that data without refreshing the particular page. This all can be done behind the scenes with the help of Ajax.

Ajax can be thought as web developer's dream. This unit gives a complete introduction to Ajax. Ajax is a new technique of web development for creating enhanced, quicker and more interactive web application.

4.2 OBJECTIVE

After successful completion of this unit you will be able to-

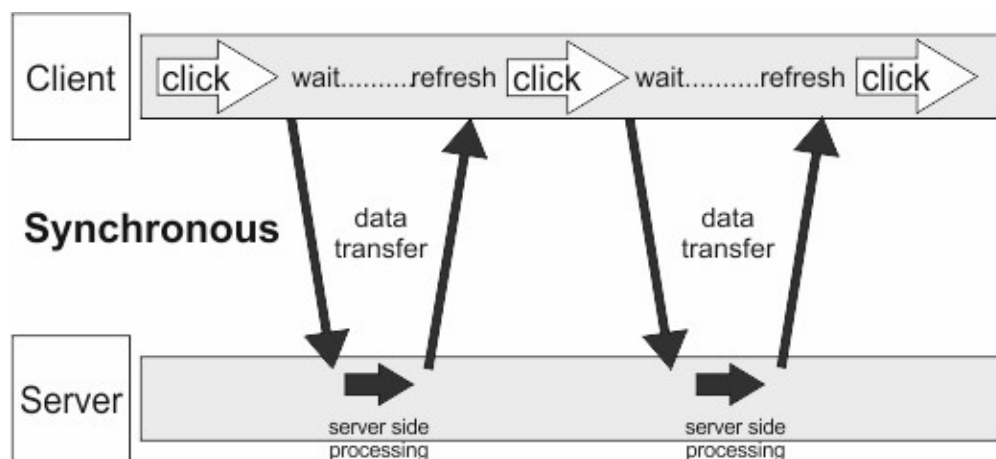
- Know AJAX Concepts
- Know to send AJAX requests
- Know to receive AJAX response
- Know calling data from text file and xml file as well

4.3 WHAT YOU NEED TO USE AJAX?

To understand AJAX concepts better, knowledge of HTML & JavaScript is needed. You do not have to be an expert in these topics, but basic understanding of web page creation is needed. As far as the requirement of software is concerned, you need a browser that can run JavaScript (such as Google Chrome, Internet Explorer or Mozilla Firefox) and a server (such as Apache or IIS).

4.4 WHAT IS AJAX?

AJAX stands for Asynchronous JavaScript and XML, is a set of techniques for creating



highly interactive websites and web applications. Ajax is all about updating content of webpages without reloading it.

Traditional Web page approach (Synchronous):

The web page contains forms and links that, when clicked make a request to a new URL (Web address) on the web server. The server then responds by sending the whole new page (HTML) to the web browser, and browser displays it replacing the original page. This approach is time consuming, as user has to wait while the new page loads.

The client is blocked by synchronous request until operation completes (figure).

AJAX Approach (Asynchronous):

The request and response from client and server are happen in background without affecting the overall experience of the user. The whole process is invisible to the user.

An Asynchronous request does not block the client (figure) i.e user can perform another operation at that time.

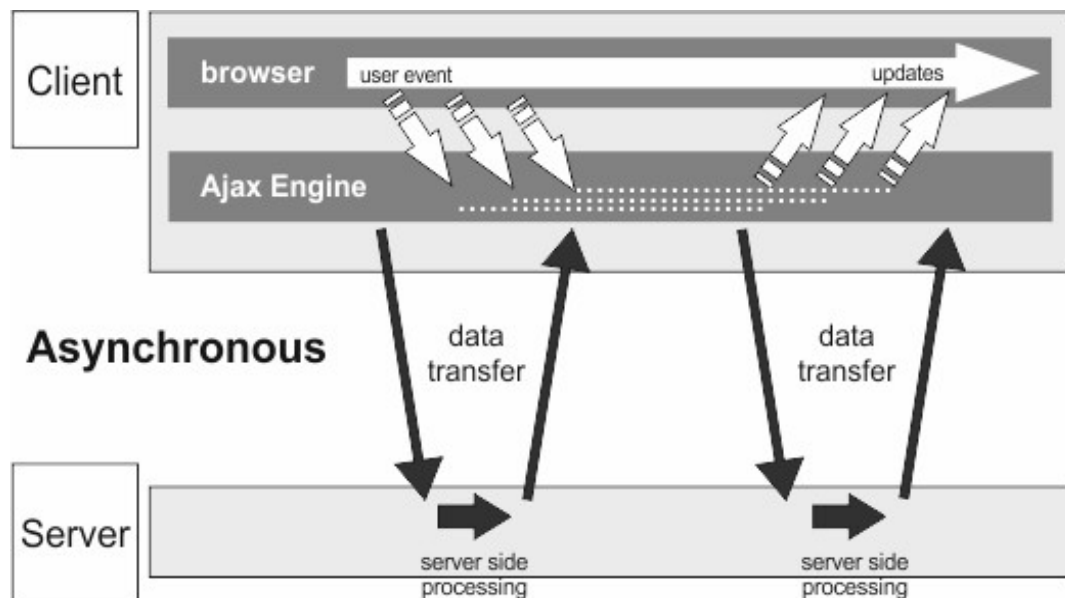


Fig 4.2 (Asynchronous Request and Response)

4.5 XML – A BRIEF INTRODUCTION

XML is the short form of Extensible Markup Language. It is used to store and transport data in multiple platforms. Simply it can be thought of a simple text file having well-formed XML document.

Example- A sample XML Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<email>
  <to>Rakesh</to>
  <from>Mohan</from>
  <subject>Celebration</subject>
  <message>Birthday Party at home around 5:00 PM</message>
</email>
```

The above example explains that an email has been written to rakesh by mohan communicating about the time of celebration. It can be viewed as a document tree in browser (figure).



Figure (Displaying XML in a browser (Chrome, IE, etc.))

You can see that these tags are self-explanatory. XML file has all the information wrapped up in self descriptive tags. But still XML does not do anything; it is just used to carry data. You have to write a piece of program in any language to send, receive, store, or display XML. Some properties of XML are:

1. XML have no predefined tags.
2. XML is extensible.
3. W3C (word wide web construium) recommends XML
4. XML simplifies data sharing.

AJAX uses XML for data interchange and manipulation. When data is received from the server using XMLHttpRequest (discussed in next section) object, that data is often in XML format. Therefore we can say that XML forms the base for AJAX.

4.6 “XMLHTTPREQUEST” OBJECT

AJAX approach is handled by a javascript object called “XMLHttpRequest”. This object request information from the server and then loads that information on the web page asynchronously. Asynchronous means exchanging the information with a web server without reloading (refreshing) the web page.

You can perform the following operation with the help of “XMLHttpRequest” Object:
As-

1. Update a web page without reloading it.
2. Request/ accept data from the server after the webpage has loaded.
3. Send data to the server in the background.

Note: “XMLHttpRequest” is handled differently in different browsers. As-

For Internet Explorer (7.0 or previous versions)

```
var objXMLHTTP = new ActiveXObject("Microsoft.XMLHTTP");
```

For Browsers other than Internet Explorer

```
var objXMLHTTP = new XMLHttpRequest();
```

You can check whether the user’s browser is IE (Internet Explorer) or any other with the help of Javascript. The below section explains you, how to create a browser specific “XMLHttpRequest” object.

```
<script>
var objXMLHTTP;
if (window.XMLHttpRequest) {
    // code for all modern browsers including IE 9.0 and above.
    objXMLHTTP = new XMLHttpRequest();
} else {
    // code for old IE browsers
```

```

        objXMLHTTP = new ActiveXObject("Microsoft.XMLHTTP");
    }
</script>

```

The <script> tag is used because all the code written for AJAX is finally Javascript code.

4.7 AJAX REQUEST

As we are aware of Asynchronous and Synchronous request (refer to section 4.3) theoretically. In this section we get to know practically how these request work. To send AJAX request to the server, we have to use open() and send() methods of the XMLHttpRequest object.

Example to show sending AJAX Request-

```

<script>
    var objXMLHTTP;
    if (window.XMLHttpRequest) {
        // code for all recent browsers including IE 9.0 and above.
        objXMLHTTP = new XMLHttpRequest();
    } else {
        // code for old IE browsers
        objXMLHTTP = new ActiveXObject("Microsoft.XMLHTTP");
    }
    //code for AJAX request
    objXMLHTTP.open(GET, "data.txt", true);
    objXMLHTTP.send();
</script>

```

Description of XMLHttpRequest open() method	
Syntax: open(method, url, async)	
Parameters	Description
Method	Request Type (GET or POST)
	<ul style="list-style-type: none"> Properties of GET The GET method is used to send information to the server as part of the URL. It is simpler and faster than POST. GET request should never be used when dealing with sensitive data as it will be attached and shown in URL. It is less secure.

	<ul style="list-style-type: none"> • It also has size limit as some browser has 2048 character URL restriction. Only ASCII data can be sent to the server. • It can be bookmarked. • Parameters saved in the browser's history.
	<ul style="list-style-type: none"> • Properties of POST • The POST method is used to send information as an enclosed entity. • It is more secure than GET. Data is not shown in browser's URL. • No size limit. • It can not be bookmarked. • Parameters not saved in browser's history. • Can send parameters, including files, binary data to the server.
url	<p>This defines the location of file in server.</p> <p>In the above example we have called a file "data.txt" from the server. This parameter gives the exact location and name of file in the server from which data is called in AJAX. This url can be as simple as text/image/sound/video file or as complex as server-side scripting file used for database handling.</p>
async	<p>Boolean (true for asynchronous and false for synchronous)</p> <p>The async argument should always be true since 'A' in AJAX stands for asynchronous. Synchronous request forces the browser to wait for a response from the server before continuing. (Refer to section 4.2)</p> <p>Note: The value of async = false is not suggested because the JavaScript will not behave properly (even stop) while executing until the server response is ready. If the server is busy or slow, the request (webpage) will suspend or stop.</p>

Description of XMLHttpRequest send() method	
Syntax:	
Send()	
This method will send the XMLHttpRequest object to the server.	

4.8 *AJAX RESPONSE*

Now, when the request is sent by client it is the duty of server to reply the response back to the client. But as we are using asynchronous calls, we cannot be confident when the response will come. So, we have to write piece of code that waits and handles the response when it arrives to the client. This is done by using a callback function.

Callback function:

A function of executable code, or a piece of executable code, that is passed as an argument to other code/function. These functions can also be triggered by some automatic events using assignment operator. Below is the example of callback function used for getting AJAX response. As-

```
objXMLHTTP.onreadystatechange = function() {  
    //function body here  
};
```

We will discuss above function (callback) in detail further in this section.

The Example for getting AJAX response-

```
<script>  
    var objXMLHTTP;  
    if (window.XMLHttpRequest) {  
        // code for all recent browsers including IE 9.0 and above.  
        objXMLHTTP = new XMLHttpRequest();  
    } else {  
        // code for old IE browsers  
        objXMLHTTP = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    //below code is for AJAX Response  
    objXMLHTTP.onreadystatechange = function(){  
        if (this.readyState == 4 && this.status == 200){  
            // Display data after getting data from server  
        }  
    };  
    //code for AJAX Request  
    objXMLHTTP.open(GET, "data.txt", true);  
    objXMLHTTP.send();  
</script>
```

The “XMLHttpRequest” object’s readyState property holds the current (existing) state of the response. It has five possible states (0 to 4).

readyState property is described below-		
Property	State	Description
readyState	0	client request to server is not initialized
	1	server connection established
	2	request received by server
	3	server is processing request
	4	request finished by server and response is ready

The change in the readyState property is captured by XMLHttpRequest object's onreadystatechange event handler. To handle the response a callback function is assigned to this event handler.

```
objXMLHTTP.onreadystatechange = function() {
    //Do something here
}
```

Note: This callback function is also known as anonymous or unnamed function. This function is called again and again when state of “readState” property changes from 0 to 4.

The “status” property holds the status of “XMLHttpRequest” object. Some of the main values of status and their description are described here:		
status property is described below-		
Property	State	Description
status	200	OK (The request is OK).
	400	Bad request (The request cannot be fulfilled due to bad syntax).
	403	Forbidden (The request was a legal request, but the server is refusing to respond to it).
	404	Not Found (The requested page could not be found but may be available again in the future).

In the program above, “onreadystatechange” event occurs everytime the readyState changes. Before doing anything with the response data, we check that readyState is

equals to 4 (request finished and response is ready) and status is equals to 200 (request is OK).

When the required condition of readyState & status are met, we enter into the condition's True block. After that we have displayed the response data into a div element with id="demo" (As code written below).

The use of this keyword is to denote the current object i.e "objXMLHTTP" (name given to "XMLHttpRequest" object).

Example-

```
objXMLHTTP.onreadystatechange = function(){
    if (this.readyState == 4 && this.status == 200){
        document.getElementById("demo").innerHTML =this.responseText;
    }
};
```

Server Response Properties	
Property	Description
responseText	get the response data from server as a text string
responseXML	get the response data from server as XML data

4.9 SAMPLE WEB PAGE USING AJAX – CALLING DATA FROM TEXT FILE

In this sample web page we will use two files- one is HTML & other is TEXT file written in notepad. First of all, create a text file named as "data.txt" and write some text in that file. Below the sample text you can use for your reference. As-

"data.txt"

Asynchronous JavaScript and XML AJAX stands for Asynchronous JavaScript and XML. XMLHttpRequest Object can be used to call data from text files, databases, XML file etc. in the background unknown to the user.

Now, create an HTML file and name it as ajaxPractice.html. We will code for HTML, Javascript, & XML in this file. We will send request for data.txt file with the help of XMLHttpRequest object and then use the data for display in this HTML file.

The below example shows the handling of TEXT file with the help of AJAX-

```
<!DOCTYPE html>
<html>
  <body>
    <div id="demo">
      <button type="button" onclick="loadTxt()">Change Content</button>
    </div>
```

```

        <script>
            function loadTxt () {
                var objXMLHTTP = new XMLHttpRequest();
                objXMLHTTP.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        document.getElementById("demo").innerHTML =
this.responseText;
                    }
                };
                objXMLHTTP.open("GET", "data.txt", true);
                objXMLHTTP.send();
            }
        </script>
    </body>
</html>

```

After the above code has been written we have two files. As:

- 1) "data.txt" – It has the data which we used to call in HTML (ajaxPractice.html) page, with the help of AJAX. This text data is displayed in HTML file.
- 2) "ajaxPractice.html" – It has all the source code for HTML, JAVASCRIPT, AJAX.

Now, we have to place these files in a server and than call html file from browser. But we cannot do this at this time as we are unaware of a Server. We will get the information about server in next chapter.

According to above code, data in text file will be displayed in the div block that has id="demo". Fuction loadTxt() will be called when button is clicked.

The following operation will be performed inside the function

1. A new XMLHttpRequest object XMLHTTP will be created.
2. Onreadystatechange event will be fired automatically when the state of XMLHTTP object changed.
3. When XMLHTTP object will change its state to 4 and status to 200, the txt file data will be displayed on the div tag specified.

4.10 *SAMPLE WEB-PAGE USING AJAX – CALLING DATA FROM XML FILE*

In this sample example we will use two files one is HTML & other is XML file written in notepad. First of all, create an xml file named as "data.xml" and write some xml data in that file. Below is the sample xml file that you can use for your refrence. As-

"data.xml"

```

<?xml version="1.0" encoding="UTF-8"?>
    <bookstore>
        <book>
            <title>Let Us C</title>

```

```

        <author>Yashwant Kanetkar</author>
        <year>2010</year>
        <price>300.00</price>
    </book>
    <book>
        <title>Web Programming</title>
        <author>Dr Tewari</author>
        <year>2015</year>
        <price>400.00</price>
    </book>
</bookstore>

```

Now, create an HTML file and name it as “ajaxPractice.html”. We will code for HTML, Javascript, & XML in this file. We will send request for “data.xml” file with the help of XMLHttpRequest object and then use the data for display in this HTML file. Below is the sample example for handling XML file with the help of AJAX:

```

<!DOCTYPE html>
<html>
    <body>
        <div id="demo">
            <button type="button" onclick="loadTxt()">Change Content</button>
        </div>
        <script>
            function loadTxt () {
                var objXMLHTTP = new XMLHttpRequest();
                objXMLHTTP.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        displayData(this);
                    }
                };
                objXMLHTTP.open("GET", "data.xml", true);
                objXMLHTTP.send();
            }
            function displayData(xml){
                var i;
                var xmlDoc = xml.responseXML;
                var table="<tr><th>Title</th><th>Author</th></tr>";
                var x = xmlDoc.getElementsByTagName("book");
                for (i = 0; i < x.length; i++) {
                    table += "<tr><td>" +
                        x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue +
                        "</td><td>" +
                        x[i].getElementsByTagName("author")[0].childNodes[0].nodeValue +
                        "</td></tr>";
                }
                document.getElementById("demo").innerHTML = table;
            }
        </script>
    </body>
</html>

```

