Class: - B.Com, B.Com (Hons), BBA & BAJMC II Year

Subject: - Web Development Using PHP & MySQL

Unit	Contents
UNIT – I	**BASICS OF PHP:** Introduction to PHP, PHP features, installation of XAMPP/WAMP, Benefits of using PHP MYSQL, Server Client Environment, Web Browser, Web Server Installation & Configuration Files. OOPS with PHP, language basics, syntax, comments, variables, constants and data types, expressions and operators, flow control statements, looping structures, Arrays Including html code in PHP, Embedding PHP in web pages.
UNIT – II	**FUNCTIONS & STRINGS in PHP:** Defining a function, Calling a function, variable scope, function parameters, return values, User Defined Function, System Defined Function, Parameterized Function, Date & Time Function, Hash Function, Mail Function, predefined functions. Strings: Creating & accessing string, searching and replacing strings, encoding and escaping, comparing strings, formatting strings, regular expression.
UNIT – III	Data & File Handling: PHP Forms: $_GET, $_POST, $_REQUEST, S_FILES, $_SERVER, SGLOBALS, $ ENV, input/output controls, validation, Cookies and Sessions. File Handling: File and directory, open, close, read, write, append, delete, uploading and downloading files. File exists, File Size, Rename. Reading and display all/selected files present in a directory.
UNIT – IV	**MySQL an Overview:** Introduction, What is a Database, Understanding an RDBMS, Tables, Record & Fields, SQL Language. **Working with phpmyadmin:** Creating and using a database, Selecting a database, creating/dropping a table, loading data into a table, Retrieving information from a table, selecting all data, selecting particular rows, selecting particular columns, writing queries, sorting, date, calculations, working with NULL values, pattern matching, counting rows, using more than one tables, using table and column aliases.
UNIT – V	**MySQL DATABASES IN PHP:** Introduction, connecting to a MySQL database, querying the database, Retrieving and displaying the results, modifying data and deleting data through front end. Designing applications using PHP & MySQL.

Class: - B.Com, B.Com (Hons), BBA & BAJMC II Year **Subject: Web Development Using PHP & MySQL**

UNIT -1

Introduction to PHP & Features

PHP is a server scripting language, and a powerful tool for making dynamic and interactiveWeb pages.

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browseras plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

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

To start using PHP, you can:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL

Use a Web Host with PHP Support

- If your server has activated support for PHP you do not need to do anything.
- Just create some .php files, place them in your web directory, and the server will

automatically parse them for you.

- You do not need to compile anything or install any extra tools.
- Because PHP is free, most web hosts offer PHP support.
- Set Up PHP on Your Own PC

However, if your server does not support PHP, you must:

- install a web server
- install PHP
- install a database, such as MySQL

PHP Scripts Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with **<?php** and ends with **?>**:
- PHP statements end with a semicolon (;)

Example

```
<?php

// PHP code goes
here ?>
```

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

</html>
```

Comments in PHP

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 looking at the code.

Comments can be used to:

- Let others understand what you are doing
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code

Example

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

```
lines
*/
// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo
$x; ?>
</body>
</html>
```

Variables

- Variables are "containers" for storing information.
- Creating (Declaring) PHP Variables
- In PHP, a variable starts with the $ sign, followed by the name of the variable:

Rules for PHP variables:

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

Output Variables

- The PHP echo statement is often used to output data to the screen.

PHP Variables Scope

- In PHP, variables can be declared anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

- local
- global
- static

Global and Local Scope

A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function:

A variable declared **within** a function has a LOCAL SCOPE and can only be accessedwithin that function:

The global Keyword

- The global keyword is used to access a global variable from within a function.
- To do this, use the global keyword before the variables (inside the function):

The static Keyword

Normally, when a function is completed/executed, all of its variables are deleted. However,

sometimes we want a local variable NOT to be deleted. We need it for a further job.

echo and print Statements

- In PHP there are two basic ways to get output: echo and print.
- In this tutorial we use echo (and print) in almost every example. So, this chapter contains a little more info about those two output statements.
- echo and print are more or less the same. They are both used to output data tothe screen.
- The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

Echo Statement

The echo statement can be used with or without parentheses: echo or echo().

Data Types

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

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

PHP String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes:

PHP Integer

An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.

Rules for integers:

- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)
- In the following example $x is an integer. The PHP var_dump() function returns the data type and value:

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- In the following example $x is a float. The PHP var_dump() function returns the

data type and value:

PHP Boolean

A Boolean represents two possible states: TRUE or FALSE.

```
$x = true;
$y = false;
```

Booleans are often used in conditional testing. You will learn more about conditional testing in a later chapter of this tutorial.

PHP Array

- An array stores multiple values in one single variable:
- An array is a special variable, which can hold more than one value at a time.
- An array can hold many values under a single name, and you can access the values by referring to an index number.

Create an Array in PHP

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

```
array();
```

In PHP, there are three types of arrays:

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

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators

PHP Arithmetic Operators

- The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y

-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power (Introduced in PHP 5.6)

PHP Assignment Operators

- The PHP assignment operators are used with numeric values to write a valueto a variable.
- The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

PHP Comparison Operators

- The PHP comparison operators are used to compare two values (numberor string):

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y

===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

PHP Increment / Decrement Operators

- The PHP increment operators are used to increment a variable's value.
- The PHP decrement operators are used to decrement a variable's value.

Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

PHP Logical Operators

- The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
And	And	$x and $y	True if both $x and $y are true
Or	Or	$x or $y	True if either $x or $y is true
Xor	Xor	$x xor $y	True if either $x or $y is true, but not both

&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

PHP String Operators

- PHP has two operators that are specially designed for strings.

Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

PHP Array Operators

- The PHP array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y

==	Equality	$x == $y	Returns true if $x and $y have the same key/valuepairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/valuepairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

1. PHP Flow Control Statements
   - Sequential (discussed till now)
   - Selection(if, if..else,switch ...etc)
   - Iterative(Loops)

- Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.

In PHP we have the following conditional statements (selection)

- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif ..... else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

The if Statement

- The if statement executes some code if one condition is true.

Syntax

if (*condition*) {
code