```

    }
  </script>
</body>
</html>

```

Now, we have to place these files in a server and then call html file from browser.

According to above code data in xml file will be displayed in the div block that has id="demo". Function loadTxt() will be called when button is clicked.

The following operation will be performed inside the function

1. A new XMLHttpRequest object XMLHttpRequest will be created.
2. Onreadystatechange event will be fired automatically when the state of XMLHttpRequest object changed.
3. When XMLHttpRequest object will change its state to 4 and status to 200, a new function displayData() is called and XMLHttpRequest object will pass into that function.
4. Inside the displayData() function XMLHttpRequest object will be copied to xml object.
5. Data from xml object will be transferred to xmlDoc variable with the help of `xml.responseXML` code fragment.
`var xmlDoc = xml.responseXML;`
6. Title and author name will be extracted from xmlDoc variable with the help of `getElementsByTagName()` method.
`var x = xmlDoc.getElementsByTagName("book");`
7. Nodes (elements) are extracted from the XML file, and finally updated into element "demo" with the HTML table filled with XML data.
8. `document.getElementById("demo").innerHTML = table;`

4.11 *AJAX SECURITY*

1. An AJAX-based Web application does not have any in-built security features. If not handled properly, they have the same server & client-side security concerns as regular Web applications.
2. JavaScript code is always detectable to a user/hacker. Hacker can use this code for executing malicious program in Server.
3. If XMLHttpRequest object is passed using non-SSL connection, then it could lead to various attacks like
 - a. SQL injection,
 - b. Cross Site Scripting (XSS),
 - c. Denial of Service
 - d. Cross Site Request Forgery (CSRF)

4.12 *POINTS TO REMEMBER*

- AJAX always sends and receives data to and from the server; therefore, Apache http server (included in XAMPP) has to be installed before working on AJAX.
- You can also use WAMP (windows), LAMP (Linux) for the http server in place of XAMPP.

4.13 GLOSSARY

- **HTML:** Hyper Text Markup Language is a markup language for creating web pages.
- **JAVASCRIPT:** Javascript is an interpreted scripting language commonly used in web development.
- **XML:** Extensible markup language (XML) is used to store and transport data in text format.
- **ASynchronous:** Asynchronous means that the script will send a request to the server, and continue it's execution without waiting for the reply.
- **Synchronous:** Synchronous means that the script will send a request to the server, and wait for the reply from the server.
- **GET:** The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI.
- **POST:** The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.
- **CallBack:** A callback function, also known as a higher-order function, is a function that is passed to another function as a parameter.
- **XMLHttpRequest:** XMLHttpRequest object is used by all modern browsers to request data from a server.
- **HTTP:** HTTP is a hyper text transfer protocol used as a request-response protocol between a client and server.
- **XMLHttpRequest object forms the beginning of Ajax.** An Ajax request is initiated from a browser by first creating an XMLHttpRequest object and opening the XMLHttpRequest object using the open() method. The method used to create the XMLHttpRequest varies with the browser used.
- **An Ajax request is sent using the send() method of XMLHttpRequest.** When the request completes the Ajax XML response is retrieved using the responseXML attribute of the XMLHttpRequest object. The web page that sent the Ajax request is updated with the Ajax XML response by retrieving the XML data and setting the data into the web page elements using DOM functions

4.14 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type questions)

- (a) AJAX Stands for
- (b) AJAX is made popular by.....
- (c) AJAX allows page content to be updated without requiring a full page reload.
- (d) AJAX is concise and simple to use.
- (e) To use AJAX every XML document must have a DOCTYPE tag.
- (f) Browser based applications and platform independent applications are used by Ajax.
- (g) Bandwidth utilization, More interactive and Speeder retrieval of data are the advantages of Ajax.
- (h) AJAX is dependent on Javascript. If there is some Javascript problem with the browser or in the OS, Ajax will not work.
- (i) Live data binding is not a key feature of AJAX.
- (j) In AJAX "XMLHttpRequest" is used to sends data in the background, receives data and updates data without reloading the page.

Part- B (Short answer type questions)

- (1) Define Ajax.
- (2) What are the advantages of Ajax?
- (3) What are the technologies used by Ajax?
- (4) What are the features of Ajax?
- (5) What are the real web applications of AJAX currently running in the market?
- (6) What are the security issues with AJAX?
- (7) What is the difference between synchronous and asynchronous requests?
- (8) What does mean by XMLHttpRequest?
- (9) What are the properties of XMLHttpRequest object?
- (10) Explain callback function used in Javascript.
- (11) What is the name of object used for AJAX request?
- (12) How many types of ready states in Ajax?
- (13) Which request is better, Get or Post? Why?
- (14) Differentiate between responseText & responseXML property.
- (15) When can we use Ajax? Give few examples?
- (16) Write method in javascript to get information from text files using AJAX.
- (17) Is Ajax browser dependent or browser independent script? Explain why?

- (18) What is W3C? What is the work of W3C?
- (19) What is DOM?
- (20) How XMLHttpRequest is different from HTTP request? Explain.
- (21) Define GET() and POST() method in detail. Explain the merits and demerits of both.
- (22) What is XML? How to represent the XML Document?
- (23) What is AJAX? What are the types of request handled by AJAX?
- (24) What are the properties of XMLHttpRequest? Explain in detail.
- (25) What is Rich Internet Application (RIA)?

Answers- (Part- A- Fill in the blanks and True/False type)		
Asynchronous Javascript and XML	Google	True
True	False	True
True	True	False
True		

4.15 BIBLIOGRAPHY/REFERENCES

- Steven Holzner, 2009, “AJAX, A Beginner’s Guide”, published by McGraw Hill.
- Edmond Woychowsky, 2006, “Ajax Creating Web Pages with Asynchronous JavaScript and XML”, published by Prentice Hall
- https://www.w3schools.com/xml/ajax_intro.asp
- <http://www.tutorialspoint.com/ajax/>
- https://www-s.acm.illinois.edu/webmonkeys/files/AJAX_with_jQuery.pdf
- <http://www.springer.com/gp/book/9783540775959>
- <https://www.phptpoint.com/ajax-tutorial/>

4.16 SUGGESTED READINGS

- “Ajax: The Complete Reference”, Thomas A. Powell, ISBN-13: 978-0071492164
- “Head First Ajax”, Rebecca M. Riordan, ISBN-13: 978-0596515782.
- https://www.w3schools.com/xml/ajax_intro.asp

UNIT 5

SERVER CONFIGURATION FOR SERVER-SIDE PROGRAMMING & INTRODUCTION TO PHP

5.1. INTRODUCTION

5.2. OBJECTIVE

5.3. PREREQUISITE

5.4. WHAT IS XAMPP?

5.5. INTRODUCTION TO PHP

5.6. BASIC PROGRAMMING COSTRUCTS IN PHP

5.7. POINTS TO REMEMBER

5.8. GLOSSARY

5.9. CHECK YOUR PROGRESS

5.10. BIBLIOGRAPHY/REFERENCES

5.11. SUGGESTED READINGS

5.1 INTRODUCTION

In unit 4 we have learned about AJAX (Asynchronous JavaScript and XML) where we saw minimum software requirement to execute AJAX in your web page is browser and a server. We are aware about browsers, here we will discuss in detail about servers, there functions and importance.

A server is a computer that provides services to its clients (other computers). A server processes requests send by client and delivers response (data) over a network connection. Basically, web server is used to host the web sites (web pages) but there exist other web servers also such as gaming, storage, FTP, e-mail etc. There are number of webservers available working for specific technologies.

Some examples of servers are-

1. Apache HTTP server – generally used for PHP/MySql etc.
2. IIS-(Internet Information Service) – generally used for ASP, ASP.net & other Microsoft technologies.
3. Apache Tomcat server – generally used for JAVA technologies such as Servlets, JSP, Beans etc.

4. Oracle iPlanet Web Server – generally used for Oracle based technologies.

In this course (CWDD) we are using PHP, so we used “Apache HTTP server” in our examples throughout this book. To install “Apache HTTP server” into your computer system you should use “XAMPP²” package supplied by Apache. This (XAMPP) package will install Apache, PHP & MySQL in your Computer system.

5.2 OBJECTIVES

After successful completion of this unit you will be able to-

- Download and Configure Server (XAMPP)
- Install Server in your Local machine and can execute programs on server.
- Maintain Active Directory
- Know basic PHP concepts.

5.3 PREREQUISITE

Why would I want to install XAMPP?

Because, it allows development and testing on my local machine instead of on a remote (and probably live) server. I use XAMPP to test out changes to my web pages before uploading and committing changes to the real site.

Download XAMPP [its open source].

Knowledge of HTML, XML is needed.

5.4 WHAT IS XAMPP?

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). XAMPP is small and light-weighted Apache software containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP (Server-side scripting) and MySQL (database).

XAMPP is available as a free download (open source). The version contains Apache HTTP Server, PHP, MySQL, phpMyAdmin. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows operating systems. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Components of XAMPP

XAMPP has the following primary components. As-

Apache-

Apache is the most widely and number one used web server software. It is free and cross-platform software.

MySQL-

² The “X” in XAMPP stands for “operating system”. The rest of the letters stands for: Apache, MySQL, PHP, and PERL. Here “X” means any operating system.

Almost every web application, requires a database for storing data. MySQL is the world's most popular database management system. It easily runs with almost every programming language available.

PHP-

PHP stands for Hypertext Preprocessor. It is a server-side scripting³ language for making dynamic and interactive Web pages. It powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn.

Perl-

Perl is a programming language used for network programming, system administration, etc. It is less popular for web development purposes.

PhpMyAdmin-

PhpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. PhpMyAdmin supports a wide range of operations on MySQL.

Installing XAMPP on Windows

Downloading XAMPP

Download XAMPP from “<https://www.apachefriends.org/>”. It is available in three file formats:

- .EXE – We will use this one to download and install.
- .7z – 7zip file. This is the purist of all, but has high complication in installing.
- .ZIP – zip file. This is compressed format and .ZIP files is considerably more difficult than using .EXE

Installing XAMPP-

Follow these steps for installing XAMPP on windows:

Step 1: Disable your antivirus as it can cause some XAMPP components to behave irregularly.

Step 2: Start the installation process by double-clicking on the XAMPP installer. Select your language from the display screen.

³ Server-side scripting language means that the program written in PHP language will execute on the Web server, not within the browser on your local/client machine. When you want to execute PHP program/code in a client machine you must install a web server in your client/local machine. Such as- WAMP Server (can freely download from- <http://www.wampserver.com/en/>)

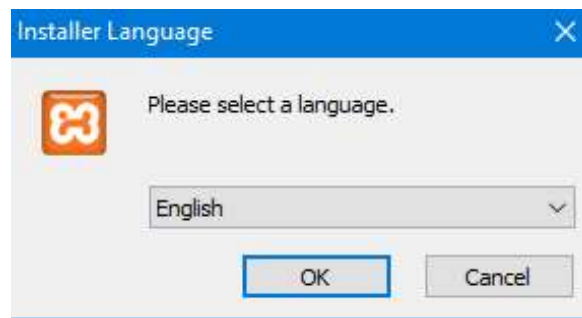


Fig 5.1 (Step 2 in XAMPP installation)

Step 3: Click 'Next' after the splash screen.

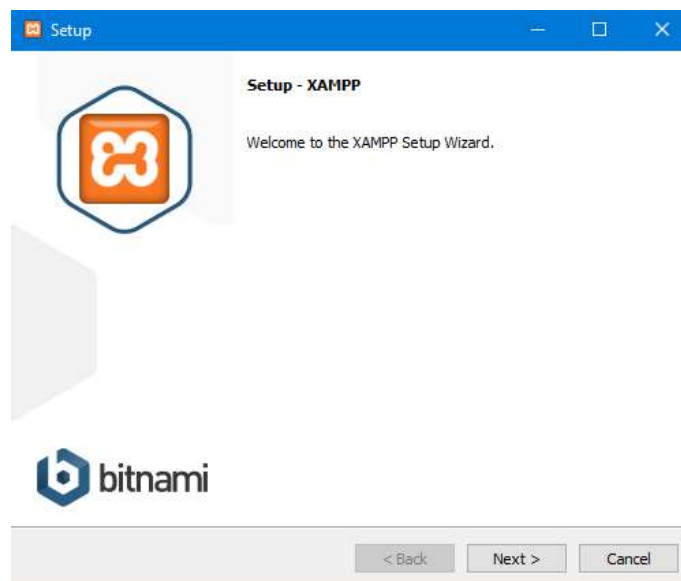


Fig 5.2 (Step 3 in XAMPP installation)

Step 4: Here, you can select the components you want to install. Choose the default selection and click 'Next'.

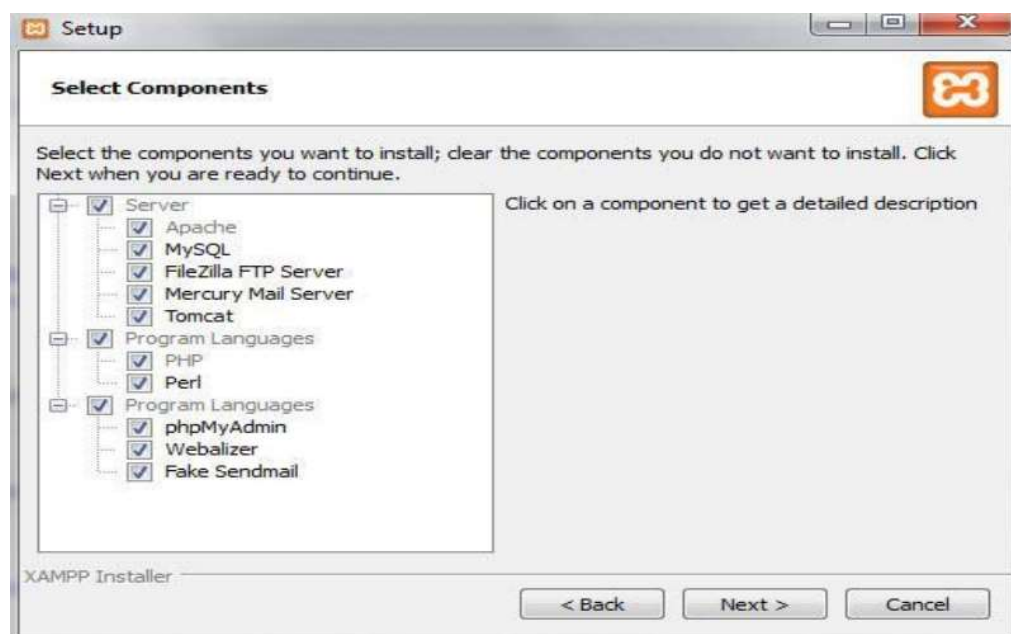
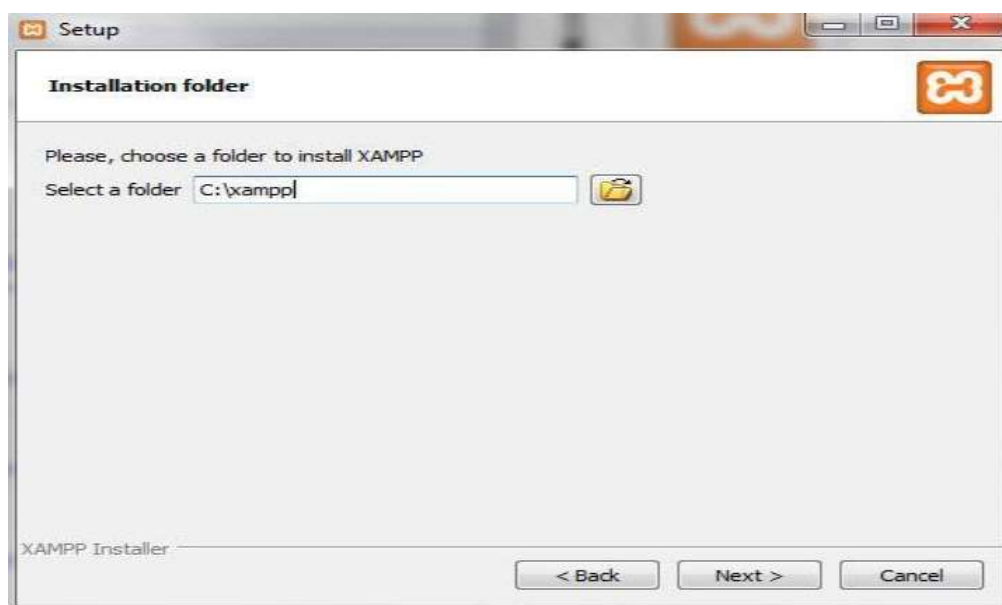


Fig 5.3 (Step 4 in XAMPP installation)

Step 5: Choose the folder you want to install XAMPP in. This folder will hold all your web application files, so make sure to select a drive that has sufficient space.

*Fig 5.4 (Step 5 in XAMPP installation)*

Step 6: The next screen is a promo for BitNami – “an app store for server software”. For now, deselect the ‘Learn more about BitNami for XAMPP’ checkbox.

*Fig 5.5 (Step 6 in XAMPP installation)*

Step 7: Setup is now ready to install XAMPP. Click next and wait for the installer to install the selected components. This may take a few minutes. You may be asked to approve Firewall access to certain gears (such as Apache) during the installation process.

Step 8: Installation is complete! Select the ‘Do you want to start the Control Panel now?’ checkbox to open the XAMPP control panel.



Fig 5.6 (Step 8 in XAMPP installation)

What is XAMPP Control Panel?

The XAMPP control panel gives you total control over XAMPP components. You can use the control panel to start/stop different modules, open Windows explorer and see all operations running in the background. Here is a general idea of the Control Panel. For now, you only need to know how to start and stop an Apache server.

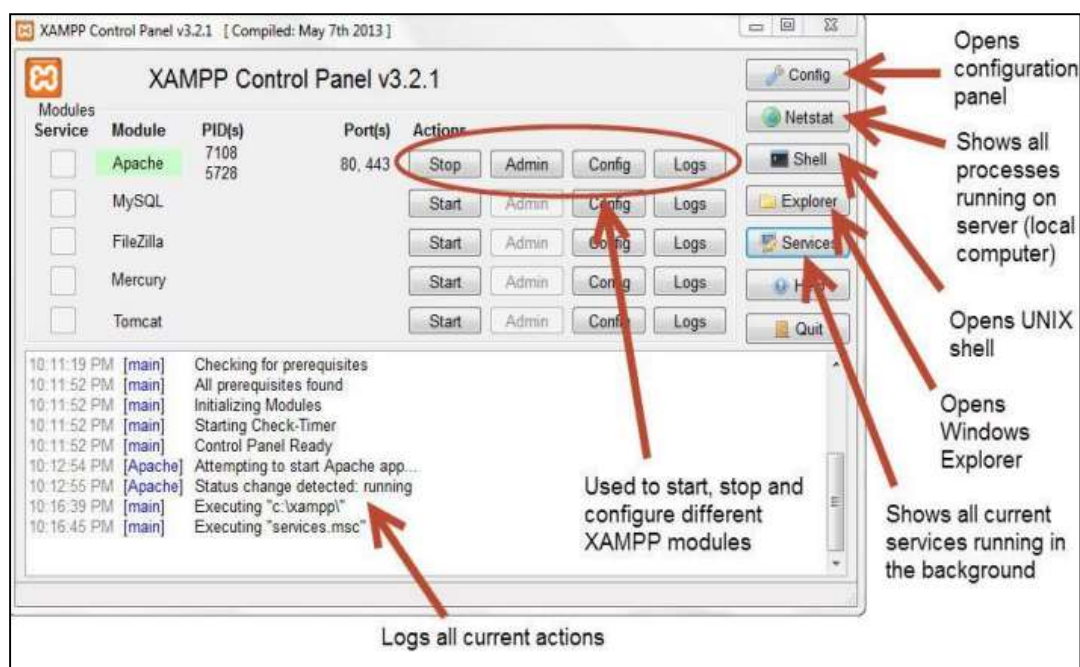


Fig 5.7 (XAMPP Control Panel)

Checking XAMPP Installation-

Follow these steps to test your XAMPP installation by starting the Apache web server.

Step 1: In the XAMPP control panel, click on '**Start**' under '**Actions**' for the Apache module. This instructs XAMPP to start the Apache webserver.

Step 2: Open your web browser and type in: ***http://localhost***

Step 4: You should see the following screen. This means you've successfully installed XAMPP on your computer.

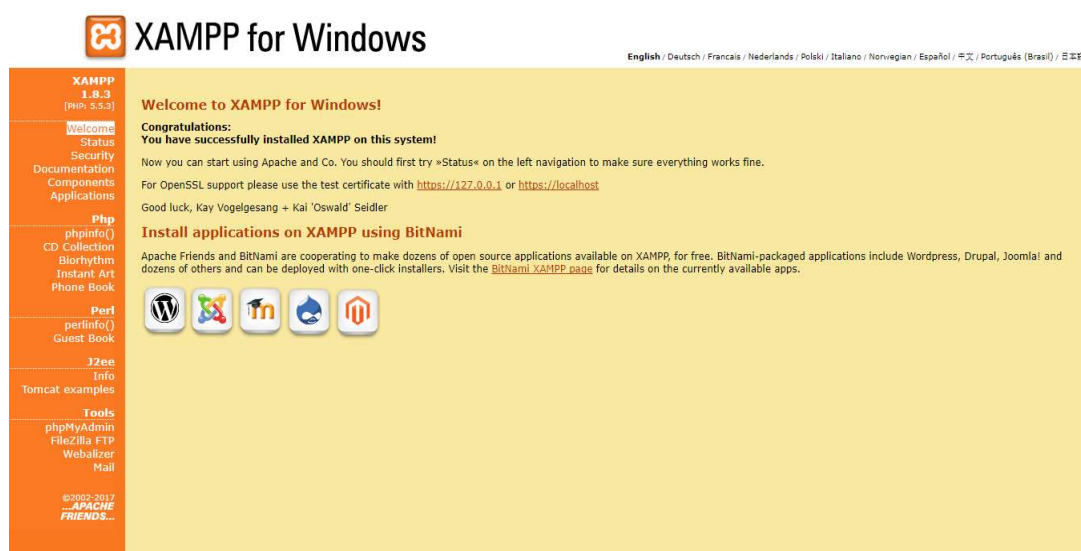


Fig 5.8 (Home page of XAMPP in a browser)

Congratulations! This screen shows that you have successfully installed XAMPP. You can now use your local Apache webserver to create and test AJAX, PHP-MYSQL applications.

Running files from section 4.8 & 4.9

Create a folder named testXML in c: /xampp/htdocs/ and put those files created in section 4.8 & 4.9 there. Now for running these files, open browser and type URL as:

“http://localhost/testXML/ajaxPractice.html”

Run files separately for loading text and xml data. This will call the AJAX function within html files and load data from TEXT/XML files.

5.5 INTRODUCTION TO PHP

PHP stands for Hypertext Preprocessor, is a widely-used open source general-purpose scripting language. PHP is server-side scripting language, which is especially suited for web development. PHP scripts can run on many different operating systems and Web servers.

PHP is a powerful scripting language for making dynamic and interactive Web pages. Another feature of PHP script is HTML embedded, means that PHP statements and commands are actually embedded in your HTML documents and the file must be saved with valid extension (PHP file extension is: “filename.php”). PHP files can contain text, HTML, CSS, JavaScript, and PHP code too. When the Web server sees

the PHP statements in the Web page, the server executes the statements and sends the resulting output along with the rest of the HTML.

PHP's popularity continues to enhance as businesses and individuals the world over embrace it as an alternative to Microsoft's ASP (Active Server Pages) language and recognize that PHP's benefits most certainly outweigh the costs.

For the purposes of working through this chapter, we assume that you have installed XAMPP (see previous section) on your computer. PHP, Apache & MySQL is installed on your computer if you have XAMPP.

What is the use of PHP File?

- 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 performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- It can add, delete, and modify data in your database.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.

How PHP Works?

PHP is a server-side language, which means that the code you write in PHP resides on a server that serves Web pages to Web browsers.

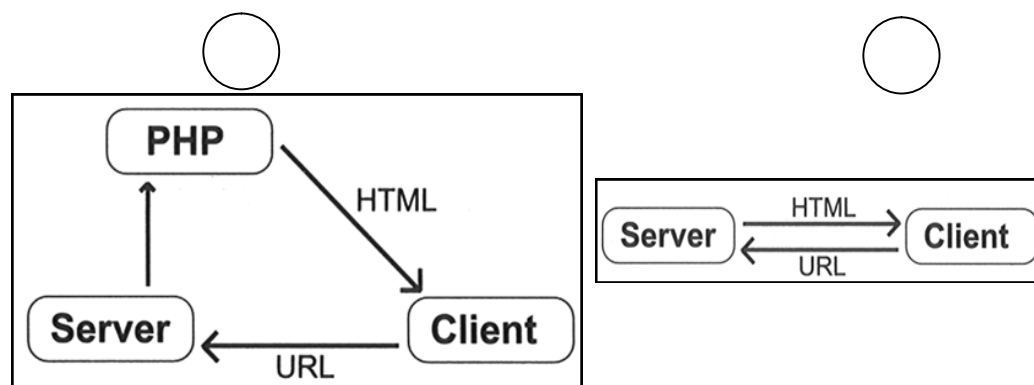


Fig 5.9 - Client Server model of PHP (a) and HTML Model

The central difference between using PHP and using straight HTML is that everything PHP does, it does on the server and it then sends the appropriate information to the browser (fig 5.9). This chapter covers how to use PHP to send the right data to the browser.

Getting started with PHP

Syntax to start and close PHP in a web page-

```
<?php
```

```
....
```


....

PHP statements here

?>

Start using PHP in your web page-

(I) HTML Embedded

This program is embedded with HTML document and prints output as “Hello World”. In this example, you see that your PHP code sits embedded in your HTML. Every time the PHP interpreter reaches a PHP open tag "<?php" , it runs the enclosed code up to the delimiting "?>" marker.

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

(II) Non - HTML Embedded

```
<?php
  echo "Hello, World";
?>
```

Note: You can create a PHP file directly, without HTML embedding on it and save it with the extension .php. You can call this file in any web page whenever required with the help of PHP library function ‘require()’ or ‘include()’. We will describe the same in details in later topic.

Benefits of PHP-

- PHP is open source software.
- PHP costs nothing, it is free to download and use.
- PHP is a server-side scripting language and is used for websites and the web applications.
- PHP supports a wide range of databases.
- PHP runs on various platforms like Linux, Windows, Unix etc.
- PHP is a Object Oriented language.

PHP can do the following tasks- (Some of the tasks are listed below)

- PHP can generate dynamic page content.
- PHP can create, open, read, write, delete, and close files on the web server.
- PHP can collect data from forms (as user interface).
- PHP can add, delete and update data in your database.

- PHP can be used to control user-access.
- PHP can encrypt the data and many more.

Basic Syntax

In order to create your first PHP page, you will begin exactly as you would if you were creating an HTML file from scratch.

There are two main differences between a standard HTML document and a PHP document.

1. PHP scripts must be saved with the .php extension (e.g., index.php).
2. You wrap your PHP code with the `<?PHP` and `?>` tags to indicate what is PHP as opposed to what is HTML.

To create and run a new PHP script on your computer:

1. Open a text editor such as notepad, notepad++, or any text editor whichever you wish to use.
2. Choose File > New to create a new, blank document and type the following code fragment

```
<HTML>
<HEAD>
    <TITLE>First PHP Script</TITLE>
</HEAD>
<BODY>
    <?PHP
        echo ("HELLO WORLD");
        //echo is used to display text in the browser
    ?>
</BODY>
</HTML>
```

3. Now Choose File > Save As
4. Choose the location "c:/XAMPP/htdocs/".
(if your server [in our example we used XAMPP] installed in C: [C Drive] otherwise use the appropriate drive where server is installed.)
5. Save the script as "first.php"

Now that you have created your first PHP script, it's time to check the PHP code is working or not.

- Open XAMPP control panel, click on 'Start' under 'Actions' for the Apache module. This instructs XAMPP to start the Apache webserver.
- Now open your web browser and change URL to- "http://localhost/first.php"

If all things goes well then you see the output HELLO WORLD in the browser. Check view source option in the browser. It will display the HTML code written in

“first.php” without any php code. It confirms that PHP runs on server, generates output as HTML and send that output to the client.

Comments in PHP

Adding comments to your work helps remind you of what your idea at the time of coding, which are not always so clear months later. A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is checking the code.

PHP supports three methods of adding comments // , #, /* */. Let see an example to get clear idea about comments.

```
<html>
  <body>
    <?php
      // This is single-line comment
      # This is also a single-line comment
      /*
        This is a multiple-lines comment block
        that spans over multiple lines
      */
    ?>
  </body>
</html>
```

5.6 BASIC PROGRAMMING COSTRUCTS IN PHP

Variables in PHP

A variable is thought of as a container for data. Once data has been stored in a variable, that data/variable can be altered, printed to the Web browser, saved to a database, e-mailed etc. But, variables in PHP are also temporary: they only exist- that is, they only have a value- while they are used within a script. Once you are in a new page, those variables cease to exist, unless you pass them along to the new page, which I'll discuss in the further section.

Rules for PHP variables:

- 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 (\$name and \$NAME are two different variables).

Note: PHP is a Loosely Typed Language-

In PHP, a variable does not need to be declared which data type the variable will store; before assigning a value to it. PHP will automatically convert the variable to the correct data type, depending on its value. This feature makes PHP as Loosely Typed Language. In a strongly typed programming language, you have to declare (define) the type and name of the variable before using it.

Types of Variables

Variables can store data of different types, and different data types can do different things. As-

PHP supports the following data types:

String

Strings are sequences of characters, where every character is the same as a byte. A string can hold letters, numbers, and special characters and it can be as large as up to 2GB (2147483647 bytes maximum).

For example-

```
<?php
$a = 'Hello world!';
echo $a;
?>
```

Example 1-

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $value1=34;
      $value2=45;
      $fee=3442.56;
      $myname="Abc Xyz";
    ?>
  </body>
</html>
```

Note- In the above code \$value1, \$value2, \$fee and \$myname are the variable names.

Integer

Integers are whole numbers, without a decimal point (... , -2, -1, 0, 1, 2, ...). Integers can be specified in decimal (base 10), hexadecimal (base 16 - prefixed with 0x) or octal (base 8 - prefixed with 0) notation, optionally preceded by a sign (- or +).

```
<?php
$a = 123; // decimal number
echo $a;
$b = -123; // a negative number
```

```
echo $b;
?>
```

Floating point numbers or Double

Floating point numbers (also known as "floats", "doubles", or "real numbers") are decimal or fractional numbers, like demonstrated in the example below.

```
<?php
$a = 12.34;
echo $a;
$b = 10.2e3;
echo $b;
?>
```

Boolean

Booleans are like a switch it has only two possible values either 1 (true) or 0 (false).

```
<?php
// Assign the value TRUE to a variable
$showBool= true;
echo $showBool;
?>
```

Array

An array is a variable that can hold more than one similar type of value at a time. It is useful to aggregate a series of related items together.

```
<?php
$cars = array("maruti","Toyota","Datsun");
echo ($cars);
?>
```

Object

An object is a data type that not only allows storing data but also information on, how to process that data. An object is a specific instance of a class which serve as templates for objects. Objects are created based on this template via the new keyword.

First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods:

```
<?php
// Class definition
class greeting{
    // properties
    public $str = "Hello World!";
    // methods
    function show_greeting(){
```

```

        return $this->str;
    }
}
// Create object from class
$message = new greeting();
echo $message->show_greeting();
?>
NULL

```

The special NULL value is used to represent empty variables in PHP. A variable of type NULL is a variable without any data. NULL is the only possible value of type null.

```

<?php
    $x = "Hello world!";
    $x = null;
    echo ($x);
?>

```

Resource

A resource is a special variable, holding a reference to an external resource. Resource variables typically hold special handlers to opened files and database connections. We will not talk about the resource type here, since it is an advanced topic.

Constant

A constant is a name or an identifier for a fixed value. Constant are like variables except that one they are defined, they cannot be undefined or changed.

Constants are very useful for storing data that doesn't change while the script is running. As-

```

<?php
// Defining constant
define("SITE_URL", "http://google.com");
// Using constant
echo 'Thank you for visiting - ' . SITE_URL;
?>

```

The output of the above code will be:

Thank you for visiting - http://google.com

Operators in PHP

Operators are the mathematical, string, comparison, and logical commands such as plus, minus, times, and divide, which in PHP looks a lot like plain arithmetic;

For instance, the following statement output is: 10

```
echo 6 + 4;
```

Before moving on to learn what PHP can do for you, take a moment to learn about the various operators provided by php. As-

Arithmetic Operator			
Operator	Description	Example	Result
+	Addition	If x=2 x+2	4
-	Subtraction	If x=2 5-x	3
*	Multiplication	If x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%2	1 2
++	Increment	If x=5 x++	X is 6
--	Decrement	If y=12 y--	Y is 11
Assignment Operator			
Operator	Description	Example	
=	x=y	x=y	
+=	x+=y	x=x+y	
-=	x-=y	x=x-y	
=	x=y	x=x*y	
/=	x/=y	x=x/y	
.=	x.=y	x=x.y	
%=	x%=y	x=x%y	
Comparison Operator			
Operator	Description	Example	
==	is equal to	5==8 returns false	
!=	is not equal	5!=8 returns true	
<>	is not equal	5<>8 returns true	
>	is greater than	5>8 returns false	
<	is less than	5<8 returns true	
>=	is greater than or equal to	5>=8 returns false	
<=	is less than or equal to	5<=8 returns true	
Logical Operator			
Operator	Description	Example	
&&	and	If x=6, y=3 (x < 10 && y > 1) returns true	
	or	If x=6, y=3 (x==5 y==5) returns false	

!	not	If x=6, y=3 !(x==y) returns true
---	-----	-------------------------------------

5.7 POINTS TO REMEMBER

- PHP script always runs from the server, therefore, Apache http server (included in XAMPP, WAMP & LAMP) has to be installed before working on PHP.

5.8 GLOSSARY

- **XAMPP:** XAMPP is a completely free, easy to install Apache distribution containing Apache, MySQL, PHP, and Perl.
- **Apache:** The Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software.
- **PHP:** PHP is a server-side scripting language designed primarily for web development.
- **MySQL:** MySQL is an open-source relational database management system.
- **PhpMyAdmin:** phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. PhpMyAdmin supports a wide range of operations on MySQL and MariaDB.
- **PERL:** Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.
- **XSS:** Cross-site scripting is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users.
- **SQL:** SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.
- **Web Server:** A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web.
- **Client:** A client is a piece of computer hardware or software that accesses a service made available by a server.

5.9 CHECK YOUR PROGRESS

Part- A (Multiple choice type questions)

(1) PHP files have a default file extension of?

A. html

B. php

- C. js D. phh
- (2) A PHP script should start with ____ and end with ____
- A. < php > B. < ? php >
C. < ? ? > D. < ?php ?>
- (3) Which of the looping statements is/are supported by PHP?
- i) for loop
ii) while loop
iii) do-while loop
iv) foreach loop
- A. (i) and (ii) B. (i), (ii) and (iii)
C. All of the mentioned D. None
- (4) Which of the following php statement/statements will store 111 in variable num?
- A. int \$num = 111; B. int mum = 111;
C. \$num = 111; D. 111 = \$num;
- (5) What is the correct variable definition?
- A. \$3hello B. \$3_hello
C. 3\$ hello D. \$ hello

Part- B (Descriptive type questions)

- (1) Define PHP.
- (2) How can you display text with a PHP script?
- (3) How do you define a constant in PHP?
- (4) What is the functionality of Strtolower() function?
- (5) What are the real web applications of AJAX currently running in the market?
- (6) What is phpmyAdmin?
- (7) What is Xampp?
- (8) What are the components used within XAMPP?
- (9) What is the use of "echo" in php?
- (10) How to declare an array in php?
- (11) What is use of count() function in php ?
- (12) What types of loops exist in php?
- (13) How to redirect a page in php?
- (14) What is mean by an associative array?
- (15) What is sql injection?
- (16) How variables are declared in PHP (syntax only).

- (17) How function is declared in php?
- (18) How can we count the number of words in a String using predefined PHP function?
- (19) What is the use of Ucwords()?
- (20) Write a program that displays a different message based on the time of day. For example, if it is in the morning, have the site display “Good Morning!”.

Answers- (Part- A- Multiple choice type questions)				
1. B	2. D	3. C	4. C	5. D

5.10 BIBLIOGRAPHY/REFERENCES

- Robin Nixon, 2012, “Learning PHP, MySQL, JavaScript, and CSS”, published by O’Reilly Media.
- Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, 2005, “Beginning PHP5, Apache, and MySQL® Web Development”, published by Wiley Publishing Inc.
- <http://easyphp.org>
- <http://free-php.net>
- <http://hotscripts.com/category/php/>

5.11 SUGGESTED READINGS

- Robin Nixon, 2012, “Learning PHP, MySQL, JavaScript, and CSS”, published by O’Reilly Media. “Head First Ajax”, Rebecca M. Riordan, ISBN-13: 978-0596515782
- David Powers, “PHP Solutions: Dynamic Web Design Made Easy 2nd ed. Edition” ISBN-13: 978-1430232490
- Larry Ullman, “PHP for the Web: Visual QuickStart Guide (4th Edition)”, ISBN-13: 978-8131766729.
- Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, 2005, “Beginning PHP5, Apache, and MySQL® Web Development”, published by Wiley Publishing Inc.

UNIT 6

INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

6.1. INTRODUCTION

6.2. OBJECTIVE

6.3. HISTORY OF MOBILE COMMUNICATION DEVELOPMENT

6.4. WHAT IS SMARTPHONE AND MOBILE APPLICATION?

6.5. WHY WE NEED MOBILE APPLICATIONS?

6.6. PROGRAMMING LANGUAGES USED IN MOBILE APPLICATIONS

6.7. OPERATING SYSTEM USED IN MOBILE PHONES

6.8. MOBILE APPLICATION DEVELOPMENT PLATFORM

6.9. CREATING HELLO WORLD APP USING PHONEGAP

6.10. POINTS TO REMEMBER

6.11. GLOSSARY

6.12. CHECK YOUR PROGRESS

6.13. BIBLIOGRAPHY/REFERENCES

6.14. SUGGESTED READINGS

6.1 INTRODUCTION

In the past couple of years, we have witnessed “tremendous growth of mobile users” all over the world as the entry of “smart phones” in the market at affordable prices has triggered their usage. We have experienced a major shift in the way we access the internet today with mobiles becoming the primary access point for internet usage. In today’s fast paced world, “phones are not just used for calling, playing games etc.”, but with smartphones we can schedule our complete day, as- “check emails, make conference calls, connect using social network and perform a host of other activities”.

The growth of mobile phone market has generated a huge demand for various “mobile applications”. Numerous mobile phone applications are available that simplify various tasks for the users due to which we saw an accelerated growth of software/application development for mobile devices. Mobile application development is the course of action by which application software is designed and developed for hand-held devices like mobile phones, personal digital assistants, etc.

Earlier mobile developers faced many difficulties while writing applications as they had to build better, unique, competing and “hybrid applications” which would incorporate command tasks like messaging, contact list and calling in a user-friendly manner. The launch of “Android smartphones” in the market brought a revolution in Mobile Application Development. If you have some basic knowledge about scripting and coding you can start building your own applications. Mobile Application Development was never as easy as it is now.

Every mobile phone vendors/proprietary platform like “Android, Apple iPhone OS, RIM Blackberry OS, Bada OS, Symbian OS” provide their own “SDK (Software Development Kit)” to the developer. Developers can create applications using this kit. Developers are also provided a space/place/market where they can publish their creations to the world.

6.2 OBJECTIVES

After successful completion of this unit you will be able to-

- **Explain Smartphone and History of mobile development**
- **Define programming languages used in mobile application development**
- **Explain the different types of mobile Operating Systems.**
- Understand briefly the phonegap mobile development platform

6.3 HISTORY OF MOBILE COMMUNICATION DEVELOPMENT

The “AT&T” was one of the first to commercialize mobile telecommunication in 1947.

The service known simply as “Mobile Telephone Service” (MTS) spread to more than a hundred towns and highway paths by the end of the year. The service relied on an operator to connect both incoming and outgoing calls.

The telephones used were not particularly portable and used a “half-duplex” “press to speak” system where the caller would



have to release the button to hear the other person. That very same year two “Bell Labs engineers” proposed the foundations for the modern “cellular network”. At the time the plans were daring, and it took until the 1960s for the plans to be implemented and even longer to come to market.

MTS was used in North America until the 1980s, despite AT&T’s introduction of the aptly-named “Improves Mobile Telephone Service” (IMTS) in 1965. The new service introduced user dialing, removed the need for operator forwarding and used additional radio channels which increased the number of possible subscribers and calls, as well as area coverage. IMTS was still mobile telephony in its infancy however, and was limited to 40,000 subscribers nationwide. In New York City, 2,000 customers shared 12 radio channels which on average took 30 minutes to place a call.

“Radio Common Carriers (RCCs)” were another solution designed to compete with AT&T’s “MTS” and “IMTS” systems. Not only were the units huge (see above) but standards varied widely. Some phones were half-duplex “push to talk”, some were “full-duplex” much like a “wired telephone”. Some lucky customers even carried around briefcase-sized full duplex devices, though RCC units were mainly limited to cars.

In 1960 the world’s first fully “automated mobile telephone” was introduced in Sweden. The system allowed for automated connection from a rotary handset (that’s the circular dialing knob to me and you) mounted within a car, but required an operator to forward calls. The system was known as “Mobile Telephone system A (MTA)” and was replaced by “MTB” two years later.

In this interim period there were several other solutions including the arrival of “Motorola” on the scene in 1959 and “Bulgarian and Russian” (then USSR) solutions sprouted up too. It wasn’t until 1971 when the “ARP network” was introduced to Finland that the world’s first successful commercial network was launched. Dr. Martin Cooper, a Motorola researcher and executive made the first phone call from a “handheld mobile phone” on April 3, 1973. This ushered in a new dawn of communication.

31 July 1995 and on the same day first mobile phone was launched in India. The then “Union Telecom Minister Sukh Ram” and the then Chief Minister of West Bengal Jyoti Basu spoke to each other using hand held mobile phones on July 31, 1995.

6.4 WHAT IS SMARTPHONE AND MOBILE APPLICATION?

Smartphone-

A smartphone is a “handheld personal computer” with a “mobile operating system” and an integrated “mobile broadband cellular network” connection for “voice, SMS, and Internet data” communication; most if not all smartphones also support ‘Wi-Fi’. Smartphones are typically pocket-sized, as opposed to “tablet” computers, which are much larger. They are able to run a variety of software components, known as “apps”. Most basic apps (e.g. event calendar, camera, web browser) come pre-installed with the

system, while others are available for download from official sources like the “Google Play Store” or “Apple App Store”. Apps can receive bug fixes and gain additional functionality through software updates; similarly, operating systems are able to update. Modern smartphones have a “touchscreen” color display with a graphical user interface that covers the “front surface” and enables the user to use a “virtual keyboard” to type and press on screen icons to activate "app" features. “Mobile payment” is now a common theme amongst most smartphones.

A smartphone is a “cellular telephone” with an integrated computer and other features not originally associated with telephones, such as an “operating system”, Web browsing and the ability to run software applications. Smartphone is based on different type of Operating System, Like “Android, iOS, Windows”.

Mobile Application-

A “mobile app” is a computer program designed to run on a mobile device such as a phone/tablet or watch. Mobile applications often stand in contrast to desktop applications which run on “desktop computers”, and with web applications which run in “mobile web browsers” rather than directly on the mobile device.

In other words A “mobile application”, most commonly referred to as an “app”, is a type of “application software” designed to run on a mobile device, such as a “smartphone” or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs.

6.5 WHY WE NEED MOBILE APPLICATIONS?

There are following reasons to use mobile applications. As-

- **Anywhere, anytime-** By having a “mobile app”, businesses get placed in the consumer’s pocket and are with them all the time. Adding social share buttons inside the app makes it convenient for people to share with their friends any interaction they have with a brand whenever they want. Being with the customer 24/7 is an opportunity businesses shouldn’t pass up.
- **Your Mobile App is your “Direct Marketing Channel”-** Mobile Apps serve many functions: they can provide “general info, prices, booking forms, search features, user accounts, messengers, news feeds, and much more”.
- **Helps to build your business brand & recognition-** With “mobile app” you can do what you want with it; you can make it stylish, hip, functional, shocking, or informative. But what you really want to do is create an app that has features your customers will love, while at the same time is well branded and beautifully designed. The more often you can get customers involved with your app, the sooner they will be inclined to buy your product and/or service. In advertising this is called the “effective frequency”: as a rule of thumb, hearing and/or seeing your brand approximately 20 times is what will get you truly noticed.

- **Helps to improves Customer Engagement-** No matter whether you are selling flowers or spa services, your customers need a way to reach you. Having a messaging or help desk feature within your app can really make a difference in the way you communicate with your customers. Think about it.
- **Builds Customer Loyalty- With all the noise out there-** “roadside banners, billboards, flashing signs, newspaper ads, flyers, coupons, websites, website banners, Facebook ads, and email marketing” — we slowly lose our impact on customers because of the immense amount of advertising surrounding us all. It’s time to go back to making a true and sincere connection with your customers, and making them a loyal lover of your product and/or service. Not saying a “mobile app” is going to save your business, but it can be a way of staying closer to your customers, and being just a “fingertip” away at all times.
- **Helps to enlarge your customer base for your business-** Apps can help you bring together multiple marketing tactics and make them all work more effectively in your favor. For example, a loyalty program is a time-proven marketing tool.
- **Offers New Stream of Conversions-** If you’re looking to increase conversions, mobile apps can be a great medium to push users down the conversion funnel. Mobile apps can be used to acquire both top-of-the-funnel, and bottom-of-the-funnel users. For instance, utility apps can bring-in Top funnel users, which can be later nurtured into Bottom funnel leads. On the other hand, apps like “eCommerce” already have BOTF users, who have a higher possibility of converting.
- **Mobile App Helps Boost Revenue-** Consumers are used to buying your products or services through your “mobile website”. “Building purchasing capabilities” into your mobile app ensures they can buy what they need from you at any time. Your app can include a product catalog, ordering and reservation capabilities — so if customers are away from their PC, they can order what they need and pay for it from their mobile devices.
- **Enhances your digital marketing experience-** Incorporate data gathering into your app, and you can use it to generate metrics to empower all your marketing efforts. These metrics can provide you with important information on how your customers interact with your business such as visits, checkouts, purchases, searches and more. Those numbers can help make other marketing tactics, such as email marketing, more effective and fruitful.
- **At affordable costs-** Unless you’re a small business creates and sells apps to others, chances are good that as a small business owner you don’t have the technical chops to create an app on your own. Of course, there’s no dearth of vendors and individuals who will create an app for you. Once upon a time, those people charged a lot of money to create custom apps — and many still do.

6.6 PROGRAMMING LANGUAGES USED IN MOBILE APPLICATIONS DEVELOPMENT

Some of the programming languages used in mobile application development are briefly defined. As-

HTML5

“HTML5” is the ideal programming language if you are looking to build a “Web-fronted app” for “mobile devices”. Although it makes various data types simple to insert, accounts for different screen sizes, rationalizes input parameters, and even levels the browser playing field, the problem with HTML5 is that it is still a proposed standard. Currently supported in a lot of different ways by a lot of different browsers, HTML5, from the cost-efficiency point of view, has the advantage of building on the current version of HTML- making the learning curve much shallower than that for a completely new language.

Objective-C

The primary programming language for “iOS apps”, “Objective-C” was chosen by “Apple to build apps” that are robust and scalable. Being a C-language superset, it does have a number of functions that precisely deal with graphics, I/O, and display functions. Moreover, as part of the Apple development framework, Objective-C is fully integrated into all iOS and MacOS frameworks. However, it’s now slowly being replaced in the Apple ecosystem by a more powerful language called Swift.

Swift

Swift is the latest programming language to foray into the Apple ecosystem, mainly considering its prevalence in writing code for Apple’s latest APIs, Cocoa and Cocoa Touch. Even though it is a language written to work along with Objective-C, the Cupertino company is making it obvious for iOS developers to turn to Swift for complete programming. Designed to eliminate the likelihood of many of the security vulnerabilities possible with Objective-C, it’s time for mobile app developers to turn to Swift, as many businesses are looking to hire Swift developers with expertise in developing cutting-edge mobile apps using this language.

C++

This is the most appropriate and robust programming language when it comes to building mobile apps for Android and Windows- and, mainly for low-level programming, it’s still the go-to language on platforms for mobile app developers. As a powerful programming language, C++ allows mobile apps to be developed for practically every purpose on every platform that exists. It might not be super chic or trendy, but it dominated the programming world even before the smartphone revolution.

C#

The most coveted programming language for Windows Phone app development, C# does the trick for Microsoft that Objective-C does for Apple. Although a Windows Phone

platform couldn't emerge as the game-changer in the mobile application development industry, for loyal Microsoft users, C# makes the perfect programming language to build the robust Windows Phone apps.

Java

Java programming language is one of the most preferred languages when it comes to Android app development. An object-oriented programming language developed at Sun Microsystems (now owned by Oracle), Java can be run in two different ways: either in a browser window, or in a virtual machine that can do without a browser.

6.7 OPERATING SYSTEM USED IN MOBILE PHONES

A mobile operating system (or mobile OS) is an operating system for phones, tablets, smart watches, or other mobile devices. While computers such as typical laptops are 'mobile', the operating systems usually used on them are not considered mobile ones, as they were originally designed for desktop computers that historically did not have or need specific mobile features. This distinction is becoming blurred in some newer operating systems that are hybrids made for both uses.

Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use; usually including, and most of the following considered essential in modern mobile systems; a touch screen, cellular, Bluetooth, Wi-Fi Protected Access, Wi-Fi, Global Positioning System (GPS) mobile navigation, video- and single-frame picture cameras, speech recognition, voice recorder, music player, near field communication, and infrared blaster.

List of Some Mobile Operating System-

Android Mobile Operating System –

(Developed by- Google Inc.)

It is the current leading OS in the market of smart phone operating systems for mobile devices. It's launched in Sept. 2008 with the name of 'Astro', then Bender & Cupcake are the updated version came out. But later Google adopted 'Android' as a title for their regular updating new versions coming out. Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice-cream sandwich, Jelly-bean, KitKat, Lollipop, Marshmallow, Nougat and Oreo. Oreo is the current updated versions of Android OS. It is programmed in C, C++, Java & powered by Linux kernel and manufactured by Google Inc. As it's coming from Google family, it has made a great impression in the mobile OS market and it can be clearly seen in the above chart. It's an open source OS that allows developers to develop as they wanted the way they want to do it creatively, rather it's now become much easy to make any Android app by online tools which furthermore don't require deep programming knowledge.

The file extension of Android app is .apk used to install apps on OS – geek people use this file via official free software Samsung Kies or any Android Manager tool to install it directly on the smartphone. Google has said more than 12k companies are

using or testing 'Android for Work' – most popular companies are Samsung, LG, HTC, Sony, Micromax, Lenovo, Motorola and much more. Multitasking, highly customizing capability, various widgets - these are the best features of Android. Including Android tablet; some of the bestselling smartphones based on Android OS are Samsung Galaxy S6, Samsung J7, LG G4, Lenovo K4 Note, Sony Xperia Z3 Compact, Samsung Note 5.

The logo of Android defines the cleanliness and greenery. The logo of Android is designed by Irina Blok a graphic designer.



iOS- (formerly known as- *iPhone Operating System*)

(Developed by- Apple Inc.).

It's one of the special operating systems for mobile devices specially designed and developed for the use of Apple devices like iPhone, iPad & iPod etc. It's programmed in C, C++, Objective-C, Swift & it's been launched in July 2007. The iOS development working by the Apple Inc. - the company does not license the iOS for third party hardware, so this operating system is only available only on their manufactured devices as its core base is provided by the Apple Mac OS X operating system. There could be a debate on the comparison of Android vs iOS operating system.

Although, iOS is not best at everything, but it offers the best package of mobile functions. Still in comparison, iPhone is best in camera features, video recordings with slow motions [up to 60 frames per sec.] and till now it holds best music producing tech specifications. If we neglect its price, then this is one of the best trading phones comes out in the market and that is why it is still giving the fight to Android. The official Apple Store include millions of apps for various functions; and with the new updates of iOS versions, it allows the users to use it for multitasking operations like from playing guitar on mobile then to make payments just by fingerprint. Usually, the file extension used of iOS apps is .ipa file package that you can use to install directly via iTunes software which is specially designed for iPhone, iPad, and other idevices.

Symbian Operating System-

Symbian OS was the most widely-used smartphone operating system in the world until 2010, when it was overtaken by Android. Development of Symbian OS was discontinued in May 2014. Symbian OS began as an operating system called EPOC, which was developed in the 1980s by a company named Psion. In 1998, in a joint venture with telephone manufacturers Nokia, Ericsson, and Motorola, Psion became Symbian, Ltd., and EPOC became Symbian OS.

In 2008, Nokia acquired Symbian, and the majority of Symbian OS's source code was released under an open source license. At the time, it was one of the largest open-source code bases ever released to the public. As of 2014, developers are no longer able to publish new Symbian applications, but existing applications are still available for download.

BlackBerry Operating System-

BlackBerry OS is a proprietary mobile operating system designed specifically for Research In Motion's (RIM) BlackBerry devices. The BlackBerry OS runs on BlackBerry variant phones like the BlackBerry Bold, Curve, and Pearl. The BlackBerry OS is designed for smartphone environments and is best known for its robust support for push Internet email. This push email functionality is carried out through the dedicated BlackBerry Enterprise Server (BES), which has versions for Microsoft Exchange, Lotus Domino and Novell Groupwise.

Other mobile operating systems like Android, Microsoft Windows Mobile and Symbian can run on different brands of mobile phones; the BlackBerry OS can run only on BlackBerry phones. BlackBerry OS is similar to Apple's iOS in this regard. Traditionally, BlackBerry applications are written using Java, particularly the Java Micro Edition (Java ME) platform. However, RIM introduced the BlackBerry Web development platform in 2010, which makes use of the widget software development kit (SDK) to create small standalone Web apps made up of HTML, CSS and JavaScript code.

6.8 MOBILE APPLICATION DEVELOPMENT PLATFORM

There are too many platforms for development of mobile applications; few of them are briefly described for your knowledge. You can explore more about them by your own.

Android-

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Kotlin-

Kotlin is a programming language introduced by JetBrains, the official designer of the most intelligent Java IDE, named IntelliJ IDEA. This is a strongly statically typed language that runs on JVM. In 2017, Google announced Kotlin is an official language

for android development. Kotlin is an open source programming language that combines object-oriented programming and functional features into a unique platform.

PhoneGap-

PhoneGap is a software development framework by Adobe System, which is used to develop mobile applications. To develop apps using PhoneGap, the developer does not require to have knowledge of mobile programming language but only web-development languages like, HTML, CSS, and JScript. PhoneGap produces apps for all popular mobile OS platforms such as iOS, Android, BlackBerry, and Windows Mobile OS etc.

Xamarin-

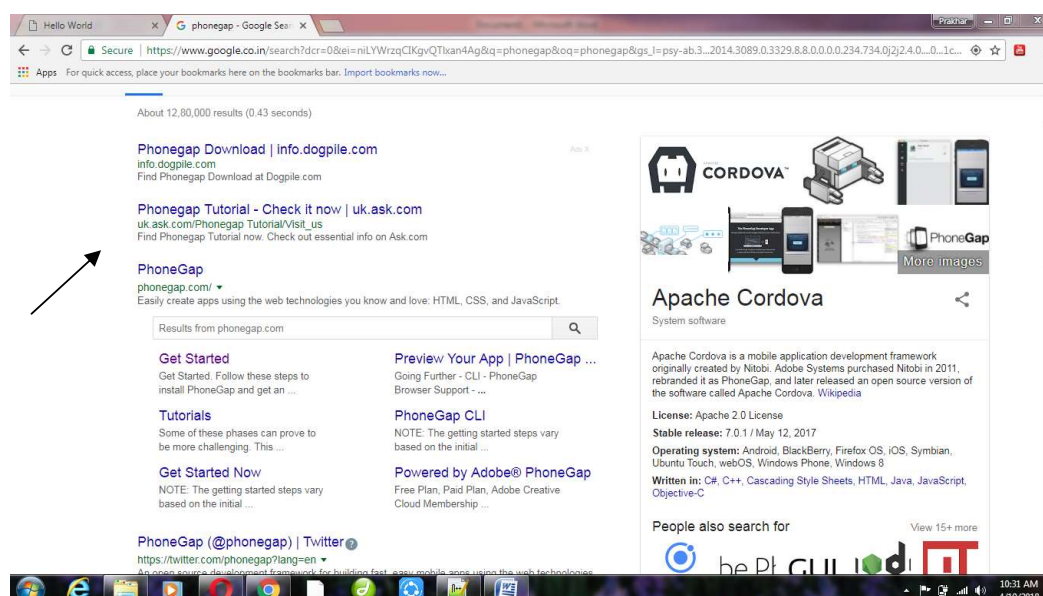
Xamarin is a software company based in San Francisco. It provides commercial software development tools that allow a user to develop applications for Android, iOS and Windows using C# language and the .NET framework. Xamarin is built on the .NET Framework. It allows one to create apps that easily run across multiple platforms.

6.9 CREATING HELLO WORLD APP USING PHONEGAP

How to install PhoneGap?

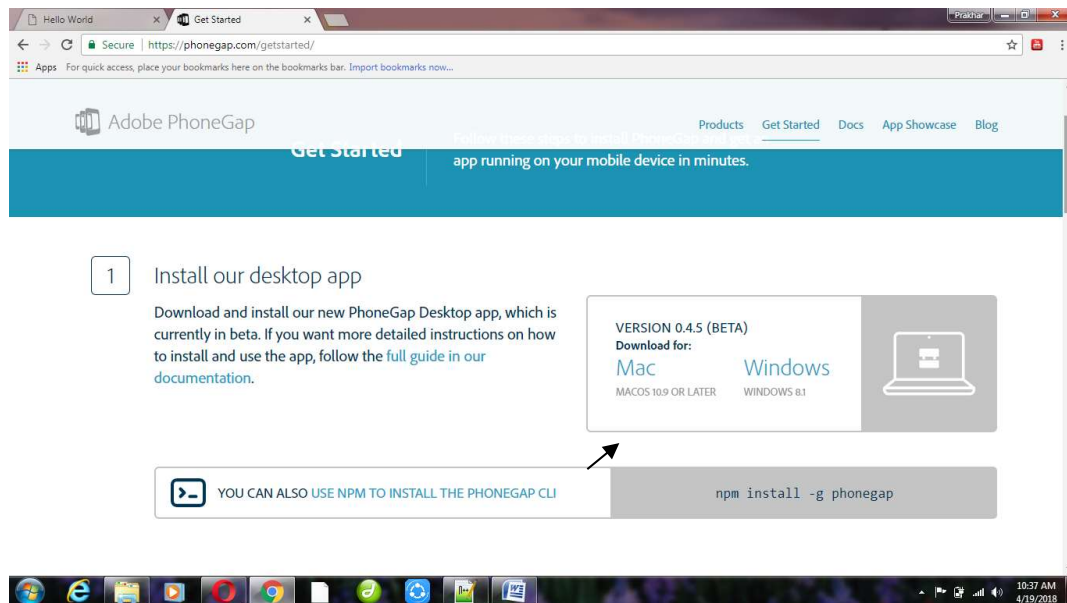
Step 1: Open any browser on your computer.

Step 2: Open Google search and type PhoneGap Desktop (as shown in below figure).

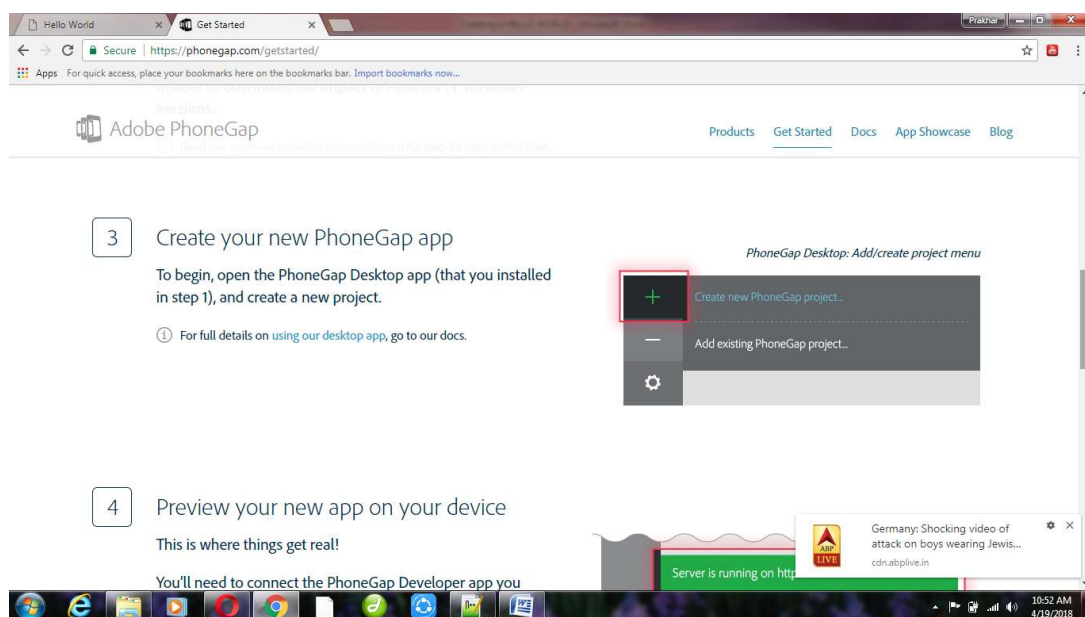


Installation of PhoneGap on Devices-

The below figure shows download link of phonegap for Apple's Macintosh system.



The below figure shows download link of phonegap for Windows's system.



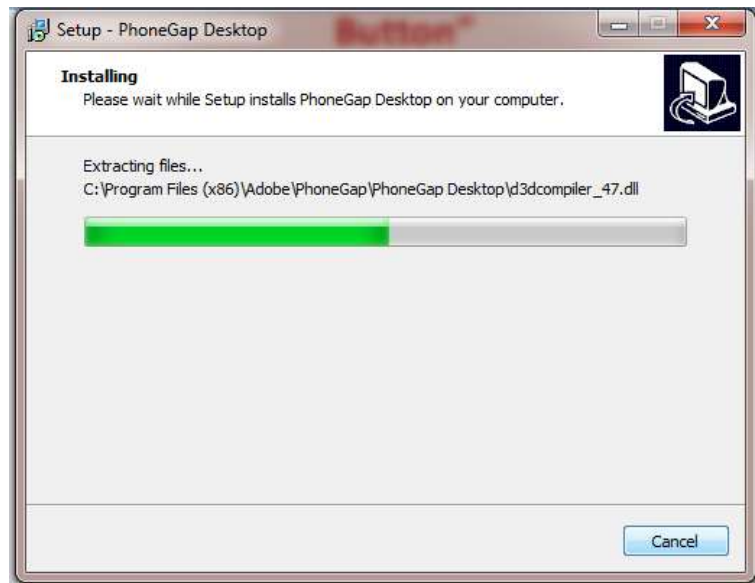
For Mobile Phone (Windows and Android)



After
Download the setup of PhoneGap, Install the PhoneGap setup in Our Computer.
Click on “I accept the agreement”

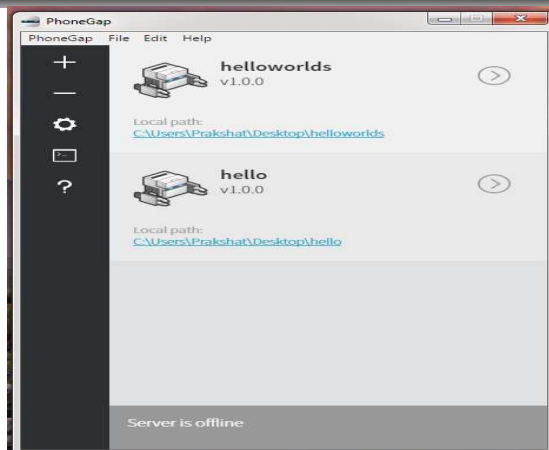
Click on “Next Button” after that Click on “Install Button”

Installation Begins



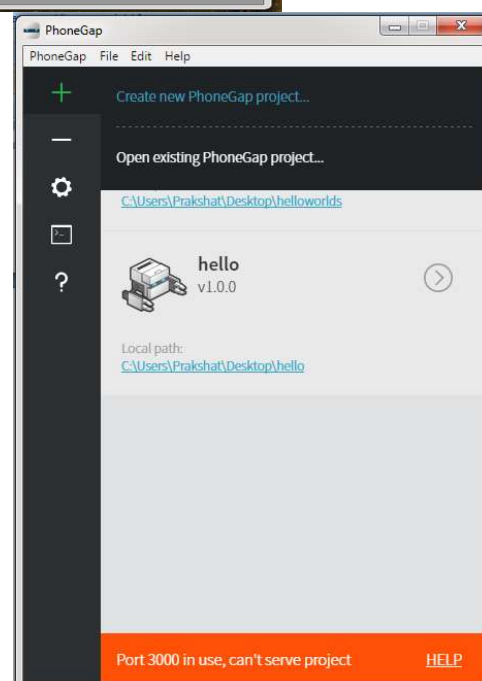
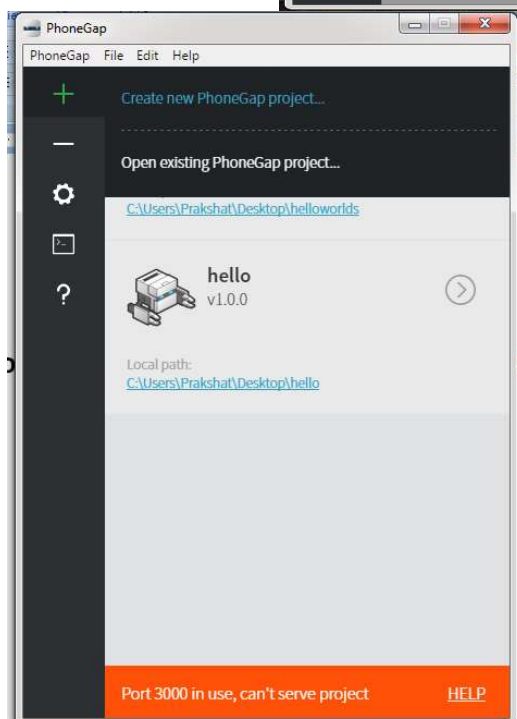
After completion of Process, You will INTERFACE like + button.

Now you will be “Create New Existing Project”

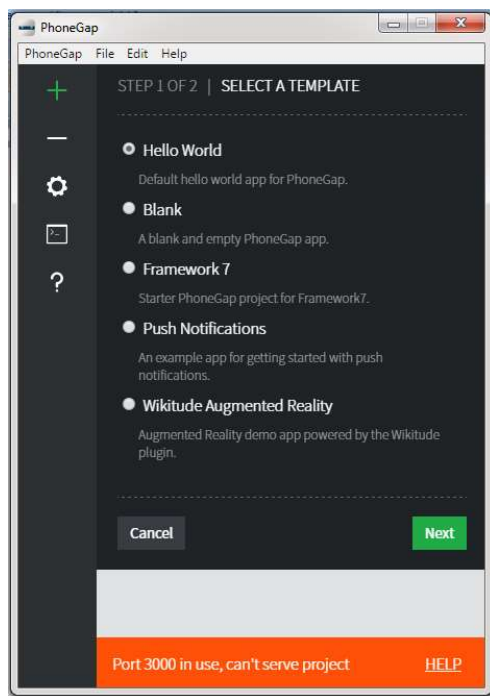


Installation see a USER this, then click on

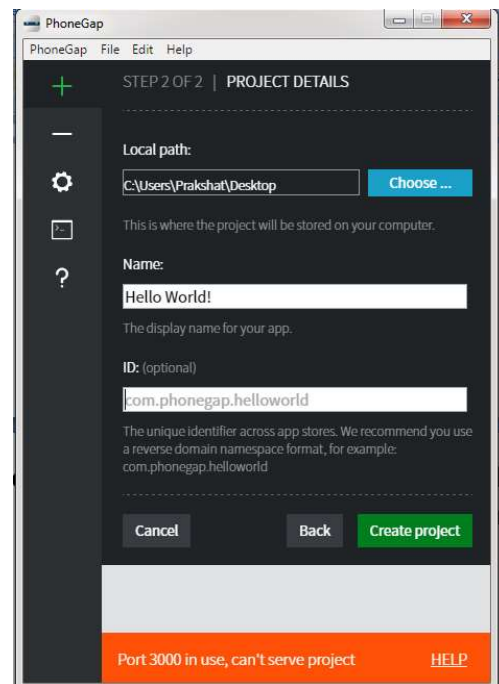
asked to choose Project or Open



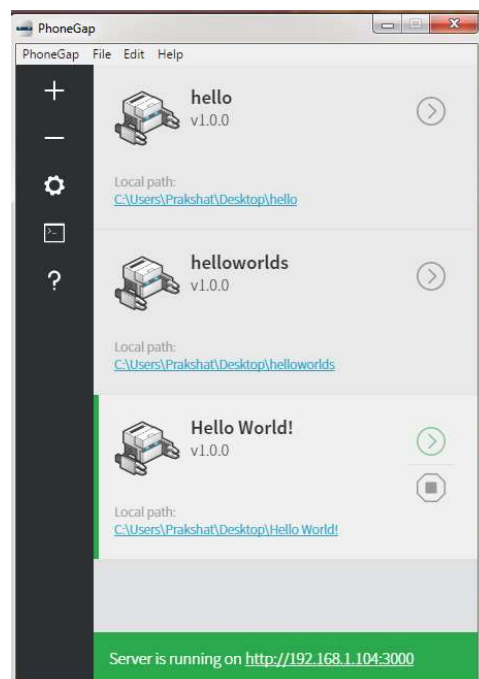
Create “New Project”



Select “Hello World”



Choose the path, create Name of App, and create ID (optional) then finally click on Create Project.



Click on Server Link.

Then browser open, after that edit in Index file which is in the www folder of your app, remove the selected data.

```

35      * Enable inline JS: add 'unsafe-inline' to default-src
36      * Enable eval(): add 'unsafe-eval' to default-src
37      * Create your own at http://cspisawesome.com
38      -->
39      <!-- <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap:
      'unsafe-inline' https://ssl.gstatic.com; style-src 'self' 'unsafe-inline'; media-src *" />
      -->
40
41      <link rel="stylesheet" type="text/css" href="css/index.css" />
42      <title>Hello World</title>
43  </head>
44
45  <body>
46      <div class="app">
47          <h1>PhoneGap</h1>
48          <div id="deviceready" class="blink">
49              <p class="event listening"></p>
50              <p class="event received">Hello World</p>
51          </div>
52      </div>
53      <script type="text/javascript" src="cordova.js"></script>
54      <script type="text/javascript" src="js/index.js"></script>
55      <script type="text/javascript">
56          app.initialize();
57      </script>

```

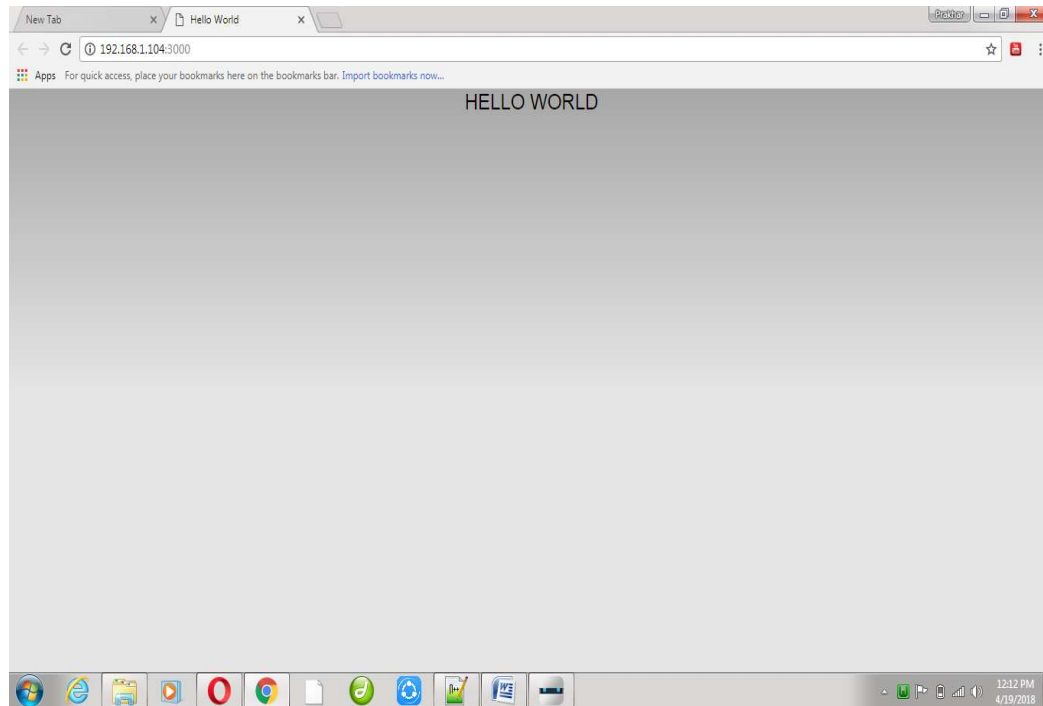
Write Hello World in H1 tag.

```

35      * Enable inline JS: add 'unsafe-inline' to default-src
36      * Enable eval(): add 'unsafe-eval' to default-src
37      * Create your own at http://cspisawesome.com
38      -->
39      <!-- <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap:
      'unsafe-inline' https://ssl.gstatic.com; style-src 'self' 'unsafe-inline'; media-src *" />
      -->
40
41      <link rel="stylesheet" type="text/css" href="css/index.css" />
42      <title>Hello World</title>
43  </head>
44
45  <body>
46      <div class="app">
47          <h1>Hello World</h1>
48      </div>
49      <script type="text/javascript" src="cordova.js"></script>
50      <script type="text/javascript" src="js/index.js"></script>
51      <script type="text/javascript">
52          app.initialize();
53      </script>
54  </body>
55  </html>

```

Then Click on server, then browser shows your desired app.



6.10 POINTS TO REMEMBER

- Compact software program that perform specific task for the mobile user.
- A software application that runs in a handled device such as smartphone or other portable device.
- There are many platforms available like Android, Blackberry, Windows, iphone etc.
- It helps in finding up a new solution in different fields.
- Development requirements- Java, Android SDK, Eclipse (optional)
- Programming Languages- (i) Java-Officially supported (ii) C/C++ also provide but not supported.
- Android is an open source and Linux based operating system for mobile devices.
- It is developed by the Open Handset Alliance led by Google.

6.11 GLOSSARY

- **SDK:** A Software development kit that enables developers to create applications for the Android platform.
- **.APK:** An android package kit is the package file format used by the android operating system for distribution and installation of mobile apps.

- **Global Positioning System:** It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the earth where there is an unobstructed line of sight to four or more GPS satellites.
- **Wi-fi:** wi-fi is a technology that uses radio waves to provide network connectivity.

6.12 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type questions)

- Owns the android platform.
- In Years Google purchase Android.
- Operating system is used as the base of the android stack.
- Required folder when Android project is created.
- language is supported by android for application development.
- .apk extension stand for
- The latest version of android is 8.0. (True/False).
- Android is an open source mobile operating system. (True/False).
- Google licensed some proprietary apps. (True/False).
- ADT stands for Android Development Tools. (True/False).
- Eclipse cannot be used for android development. (True/False).
- A device with android installed is needed to develop apps for android. (True/False).

Part- B (Descriptive type questions)

- What is Android and Android structure?
- What is google android SDK?
- What is the importance of having an emulator within the android environment?
- List out at least four versions of android.
- What is the importance of android in the mobile market?
- What do you think are some disadvantage of android?

Answers- (Part- A- Fill in the blanks and True/False type)		
a) OHA	b) 2005	c) LINUX
d) BIN	e) JAVA	f) APPLICATI ON PACKAGE

g) FALSE	h) TRUE	i) TRUE
j) TRUE	k) FALSE	l) FALSE

6.13 BIBLIOGRAPHY/REFERENCES

- <http://blog.teamgrowth.net/index.php/mobile-application-development/a-brief-introduction-to-mobile-application-development>
- https://en.wikipedia.org/wiki/History_of_mobile_phones
- https://webcache.googleusercontent.com/search?q=cache:80r_1aWfFsEJ:https://www.problogbooster.com/2016/02/Top-Mobile-Operating-Systems-Mobile-OS-Overview-Guide.html+&cd=16&hl=en&ct=clnk&gl=in
- <https://en.wikipedia.org/wiki/Smartphone>
- <https://www.tutorialspoint.com/index.htm>

6.14 SUGGESTED READINGS

- Android Application Development, Black Book, Pradeep Kothari, dreamtech press.

UNIT 7

PHP LANGUAGE CONSTRUCTS

- 7.1. INTRODUCTION
- 7.2. OBJECTIVES
- 7.3. PHP CONDITIONAL STATEMENTS
- 7.4. LOOPING IN PHP
- 7.5. PHP FUNCTIONS
- 7.6. PHP STRINGS
- 7.7. PHP SECURITY
- 7.8. POINTS TO REMEMBER
- 7.9. GLOSSARY
- 7.10. CHECK YOUR PROGRESS
- 7.11. BIBLIOGRAPHY/REFERENCES
- 7.12. SUGGESTED READINGS

7.1 INTRODUCTION

PHP (Hypertext Preprocessor) is a server-side scripting language, which is designed for web development, The PHP originally stood as Personal Home Page. PHP code may be embedded into a HTML code or can be written in a separate file with extension dot (.) php. PHP is a programming language which can be useful to evaluate form data sent from a browser, build custom web content to serve the browser, talk to a database, and etc. Php code is executed on the server and generates HTML code which is then sent to the client's browsers. The client would receive the results sent by the server and displays on the browser.

7.2 OBJECTIVES

After successful completion of this unit you will be able to-

- Overall introduction to PHP
- Include php in a web page with html and css
- Use php conditional and looping statements

- Understand the concepts of functions in php.

7.3 PHP CONDITIONAL STATEMENTS

Like the majority of programming languages, PHP also allows you to write code that execute different actions based on the results of a logical or comparative test condition at run time. You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

1. The **if** statements
2. The **if...else** statement
3. The **if...elseif.... else** statement
4. The **switch...case** statement

The if Statement-

The **if** statement is used to execute a block of code only if the specified condition evaluates to true.

Syntax-

```
if(condition){
    // Code to be executed
}
```

Example-

```
<?php
    $t = 'Hi';
    if ($t == 'Hi') {
        echo "hello dear! Have a good day";
    }
?>
```

The if...else Statement-

The **if....else** statement executes some code if a condition is true and another code if that condition is false. You can improve the decision-making procedure by providing an alternative choice through adding an **else** statement to **if** statement.

Syntax-

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

Example-

```
<?php
    $t = 'Hi';
```

```

if ($t == 'Hi') {
    echo "hello dear! Have a good day";
} else {
    echo "Bye Bye !";
}
?>

```

The if...elseif...else Statement-

The *if...elseif...else* a special statement that is used to combine multiple *if...else* statements.

Syntax-

```

if(condition){
    // Code to be executed if condition is true
} elseif(condition){
    // Code to be executed if condition is true
} else{
    // Code to be executed if condition is false
}

```

Example-

```

<?php
$t = 'Hi';
if ($t == 'Hi'){
    echo "hello dear! Have a good day";
} elseif ($t == 'Bye'){
    echo "Bye Bye dear!";
} else {
    echo "Have a good night!";
}
?>

```

The switch...case statement-

Once you get to the point where you have very difficult if-elseif-else conditionals, you may find that it saves you time and clarifies your programme to use a switch conditional instead. Use the switch statement to select one of many blocks of code to be executed.

Syntax-

```

switch ($Variable) {
    case "value1":
        statement(s)1;
    break;
    case "value2":
        statement(s)2;
    break;
}

```

```

    default:
        statement(s);
    break;
}

```

Example-

```

<?php
    $favcolor = "green";
    switch ($favcolor){
        case "pink":
            echo "Your favorite color is pink!";
            break;
        case "black":
            echo "Your favorite color is black!";
            break;
        case "white":
            echo "Your favorite color is white!";
            break;
        default:
            echo "Your favorite color is neither pink, black, nor white!";
    }
?>

```

Example- Program to check whether the given number represents the weekday or not.

```

<?php
    $day=3;
    if ($day==1){
        echo "Today is Monday";
    }elseif($day==2){
        echo "Today is Tuesday";
    }elseif($day==3){
        echo "Today is Wednesday";
    }elseif($day==4){
        echo "Today is Thursday";
    }else{
        echo "Do not represents any day";
    }
?>

```

7.4 LOOPING IN PHP

Loops are used to execute the same block of code again and again, until a certain condition is met. The basic idea behind a loop is to automate the repetitive tasks within a program to save the time and effort.

PHP supports four different types of loops.

1. **while** — loops through a block of code until the condition is evaluate to true.

2. **do...while** — the block of code executed once and then condition is evaluated. If the condition is true the statement is repeated as long as the specified condition is true.
3. **for** — loops through a block of code until the counter reaches a specified number.
4. **foreach** — loops through a block of code for each element in an array.

The while Loop-

The while loop execute a block of code, as long as the specific condition is true.

Syntax-

```
while (condition is true) {
    //code to be executed;
}
```

Example-

```
<?php
    $y = 1;
    while($y <= 10) {
        echo "The number is: $y <br>";
        $y++;
    }
?>
```

The do...while Loop-

With a do-while loop the block of code executed once, and then the condition is evaluated, if the condition is true, the statement is repeated as long as the specified condition evaluated to true.

Syntax-

```
do{
    //code to be executed;
}while (condition);
```

Example-

```
<?php
    $y = 1;
    do {
        echo "The number is: $y <br>";
        $y++;
    } while($y <= 10);
?>
```

The for Loop-

The for loop repeats a block of code until a certain condition is met. It is typically used to execute a block of code for certain number of times.

Syntax-

```
for(initialization; condition; increment/decrement){
    // Code to be executed
}
```

Initialization: Initialize the loop repeating value.

Condition: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

Increment/Decrement: Increases/decreases the loop repeating value.

Example-

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

The foreach Loop-

The foreach loop works only on arrays. It is used to loop through each key/value pair in an array.

Syntax-

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

For every loop iterations, the value of the current array element is assigned to \$currentValue and the array pointer is moved by one, until it reaches the last array element.

Example-

```
<?php
$months = array("Jan", "Feb", "March", "April");
foreach ($months as $curValue) {
    echo "$ curValue <br>";
}
?>
```

7.5 PHP FUNCTIONS

Creating functions can save your time over the course of your programming life. In fact, they constitute a strong step towards creating Web applications and building a solid library of PHP code to use in future projects. A function is a block of statements that can be used repeatedly in a program. There are following ways to create a function. As-

- Creating and using simple functions without parameters
- Creating and using functions with parameters
- Creating and using functions with default parameters
- Returning values from a function

Creating and using simple functions without parameters-

The basic syntax of creating a custom function can be given as-

Syntax

```
function FunctionName () {
    statement(s);
}
```

Example-

```
<?php
// Defining function
function dispMsg(){
    echo "Hello world";
}
// Calling function
dispMsg();
?>
```

Creating and using functions with parameters-

Information can be passed to functions through parameters. A parameter is just like a variable. You can specify parameters when you define your function to accept input values at run time.

Syntax-

```
function myFunc($Parameter1, $Parameter2){
    // Code to be executed
}
```

Example-

```
<?php
// Defining function
function getSum($num1, $num2){
    $sum = $num1 + $num2;
    echo "Sum of the two numbers is : $sum";
}
// Calling function
getSum(10, 20);
?>
```

Creating and using functions with default parameters-

The following example shows how to use a default parameter.

Example-

```
<?php
function setvalue($callValue = 50) {
    echo "The value passed is : $callValue <br>";
}
setvalue(350);
setvalue(); // will use the default value of 50
?>
```

Returning values from a function-

A function can return a value back to the script that called the function using the return statement. A function cannot return multiple values.

Example-

```
<?php
// Defining function
function getSum($num1, $num2){
    $total = $num1 + $num2;
    return $total;
}
// Printing returned value
echo getSum(5, 10); // Outputs: 15
?>
```

7.6 PHP STRINGS

A string is a series of letters, numbers, special characters and arithmetic values or combination of all. The simplest way to create a string is to enclose the string literal (i.e. string characters) in single quotation marks ('). You can also use double quotation marks (").

Example-

```
$my_string = 'Hello World';
```

PHP provides a number of built-in string functions which help in performing several operations while dealing with string data.

Basic PHP String Functions

Getting length of a String-

PHP has a predefined function to get the length of a string. Strlen() displays the length of any string. It is more commonly used in validating input fields where the user is limited to enter a fixed length of characters.

Syntax-

```
Strlen(string);
```

Example-

```
<?php
    echo strlen("Welcome to Computer world");
    //will return the length of given string
?>
```

Output- 25

Counting of the number of words in a String-

Another function which enables display of the number of words in any specific string is `str_word_count()`. This function is also useful in validation of input fields.

Syntax-

```
Str_word_count(string);
```

Example-

```
<?php
    echo str_word_count("Welcome to google");
    //will return the number of words in a string
?>
```

Output- 3

Reversing a String-

`Strrev()` is used for reversing a string. You can use this function to get the reverse version of any string.

Syntax-

```
Strrev(string);
```

Example-

```
<?php
    echo strrev("Welcome to Computer world");
    //will return the string starting from the end
?>
```

Output- dlrow retupmoC ot emoclew

Finding Text within a String-

`Strpos()` enables searching particular text within a string. If found, then it returns the specific position of the searched string.

Syntax-

```
Strpos(string,text);
```

Example-

```
<?php
    echo strpos("Welcome to Google","Google");
?>
```

Output: 11

Replacing text within a string-

Str_replace() is a built-in function, basically used for replacing specific text within a string.

Syntax-

Str_replace(string to be replaced,text,string)

Example-

```
<?php
    echo str_replace("google", "the programming world", "Welcome to google");
?>
```

Output- Welcome to the programming world

Converting lowercase into Title Case-

Ucwords() is used to convert first alphabet of every word into uppercase.

Syntax-

Ucwords(string);

Example-

```
<?php
    echo ucwords("welcome to the php world");
?>
```

Output-

Welcome To The Php World

Converting a whole string into UPPERCASE-

Strtoupper() is used to convert a whole string into uppercase.

Syntax-

Strtoupper(string);

Example-

```
<?php
    echo Strtoupper("welcome to the php world");
?>
```

Output-

WELCOME TO THE PHP WORLD

Converting a whole string into lowercase-

Strtolower() is used to convert a whole string into lowercase.

Syntax-

Strtolower(string);

Example-

```
<?php
    echo Strtolower("WELCOME TO THE PHP WORLD");
?>
```

Output-

welcome to the php world

7.7 *PHP SECURITY*

There are number of security issues in PHP (or we can say these issues are found in almost all server side languages). These issues must be taken into consideration before start developing application in php. We name some of them here.

- SQL Injection
- XSS (Cross Site Scripting)
- Source Code Revelation
- Remote File Inclusion
- Session Hijacking
- Cross Site Request Forgery
- Directory Traversal

One should know the techniques to avoid these security breaches. These issues can be handled by knowing some advanced topics of PHP. Those topics are beyond the scope of this chapter. You can try to find the solution of above topics on your own.

7.8 *POINTS TO REMEMBER*

- The PHP code can easily be embedded into HTML5 markup or HTML.
- PHP which is now officially known as ‘Hypertext Preprocessor’ was released in the year 1995.
- It is possible to use PHP in almost every operating system. PHP can be used in all major operating systems including Linux, Microsoft Windows, Mac OS X, and RISC OS.
- Some of the biggest online brands, such as Facebook, ProProfs, Digg, Friendster, Flickr, Technorati, and Yahoo! are powered by PHP.

7.9 *GLOSSARY*

- XAMPP: XAMPP is a completely free, easy to install Apache distribution containing Apache, MySQL, PHP, and Perl.
- Apache: The Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software.
- PHP: PHP is a server-side scripting language designed primarily for web development.
- MySQL: MySQL is an open-source relational database management system.
- PhpMyAdmin: phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB.

- PERL: Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.
- XSS: Cross-site scripting is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users.
- SQL: SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.
- Web Server: A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web.
- Client: A client is a piece of computer hardware or software that accesses a service made available by a server.

7.10 CHECK YOUR PROGRESS

Part- A (Multiple choice type questions)

- (1) PHP files have a default file extension of?

(A) .html	(B) .php
(C) .js	(D) .phh
- (2) Which of the looping statements is/are supported by PHP?

(i) for loop (ii) while loop (iii) do-while loop (iv) foreach loop
(A) (i) and (ii)
(B) (i), (ii) and (iii)
(C) All of the above
(D) None of the above
- (3) Which of the following php statement/statements will store 111 in variable num?

(A) int \$num = 111;	(B) int mum = 111;
(C) \$num = 111;	(D) 111 = \$num;
- (4) If \$a = 12 what will be returned when (\$a == 12) ? 5 : 1 is executed?

(A) 12	(B) 1
(C) Error	(D) 5
- (5) Which of the following PHP statements will output Hello World on the screen?

(i) echo ("Hello World"); (ii) print ("Hello World");
(iii) printf ("Hello World"); (iv) sprintf ("Hello World");
(A) (i) and (ii)
(B) (i), (ii) and (iii)
(C) All of the above
(D) (i), (ii) and (iv)

Part- B (Descriptive type questions)

1. Define PHP.
2. How can you display text with a PHP script?
3. How do you define a constant in PHP?
4. What is the functionality of Strtolower() function?
5. What are the real web applications of AJAX currently running in the market?

6. What is phpmyAdmin?
7. What are the components used within XAMPP?
8. What is the use of "echo" in php?
9. How to declare an array in php?
10. What is use of count() function in php ?
11. What types of loops exist in php?
12. How to redirect a page in php?
13. What is mean by an associative array?
14. What is SQL injection?
15. How variables are declared in PHP (syntax only).
16. How function is declared in php?
17. How can we count the number of words in a String using predefined PHP function?
18. What is the use of Ucwords()?
19. Write a program that displays a different message based on the time of day. For example, if it is in the morning, have the site display "Good Morning!"

Answers- (Part A- Multiple choice type questions)

1 D, 2 B, 3 A, 4 D, 5 B

7.11 BIBLIOGRAPHY/REFERENCES

- Robin Nixon, 2012, "Learning PHP, MySQL, JavaScript, and CSS", published by O'Reilly Media.
- Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, 2005, "Beginning PHP5, Apache, and MySQL® Web Development", published by Wiley Publishing Inc.
- <http://codewalkers.com>, February 2018
- <http://developer.yahoo.com/php/>, February 2018
- <http://easyphp.org>, February 2018
- <http://forums.devshed.com>, January 2018
- <http://free-php.net>, January 2018
- <http://hotscripts.com/category/php/>, January 2018

7.12 SUGGESTED READINGS

- Robin Nixon, 2012, "Learning PHP, MySQL, JavaScript, and CSS", published by O'Reilly Media. "Head First Ajax", Rebecca M. Riordan, ISBN-13: 978-0596515782

- David Powers, “PHP Solutions: Dynamic Web Design Made Easy 2nd ed. Edition” ISBN-13: 978-1430232490
- Larry Ullman, “PHP for the Web: Visual QuickStart Guide (4th Edition)”, ISBN-13: 978-8131766729
- Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, 2005, “Beginning PHP5, Apache, and MySQL® Web Development”, published by Wiley Publishing Inc.

UNIT 8

ADVANCE PHP

8.1. INTRODUCTION

8.2. OBJECTIVES

8.3. PREREQUISITE

8.4. OPERATIONS ON DATE & TIME IN PHP

8.5. IMAGE UPLOADING USING PHP

8.6. FILE HANDLING IN PHP

8.7. ERROR HANDLING IN PHP

8.8. MINI PROJECT IN PHP

8.9. POINTS TO REMEMBER

8.10. GLOSSARY

8.11. CHECK YOUR PROGRESS

8.12. BIBLIOGRAPHY/REFERENCES

8.13. SUGGESTED READINGS

8.1 INTRODUCTION

In previous units we have learned how to create simple web pages using php and then how to run those pages on Apache HTTP server. In this unit we will now focus on some of the advanced topics in PHP. These topics are very important when we develop project/dynamic website using php.

8.2 OBJECTIVES

After successful completion of this unit you will be able to-

- Gain knowledge about some of the advanced topics of PHP.
- Use Date & Time, Image Uploading, File Handling, Error Handling etc.
- Practice a mini project developed in PHP, so that you can have knowledge of how a project is handled using PHP.

8.3 PREREQUISITE

To understand this unit better, knowledge of HTML & Basic PHP is needed.

8.4 OPERATIONS ON DATE & TIME IN PHP

PHP provides predefined date and time functions to give better functionalities over these data types. These functions arrange the date and time into a user-friendly readable format. The date() & time() methods in PHP allow to get the date and time of the web server. We can use these functions to get different date and time formats.

Date() method-

Syntax-

date(string format, int timestamp);

Date() method returns formatted date string as a result. The second parameter “timestamp” is optional, if not provided the method will use the local timestamp.

Examples-

```
<?php
    echo date("Y-m-d"); //current Date, i.e. 2018-04-16
    echo date("h:i:sa"); //current Time in 12 Hr format, i.e. 07:57:55pm
    echo date("H:i:s"); //current Time in 24 Hr format, i.e. 18:00:23
    echo date("m"); //current Month, i.e. 08
    echo date("M"); //current Month name, i.e. Aug
?>
```

Time() method-

The Timestamp-

A timestamp is a number of seconds from January 1, 1970, at 00:00.

We can get current timestamp by using time() method.

Example-

```
<?php
    echo time(); //current Timestamp, i.e. 1523888078
?>
```

The current timestamp will display the timestamp according to the default time zone as set by PHP. The default time-zone used and displayed by PHP is UTC time (Coordinated Universal Time), and this is why PHP's time is always some hours off from our local time.

For getting correct timestamp we have to set time zone according to our location. This can be set by a method date_default_timezone_set() method.

Example-

```
<?php
    date_default_timezone_set('India/Kolkata');
?>
```

The above method sets the default time zone to Kolkata, India. Now you can check the timestamp, it will display the correct time-stamp, time and date.

PHP date format string options-

The date() function offers some formatting options. You can also modify the output from the format string. The following is the list of formatting options available:

<i>Format</i>	<i>Description</i>
a	am or pm
A	AM or PM
d	day of the month, two digits with leading zeros; i.e., “08” to “29”
D	day of the week, textual, three letters; i.e., “Sat”
h	hour, 12-hour format; i.e. “01” to “12”
H	hour, 24-hour format; i.e. “00” to “23”
i	minutes; i.e. “00” to “59”
m	month; i.e. “01” to “12”
M	month, textual, 3 letters; i.e. “Jan”
s	seconds; i.e. “00” to “59”
w	day of the week, numeric, i.e., “0” (Sunday) to “6” (Saturday)
Y	year, 4 digits; i.e. “2018”
y	year, 2 digits; i.e. “18”

8.5 IMAGE UPLOADING USING PHP

PHP can be used to allow users to upload images to the server. This is done by creating an HTML form and including PHP script in it to upload image. You can upload any type of file in the server using this method.

The procedure of uploading an Image/File to server is:

- The user opens the page containing a HTML form containing an input file field, a browse button (to select file) and a submit button.
- The user clicks the browse button and selects a file to upload from the local computer.
- The path to the selected file appears in the text field, after that user clicks the submit button.
- The selected file is sent to the proposed directory in the server.
- The PHP script reply with the success message to the user.

Steps for uploading Image File to the Server:

Step 1: Create an HTML form to upload an Image

Example-

```
<html>
  <head>
    <title>Image Uploaing using PHP</title>
  </head>
  <body>
    <form action="uploadImage.php" method="post" enctype="multipart/form-
data">
      <h1>Upload Image</h1>
      <label>Filename:</label>
      <input type="file" name="photo" id="photo">
      <input type="submit" name="submit" value="Upload">
    </form>
  </body>
</html>
```

Some rules to be followed for the above HTML form:

- Make sure that the “post” method is used by form.
- enctype=“multipart/form-data” attribute for the form is to be used to specify which content-type is used when submitting the form.

Save this file as “form.html” in the htdocs folder in XAMPP server.

Step 2: Create a folder named *uploads* where the file is going to be placed.**Step 3: Now create a file with following PHP script:**

- a) In this file we first check that user has selected an image to upload or not, with the help of if condition.

```
<?php
    if(!empty($_FILES['photo']))
?>
```

- b) After that we set the path of image with upload folder into a variable.

```
<?php
    $path = "uploads/";
    $path = $path . basename( $_FILES['uploaded_file']['name']);
?>
```

- c) At last move the image to the definid path with a php function.

```
<?php
    move_uploaded_file($_FILES['uploaded_file']['tmp_name'], $path)
?>
```

Example- (The entire code is below)

```
<?PHP
```

```

if(!empty($_FILES['photo']))
{
    $path = "uploads/";
    $path = $path . basename( $_FILES['uploaded_file']['name']);
    if(move_uploaded_file($_FILES['uploaded_file']['tmp_name'], $path))
    {
        echo "The file “. basename( $_FILES['uploaded_file']['name']),“ has
        been uploaded”;
    } else{
        echo "There was an error uploading the file, please try again!";
    }
}
?>

```

Save this file with name uploadImage.php and save where you have saved form .html.

For running the above application follow the following steps:

1. Run XAMPP server.
2. Open browser, and type “localhost/form.html” in the address bar.
3. Upload form will be displayed on the browser screen.
4. Select an image to upload and press submit.
5. PHP script will submit your image to “upload folder”.

8.6 FILE HANDLING IN PHP

Well, we know that a website is used to display data (information) to the world. Now, data is most probably stored in a database, but we can also use text files to store that data. PHP file handling feature provide us the functions for creating, uploading, and updating text files.

Opening a File-

“Fopen()” method is used to open a file.

Syntax-

```
$file = fopen( “filename.txt”, “r” );
```

Various file opening modes are-		
Modes	Action	Description
‘r’	Open a file for read only	Place the file pointer at the beginning of the file. Return FALSE if the file doesn’t already exist.
‘r+’	Opens the file for reading and writing	Place the file pointer at the beginning of the file. Return FALSE if the file doesn’t already exist

'w'	Open a file for write only	Place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'w+'	Opens the file for reading and writing	Place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'a'	Open a file for write only	Place the file pointer at the end of the file. If the file doesn't exist, attempt to create it.
'a+'	Opens the file for reading and writing	Place the file pointer at the end of the file. If the file doesn't exist, attempt to create it.

We like have two argument supplied to “fopen()” method, one is the filename and other one is mode of file (like r for reading etc.) to which file must be opened.

Reading from a File-

Fread() method is used to reading a file.

Syntax-

```
$filesize = filesize($file);
$filetext = fread( $file, $filesize );
```

First parameter in “fread()” method is the name of the file (file name is included in \$file variable from fopen() method). \$filesize variable has the total size of file opened.

Closing a file-

“Fclose()” method is used to reading a file.

Syntax-

```
fclose($file);
```

Writing into a file-

“Fwrite()” method is used to writing content a file.

For writing data into a file we have to open file with ‘w’ mode in *fopen()* method.

Syntax-

```
$file = fopen( “filename.txt”, “w” );
fwrite($file, “hello from PHP”);
fclose($file);
```

The above code writes “hello from PHP” to a file named filename.txt. This file will be created by fopen() method if not created. One thing that should be noted here is that ‘w’ mode will erase all the text from the file if the file is already existed in the system. If you want to open that file for appending the text, then use ‘a’ mode rather than ‘w’ in the fopen() method.

Renaming a file-

“rename()” method is used to change the name of a file.

Syntax-

```
Rename(“filename.txt”, “newfilename.txt”);
```

Deleting a file-

“unlink()” method is used to delete a file.

Syntax-

```
unlink(“filename.txt”);
```

8.7 ERROR HANDLING IN PHP

Handling errors is the most important part of any program. All languages have error handling techniques. PHP have large set of error handling methods.

Types of PHP errors-

There can be two types of errors in PHP program:

- Internal Errors
- External Errors

Internal Errors-

These are the most common error raised due to programming techniques. These are also known as logical errors. These errors can be prevented by careful programming.

External Errors-

These errors occur due to external agents like database, files, networking, etc. These errors can be avoided by working on error handling techniques in programming methodology.

Dealing with PHP errors-

We can deal with error in PHP by following ways:

- 1) **Display the Error:** Error will be shown to the user when the program is being executed.
- 2) **Log the Error:** Errors to be stored and reviewed from a server log file.
- 3) **Act on the Error:** Error should be corrected by taking proper action. This depends on error to error i.e different action will be taken to correct different errors.
- 4) **Ignore the Error:** Ignoring an error is not the proper solution, so always avoid ignoring error.

Showing PHP errors-

We can set the directives in “php.ini” file, in a web server for showing errors in PHP

```
“Error_reporting on”
```

```
“Display_errors on”
```

```
“Log_errors on”
```

Or, PHP errors can be shown by turn on “error_reporting ()” method at run time;

```
<?php
    error_reporting (E_ALL);
?>
```

PHP errors can be logged by PHP code;

```
<?php
    error_log( “This code has errors!” );
?>
```

You can see the errors in your server tmp directory “/tmp/php-error.log”.

PHP predefined error types-

PHP provides built-in error types to describe the nature of an error. Each level is represented by a named constant which can be used by a programmer.

Error Types (Named Constant)	Description
E_ERROR	Fatal run-time errors
E_WARNING	Run-time warnings (non-fatal errors)
E_PARSE	Compile-time parse errors
E_NOTICE	Run-time notices
E_CORE_ERROR	Fatal errors that occur during PHP’s initial startup
E_CORE_WARNING	Warnings (non-fatal errors) that occur during PHP’s initial startup
E_COMPILE_ERROR	Fatal compile-time errors
E_COMPILE_WARNING	Compile-time warnings (non-fatal errors)
E_USER_ERROR	User-generated error message
E_USER_WARNING	User-generated warning message
E_USER_NOTICE	User-generated notice message
E_STRICT	Enable to have PHP suggest changes to your code that will ensure the best interoperability and forward compatibility of your code
E_RECOVERABLE_ERROR	Catchable fatal error
E_DEPRECATED	Run-time notices
E_USER_DEPRECATED	User-generated warning message
E_ALL	All errors and warnings, as supported
E_COMPILE_WARNING	Compile-time warnings (non-fatal errors)

E_USER_ERROR	User-generated error message
E_USER_WARNING	User-generated warning message
E_USER_NOTICE	User-generated notice message
E_STRICT	Enable to have PHP suggest changes to your code that will ensure the best interoperability and forward compatibility of your code
E_RECOVERABLE_ERROR	Catchable fatal error
E_DEPRECATED	Run-time notices
E_USER_DEPRECATED	User-generated warning message
E_ALL	All errors and warnings, as supported

Using PHP Custom error handlers-

The table displayed above in 8.6.2 displays all error handlers. Now we can create an error handler in php to call these handlers and display appropriate errors to the users. We can display our own messages when an error happens. For this, we have to use the “set_error_handler()” function.

Example:

In this example when error occurs error handler function “set_error_handler ()” is called, which then call our error handler function “myErrorHandler ()” and displays the error message.

```
<?php
// error handler function
function myErrorHandler($errno, $errstr) {
    echo "Error: [$errno] $errstr";
}
// set error handler
set_error_handler("myErrorHandler");
// trigger an error
echo($i); // undefined variable!
?>
```

This will display the message

Error: [8] Undefined variable: i

Here [8] is the number of error handler E_NOTICE used for run time notices.

PHP Exception-

Exception handling is used to alter the normal flow of a program if it has some error. The error prone program code is written in “try” block and exception thrown using throw statement is handled in “catch” block.

Using Throw and Try...Catch Statements-

Example-

```
<?php
    try{
        $dividend = 10;
        $divisor = 0;
        if($divisor == 0){
            throw new Exception('Division by zero.');
        } else{
            $quotient = $dividend / $divisor;
            echo "<p>$dividend / $divisor = $quotient</p>";
        }
        echo '<p>divisions successfull.</p>';
    } catch(Exception $e){
        // Handle the exception
        echo "<p>Exception: “ . $e->getMessage() . “</p>”;
    }
    // Continue execution
    echo "<p>Hello World!</p>";
?>
```

Description (Working of above program)-

- “Try” block executes the code as normal program upto that point where exception occurs. After the exception generated PHP stops at that point and attempt to find the corresponding “catch” block. If the corresponding catch block is found, then the code within “catch” block is executed, if not, a fatal error is generated.
- The “catch” block normally catches the exception thrown from inside the try block and creates an object (\$e) containing the exception information. The error message from this object can be retrieved with the Exception's “getMessage ()” method.
- After the exception handled by catch block, execution of the program is continued.

8.8 MINI PROJECT IN PHP

Now, as we are aware of basic and advanced functionality of PHP, we are ready to create a simple yet powerful mini project “feedback system”. In this topic “mini project in php” we will create a system that is ready to take feedback from n number of users.

Prerequisite for creating a mini project in php-

HTML/CSS, PHP and MySQL (Database⁴)

⁴ We will discuss database in Unit 9. Here we discussed MySQL database in reference to the concerning topic only.

Files used-

“Home.php” //for displaying user’s feedback

“feedback.php” //for giving up feedback

“updatefeedback.php” //for updating feedback into the database

Follow the following steps for creating your first php project-**Step 1-**

Create a mysql database named “feedbackDB ” with following tables-

Table 2: “tabFeedback” with fields –

feedbackID (number(10)) //for storing autoID – primary key

feedback (varchar(500)) //for storing feedback

dof (dateandtime) //for storing current timestamp when feedback is made.

Step 2-

Now create a file named “Home.php” with the following code:

```
<html>
  <head>
    <title>Home Page</title>
  </head>
  <body style="background:blue;background-size:100%100%;">
    <table width="100%" height="100%" border="0">
      <tr>
        <td colspan="2"><H2>FEEDBACK SYSTEM</H2></td>
      </tr>
      <tr>
        <td>Feedback Given by Users</td>
        <td><a href="feedback.php">GIVE FEEDBACK</a></td>
      </tr>
      <tr>
        <td colspan="2">
          <?php
            //server connection started
            $db = "feedbackDB";
            $user = "root";
            $pass = "";
            mysql_connect("localhost", $user, $pass);
            //server connection end
            //fetching data from tabFeedback table
            $SQL_Query="select * from tabFeedback order by dof DESC";
            $result = mysql_query($SQL_Query) or die(mysql_error());
            if(mysql_num_rows($result)>0){?>
```

```

height="100%">
    <marquee direction="up" scrollamount="3" width="100%"
    <ul>
        <?php while($row = mysql_fetch_array( $result)){?>
        <li><?php echo $row['feedback']?></li>
        <?php }?>
    </ul>
    </marquee>
    <?php } }?>
</td>
</tr>
</table>
</body>
</html>

```

The above program will display all the feedback given by users in unordered list sorted by “doF (date of feedback)” field in database.

Step 2-

For inserting feedback into database create a file named “feedback.php” with the following code:

```

<html>
<head>
    <title>Feedback Insert Page</title>
</head>
<body style="background:blue;background-size:100%100%;">
    <table width="100%" height="100%" border="0">
        <tr>
            <td colspan="2"><H2>FEEDBACK SYSTEM</H2>
        </td>
        </tr>
        <tr>
            <td>Update Feedback</td>
            <td><a href="home.php">BACK TO HOME</a></td>
        </tr>
        <tr>
            <td colspan="2">
                <form action="updatefeedback.php" method="post">
                    <textarea name="txtFeedback" placeholder="GIVE
FEEDBACK"></textarea>
                    <input type="submit" value="Submit Feedback">
                </form>
            </td>
        </tr>
    </table>

```

```

        </table>
    </body>
</html>

```

When user oress the submit button on the form, it will call a new PHP page “updatefeedback.php” to submit the user feedback into the database.

Step 3-

At last create a file named “updatefeedback.php” with the following code; this file will insert the feedback record to the database:

Example-

```

<?php
    //server connection started
    $db = "feedbackDB";
    $user = "root";
    $pass = "";
    mysql_connect("localhost", $user, $pass);
    //server connection end
    // get the feedback data posted by feedback page
    $feed=$_POST[txtFeedback];
    $query = insert into tabFeedback values('$feed');
    $result = mysql_query($query) or die(mysql_error());
    if($result){ //successfully inserted
        header("Location:home.php"); // go to homepage
    }else{
        echo "error in insertion plese try again";
        header("Location:feedback.php"); go to feedback page
    }
?>

```

Step 4-

Put all three files in Xampp htdocs folder and run home.php file from browser as:

“http://localhost/home.php”

Congratulation you have successfully developed your first php project.

8.9 POINTS TO REMEMBER

- PHP script always runs from the server, therefore, Apache http server (included in XAMPP) has to be installed before working on PHP.
- When uploading image to the server, the folder (to which the file is written - uploads folder) permission is set to RW (read/write) so that file should be written properly to the folder.

8.10 GLOSSARY

- **POST-** The “POST” method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.
- **Exception-** An “Exception” is an event that interrupts the normal flow of the program and also causes the program to stop processing further.

8.11 CHECK YOUR PROGRESS

Part- A (Multiple choice type questions)

- (1) The date () function returns representation of the current date and/or time.
(A) Integer (B) String
(C) Boolean (D) Float
- (2) Which of the following statements can be used to set the time zone in individual scripts?
(A) date_set_timezone('Europe/London');
(B) date_default_timezone_set('Europe/London');
(C) date_set_default_timezone('Europe/London');
(D) date_default_timezone('Europe/London');
- (3) Which function is used to determine whether a file was uploaded?
(A) is_file_uploaded() (B) is_uploaded_file()
(C) file_uploaded("filename") (D) uploaded_file("filename")
- (4) Which version of PHP was added with Exception handling?
(A) PHP 4 (B) PHP 5
(C) PHP 5.3 (D) PHP 6
- (5) Which of the following statements invoke the exception class?
(A) throws new Exception(); (B) throw new Exception();
(C) new Exception(); (D) new throws Exception();
- (6) Which version of PHP introduced E_STRICT Error level?
(A) PHP 4 (B) PHP 5
(C) PHP 5.2 (D) PHP 5.3

Part- B (Descriptive type questions)

1. What are Date and Time methods used in PHP?
2. What are the error issues with PHP?
3. How an error is detected within a program?
4. Why we need to upload images/files to a server.

5. Define exception in a program.
6. Name the block in PHP which is used to handle exception.
7. Name two methods by which errors can be displayed in PHP page.
8. How Image/File is uploaded to the server with the help of PHP? Explain with example.
9. What is the concept of exception handling in PHP? Explain in detail.
10. What are error constants? What is there use in error handling?
11. How text file is handled in PHP? Explain with examples.

Answers- (Part A- Multiple choice type questions)		
(1)	(2)	(3)
(4)	(5)	(6)

8.12 BIBLIOGRAPHY/REFERENCES

- “Wrox Beginning PHP5 Apache and MySQL Web Development”, published by Wrox.
- Robin Nixon, “Learning PHP, MySQL, JavaScript, and CSS”, published by O’reilly
- Andi Gutmans, Stig Sæther Bakken, and Derick Rethans, “BRUCE PERENS’ OPEN SOURCE SERIES”, published by Prentice Hall
- <https://www.w3schools.com>
- <http://www.tutorialspoint.com>
- <http://www.springer.com/gp/book/9783540775959>
- <https://www.phptpoint.com/>

8.13 SUGGESTED READINGS

- “Wrox Beginning PHP5 Apache and MySQL Web Development”, published by Wrox.
- Robin Nixon, “Learning PHP, MySQL, JavaScript, and CSS”, published by O’reilly.
- <https://www.w3schools.com/php/>

UNIT 9

MYSQL BASICS- SQL QUERIES WITH PHP

9.1.	INTRODUCTION
9.2.	OBJECTIVES
9.3.	HISTORY AND DEVELOPMENT PROCESS OF MySQL
9.4.	MYSQL BUILT-IN FUNCTIONS
9.5.	MYSQL QUERIES
9.6.	WORKING WITH SQL QUERIES
9.7.	ACCESSING DATA FROM MYSQL DATABASE USING PHP
9.8.	POINTS TO REMEMBER
9.9.	GLOSSARY
9.10.	CHECK YOUR PROGRESS
9.11.	BIBLIOGRAPHY/REFERENCES
9.12.	SUGGESTED READINGS

9.1 INTRODUCTION

Any website is incomplete if it has no support to database connectivity with its web pages. PHP is a very popular language of web and is most popularly used to develop websites/web portals. It provides various aspects of web programming viz. development of web page, connecting these pages to outside world, creation of databases, handling with data stored in the databases, support of authentic transactions using session handling, cookies, and by other means to create a trusted relationship with clients such as web browsers.

Specifically, as PHP is an open source, it supports one of the popular RDBMS known as MySQL. In this unit, our content will showcase the relationship between PHP 5 scripts and MySQL database technology. The further contents are dedicated to understand the features, uses and applications of MySQL 5 database with PHP 5.

9.2 OBJECTIVES

After successful completion of this unit you will be able to-

- Use the most popular Open Source Relational SQL DBMS.
- Develop various web-based software applications.
- Able to do MySQL programming comfortably.
- Apply the knowledge on already developed projects in MySQL.

9.3 HISTORY AND DEVELOPMENT PROCESS OF MYSQL

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). Databases can be of many types such as Flat File Databases, Relational Databases, and Distributed Databases etc.

MySQL is one of the best RDBMS being used for developing various web-based software applications. It is developed, marketed and supported in 1994 by MySQL AB, which is a Swedish company. MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database.

SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MYSQL is an essential part of almost every open source PHP application.

SQL-

SQL is a standard query language for requesting information from a database and is used to communicate with a database. SQL Stands for "Structured Query Language,"

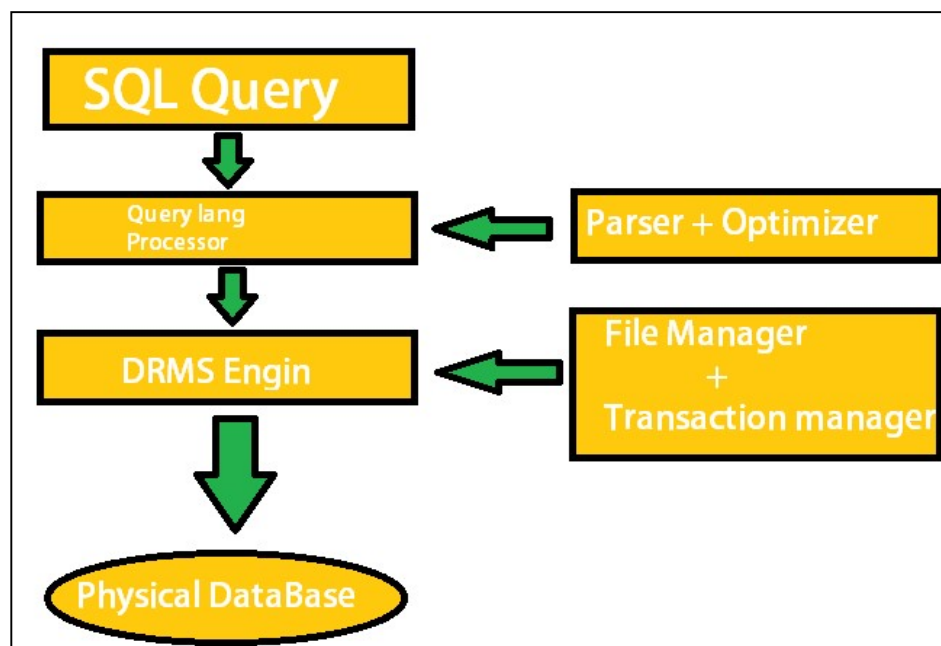
SQL is commonly used for Web database development and management. Though SQL is now considered to be a standard language by using a scripting language like PHP, SQL can execute queries against a database, can update records in a database, and delete records from a database, can delete records from a database, can insert records in a database, and can create new databases.

In 1971, IBM researchers created a simple non-procedural language called Structured English Query Language. Or SEQUEL. This was based on Dr. Edgar F. (Ted) Codd's design of a relational model for data storage where he described a universal programming language for accessing databases.

History of SQL

- 1970 – Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- 1974 – Structured Query Language appeared
- 1978 – IBM worked to develop Codd's ideas and released a product named System/R.
- 1986 – IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

SQL process-



A Typical SQL Architecture

SQL parsing-

The very first stage of SQL processing is parsing. This stage separates the SQL statement into data structure that can be processed. Whenever an application generates a SQL statement, the application gives a Parse Call to the database so that database can prepare the statement for execution. The parse call generated by application creates a Cursor. Which is a medium for the session-specific private SQL area that holds a parsed SQL statement and rest of the processing information.

During parsing, the SQL statement is passed from the user process to database and a parsed representation of the SQL statement is loaded into a shared SQL area. Many errors can be caught during this phase of statement processing. A SQL statement is parsed only if a shared SQL area for an identical SQL statement does not exist in the shared pool. In this case, a new shared SQL area is allocated and the statement is parsed.

During the parse call, the database performs the following operations.

- Syntax Check

- Semantic Check
- Shared Pool Check

Hard Parse-

If Database cannot use existing code again, then it must build a new application code which is executable. This operation is known as a hard parse.

Soft Parse-

A soft parse is any parse that is not a hard parse. If the submitted statement is same as a reusable SQL statement in the shared pool, then Database reuses the existing code. This reuse of code is known as soft parse.

The various components included in this process are:-

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine

9.4 MYSQL BUILT-IN FUNCTIONS

MySQL built-functions plays their vital role in developing the software. Built-in functions are a kind of library that provides the generalized solutions of the general problems viz. statistical functions, mathematical functions, function related to the strings, date and time related functions etc. These functions may return only single value or a table as a result of the function. MySQL has a good number of built-in functions; these functions are very necessary in carrying out various basic operations in MySQL. We cannot use these functions to Insert, Update, and Delete records in the database table. Few commonly used build-in functions available in MySQL are given below with a little description:

MySQL Statistical Functions-

AVG()- Returns the average value of an expression.

SUM()-Returns the summed value of an expression.

CEILING()- Returns the smallest integer value that is greater than or equal to a number.

COUNT()- Returns the count of an expression.

MIN()- Returns the minimum value of an expression.

SIGN()- Returns a value indicating the sign of a number.

FLOOR()- Returns the largest integer value that is equal to or less than a number.

RAND()- Returns a random number or a random number within a range.

ROUND()- Returns a number rounded to a certain number of decimal places.

MAX()- Returns the maximum value of an expression.

MySQL String Functions-

CONCAT()- Concatenates two or more strings together

CHAR()- Returns the ASCII character based on the number code

LOWER()- Converts a string to lower-case

ASCII()- Returns the number code that represents the specific character

LEN()- Returns the length of the specified string

DATALENGTH()- Returns the length of an expression (in bytes)

LEFT()- Extracts a substring from a string (starting from left)

CHARINDEX()- Returns the location of a substring in a string

9.5 MYSQL QUERIES

The SQL queries are the most common and prominent SQL operations. It is used to communicate with database. SQL queries can do the following-

- SQL queries can retrieve data from a database
- SQL queries can insert records in a database
- SQL queries can delete records from a database
- SQL queries can update records in a database
- SQL queries can create views in a database
- SQL queries can create new tables in a database
- SQL queries can create stored procedures in a database
- SQL queries can create new databases

When we want go beyond the simple SQL queries, we need to be more specific while writing the queries. These specific changes in simple SQL queries are also possible by using various clauses. The clauses used are FROM, WHERE, ORDER BY. The little details about this clause is given below:

FROM clause- It indicates the table where the search will be made.

WHERE clause – It is used to define the rows, in which the search will be carried. All rows, for which the WHERE clause is not true, will be excluded.

ORDER BY clause - This is the only way to sort the results in SQL. Otherwise, they will be returned in a random order. The order of the statements may be either ascending (written as ASC in SQL query) or descending (Written as DESC in SQL query). By default, the order is ascending.

It is also to be noted that all the SQL statements start with any of the keywords like SELECT, INSERT, UPDATE, DELETE, ALTER, DROP, CREATE, USE, SHOW and the entire statements end with a semicolon (;).

Syntax of SQL Query-

SELECT EmployeeID FROM table_name WHERE CONDITION;

Some of the frequently used SQL Queries to communicate with database are-

- UPDATE - updates data in a database
- SELECT - extracts data from a database
- DELETE - deletes data from a database
- CREATE DATABASE - creates a new database
- INSERT INTO - inserts new data into a database
- DROP DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

The standard SQL commands to interact with relational databases (RDBMS) are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature. As-

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)

Data Definition Language (DDL)-

The Data Definition Language (DDL) manages table and index structure. In other words DDL is associated with those SQL queries that creates or alters the tables in the database. The most basic queries included in DDL are CREATE TABLE, ALTER TABLE, RENAME and DROP TABLE statements as all these queries are creating, altering or deleting the table to or from the database. The description of few such queries is given below:

- CREATE- Creates an object (a table, for example) in the database.
- DROP- Deletes an entire table, a view of a table or other objects in the database.
- ALTER- Modifies the structure an existing objects in various ways. for example- adding a column to an existing table.

Data Manipulation Language (DML)-

After the structure of the table is created in the database, we need to feed data into these tables. Further this data is filtered out to generate the information. The Data Manipulation Language (DML) is used for these purposes. DML is the subset of SQL used to add, update and delete data from the database. The four operations are considered as a minimum functionality of any standard database. These four operations are Create, Read, Update and Delete and are referred to as CRUD operations. It actually refers to all of the major functions that need to be implemented in a relational database application to consider it at least minimum complete. Each letter in the CRUD can be mapped to a standard SQL statement:

Table 1: CRUD operations of the standard database		
OPERATION	SQL	DESCRIPTION
Create	INSERT INTO	Inserts new data onto a database
Read (or Retrieve)	SELECT	Extracts data from a database
Update	UPDATE	Update data in a database
Delete (Destroy)	DELETE	Delete data from a database

Data Control Language (DCL)-

DCL is short naming of Data Control Language which includes commands which controls the data manipulation or creation. GRANT is most commonly used command under this category and mostly concerned with rights, permissions and other controls of the database system. In other words, GRANT is used to give permission for using database. On the other hand, REVOKE as the name suggests, is used to get the given privileges back which are granted to him/her. These commands are explained below:

- GRANT – It allow users access privileges to database.
- REVOKE – It withdraws users access privileges given by using the GRANT command

Table 2: Grant and Revoke commands	
COMMAND	DESCRIPTION
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

9.6 WORKING WITH SQL QUERIES

The below examples includes on how can we use various queries in PHP and MySQL. Below is "Students" table sample database which will be used to apply MySQL queries for various operations.

Table 3: Sample table named “students” used for queries					
StudentID	StudentName	Course	Address	City	Country
1	Justin	Bca	Mexico	toranto	canada
2	David	Bca	120 hanover	vancour	canada
3	Sophiea	Bsc-it	Oberestr 12	abasford	canada
4	Mark	Mca	71 a.bjanil	Northcarolyn	Usa
5	Lorens	Bsc-it	Monk 34 df	dekota	Usa
6	Donald	Mca	Henry 81 a.h	diksy	canada

Update Query-

Syntax

```
UPDATE                                     table_name
    SET column1 = value1, column2 = value2, ...
    WHERE condition;
```

Example 1

```
UPDATE Students
    SET                               StudentName='Justin',      City='toronto'
    WHERE StudentID = 1;
```

Output

You have made changes to the database. Rows affected: 1

Example 2

```
UPDATE Students
    SET                               StudentName      =      '      Mark',      Country='usa'
    WHERE StudentID = ;
```

Output

You have made changes to the database. Rows affected: 4

Select Query-

Syntax

```
SELECT column1, column2,...
    FROM table_name;
```

Example 1

```
SELECT StudentName, City FROM Students;
```

Output

Table 4: Output of the SELECT query above	
StudentName	City
Justin	Toronto
David	Vancour
Sophiea	Abasford
Mark	Northcarolyn
Lorens	Dekota
Donald	Diksy

Example 2

```
SELECT studentID, Country FROM Students;
```

Output

Table 5: Output of the SELECT query above

StudentID	Country
1	Canada
2	Canada
3	Canada
4	Usa
5	Usa
6	Canada

Delete Query-

Syntax

```
DELETE FROM table_name
WHERE condition;
```

Example 1

```
DELETE FROM Students
WHERE StudentName = 'Justin';
```

Output

You have made changes to the database. Rows affected: 1

Example 2

```
DELETE FROM Students
WHERE StudentID = '4';
```

Output

You have made changes to the database. Rows affected: 4

Create Database-

Syntax

```
CREATE DATABASE databasename;
```

Example 1

```
CREATE DATABASE testDB;
```

NOTE- The above SQL statement create a database called “testDB”

Example 2

```
CREATE DATABASE MyfirstDB;
```

NOTE- The above SQL statement create a database called “MyfirstDB”

Drop Database-

Syntax

```
DROP DATABASE databasename;
```

Example 1

```
DROP DATABASE testDB;
```

NOTE- The above SQL statement drop the existing database called “testDB”

Example 2

DROP DATABASE MyfirstDB;

NOTE- The above SQL statement drops the existing database called “MyfirstDB”

Insert Into Query-

Syntax

```
INSERT INTO table_name (column1, column2,...,ColumnN)
VALUES (value1, value2,...,valueN)
```

Example

```
INSERT INTO Students (StudentName, Course, Address, City, Country)
VALUES ('John', 'Mca', 'D.R edsr gate', 'abbsford', 'Usa');
```

NOTE- Through the above insert query, you have inserted one record into a table.

Create Table-

Syntax

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,...
);
```

Example 1

```
CREATE TABLE Personal_detail (
    CandidateIDint,
    LastNamevarchar(255),
    FirstNamevarchar(255),
    Address varchar(255),
    City varchar(255)
);
```

NOTE- The data type parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.)

Example 2

```
CREATE TABLE Department_IT(
    CandidateIDint,
    Course varchar(255),
    Email-ID varchar(255),
    FatherNamevarchar(255),
    University_Rollnovarchar(255)
);
```

Drop Table-

Syntax

DROP TABLE *table_name*;

Example 1

DROP TABLE Students;

NOTE- Table “students” has been drop from the database.

Note:-Below is "Student_record" table sample database:

Table 6: Sample table named “Student_record” shown					
Name	Age	Father name	Mother name	City	Country
Justin	18	Mr carl	Mrs carl	toranto	canada
David	20	MrTrudo	MrsTrudo	vancour	canada
Sophiea	17	Mrmayer	MrsMayer	abasford	canada
Mark	21	Mrlewis	Mrslewis	Northcarolyn	usa
Lorens	19	Mr joseph h	Mrs joseph	dekota	usa

Example 2

DROP TABLE Student_record;

NOTE- Table “Student_record” has been dropped from database.

Create Index-

Indexes are used to retrieve data from the database very quickly. The Users cannot see the indexes; they are just used to speed Searches/queries. An INDEX is created on columns of a table. One table may contain one or more INDEX tables. The syntax for creating indexes varies from vendor to vendor or we can also say that syntax for creating indexes varies among different databases.

Syntax

CREATE INDEX *index_name*

ON *table_name* (*column1*, *column2*, ...);

Table 7: Sample Table named “customer”					
Customer_code	Customer_city	customer_area	Grade	Amount	Received
001d	Bangalore	Bangalore	A	50000	50000
002s	New york	New york	B	40000	40000
0003a	London	London	A+	30000	30000
004w	Canada	Canada	C	40000	40000
005u	America	America	B+	30000	30000

Example 1

CREATEINDEX Customer_name

```
ON customer(customer_city);
```

Output

Index created.

1.45seconds

Example 2

```
CREATEINDEXCustomer_earnedpoints
```

```
ON customer(customer_Grade);
```

Output

Index created.

1.02seconds

Drop Index-

Dropping indexing means deleting the created index. It may be used in variety of ways in MS Acces, MySQL, DB2/Oracle and SQL Server. The syntax of each DBMS is given for you kind clarification below:

Drop index used in MS Access database

```
DROP INDEX index_name ON table_name;
```

Drop index used in MySQL database

```
ALTER TABLE table_name
```

```
DROP INDEX index_name;
```

Drop index used in DB2/Oracle

```
DROP INDEX index_name;
```

Drop index used in SQL Server

```
DROP INDEX table_name.index_name;
```

9.7 ACCESSING DATA FROM MYSQL DATABASE USING PHP

Create Database Using Php-

CREATE DATABASE command is used to create a database. Example below create a database named "myDB". As-

```
<? Php
```

```
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}
// Create database
```

```

$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE){
    echo "Database created successfully";
}else{
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>

```

Retrieving data from MySQL Database Using Php-

To retrieve the data stored in a table various commands are used in DBMS. Here in case of PHP and MySQL we may also retrieve the records. The queries used in this section are SELECT. Let's use the SELECT command.

Syntax of using SELECT command is:

SELECT field1, field2, ... fieldN

FROM Table_name

[Where Clause];

Example of SELECT query is-

```

<? Php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT FirstName, LastName, Year FROM STUDENT WHERE
Course='MCA'";
$conn->close();
?>

```

UPDATE data using PHP-

The UPDATE statement is used to update existing records in a table. Let us have a look at the "StudentRecord" table-

Id	firstname	lastname	Email
1	Justin	trudo	justin12@gmail.com
2	David	watson	david11@gmail.com

Examples-

The below code updates the record with id=5543 in the table "StudentRecord"

```

<? Php
$servername = "localhost";

```

```

$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE StudentRecord SET lastname='trudo' WHERE id=5543";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else{
    echo "Error updating record: " . $conn->error;
}
$conn->close();
?>

```

DELETE data using PHP-

The DELETE statement is used to delete records from a table. As-

Id	firstname	lastname	Email
1	Justin	trudo	justin12@gmail.com
2	David	watson	david11@gmail.com

Examples below delete the record with id=2 in the "StudentRecord"

<? Php

```

$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// sql to delete a record
$sql = "DELETE FROM StudentRecord WHERE id=2";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else{
    echo "Error deleting record: " . $conn->error;
}
$conn->close();

```

?>

NOTE: There are so many other queries which are not possible to include in this unit. For the sake of the practice you may extend the learning content given below later in the suggested readings.

9.8 POINTS TO REMEMBER

- PHP is light-weight, simple and easy to use.
- PHP and MySQL provide user-friendly environment of web development specially for interactive websites.
- PHP is a server-side programming language and with MySQL you may develop high-quality unique solutions.
- MySQL is an open source relational database management system.
- PHP and MySQL may be used for developing any web projects.
- PHP is interoperable with multiple operating systems and servers.
- PHP is ideal for e-Commerce-based project development.
- PHP reduces the cost of development while increases the efficiency.
- Combination of PHP and MySQL is best suitable for easy, simple and faster web development of e-commerce websites.
- Disadvantage of PHP is that it is not suitable for very large applications where security is also one of the big concerns. Also, PHP has weak type which supports implicit conversions which may rise the unexpected bugs from the programmer's point of view.

9.9 GLOSSARY

- **Browser-** A software application that enables a user to display and interact with text, images, and other information typically located on a web page at a website on the World Wide Web.
- **Database-** An organized collection of data.
- **Host-** Any machine connected to a computer network, a node that has a host name.
- **HTTP-** Hyper Text Transfer Protocol is the primary method used to transfer or convey information on the World Wide Web.
- **Https-** A HTTP- connection with additional security measures.
- **JPEG-** A most commonly used standard method of lossy compression for photographic images.
- **MySQL-** A multithreaded, multi-user, SQL (Structured Query Language) Database Management System (DBMS).
- **Mysql-** The MySQL client PHP extension.
- **Mysqli-** The improved MySQL client PHP extension.
- **PHP-** PHP is “Preprocessor Hypertext”, is an open-source, server side programming language used mainly for developing server-side applications and dynamic web content, and more recently, a broader range of software applications.

- **Port-** A connection through which data is sent and received.
- **Server-** A computer system that provides services to other computing systems over a network.
- **SQL-** Stands for Structured Query Language and used to communicate with the database management system.
- **Table-** A set of data elements (cells) that is organized, defined and stored as horizontal rows and vertical columns where each item can be uniquely identified by a label or key or by its position in relation to other items.
- **URL-** Uniform Resource Locator is a sequence of characters, conforming to a standardized format, that is used for referring to resources, such as documents and images on the Internet, by their location.
- **Web server-** -a computer (program) that is responsible for accepting HTTP requests from clients and serving them Web pages.

9.10 CHECK YOUR PROGRESS

Part- A (Multiple choice and True/False type questions)

- (1) Earlier PHP was known as Preprocessor Home Page.
- (2) A NULL value is treated as a blank or 0.
- (3) The result of a SELECT statement can contain duplicate rows.
- (4) A table may be joined to itself.
- (5) In a SELECT with a GROUP BY clause, a WHERE clause, and a HAVING clause, the WHERE conditions are applied before the HAVING conditions.
- (6) UPDATE query is part of DCL in SQL.
- (7) Which one of the following statements is used to create a table?
 - (A) CREATE TABLE table_name (column_name column_type);
 - (B) CREATE table_name (column_type column_name);
 - (C) CREATE table_name (column_name column_type);
 - (D) CREATE TABLE table_name (column_type column_name);
- (8) What does PHP stand for?
 - (i) Personal Home Page
 - (ii) Hypertext Preprocessor
 - (iii) Pretext Hypertext Processor
 - (iv) Preprocessor Home Page
 - (A) (i) and (iii)
 - (B) (ii) and (iv)
 - (C) Only (ii)
 - (D) (i) and (ii)
- (9) Which one of the following databases has PHP supported almost since the beginning?

(A) Oracle Database	(B)	SQL
(C) SQL+	(D) MySQL	
- (10) The updated MySQL extension released with PHP 5 is typically referred to as

- (A) MySQL (B) mysql
(C) mysqli (D) mysqlly

Part- B (Descriptive type questions)

- (1) How can we create a database using PHP and MySQL?
- (2) What are the key features of MySQL?
- (3) What are the common usages of PHP?
- (4) Describe any two SQL queries each from DDL, DML.
- (5) Describe the use of GRANT and REVOKE queries with an example of each.
- (6) What is the use of the clauses HAVING and ORDER BY?

Answers- (Part A- Multiple choice and True/False type questions)		
(1) False	(2) False	(3) True
(4) True	(5) True	(6) False
(7) A	(8) C	(9) D
(10) C		

9.11 BIBLIOGRAPHY/REFERENCES

- W. Jason Gilmore, (2008), Beginning PHP and MySQL from Novice to professional, Apress, United States of America.
- Rasmus Lerdorf, Kevin Tatroe, Peter Macintyre (April 2006), Programming PHP, O'REILLY, United States of America.
- Paul Hudson, (October 2005), PHP in a Nut Shell, O'REILLY, United States of America.

9.12 SUGGESTED READINGS

- David Sklar & Adam Trachtenberg, PHP Cookbook, O'REILLY, United States of America.
- Andy Harris, (2009), PHP6/MySQL Programming for absolute beginner, CENGAGE Learning, Canada.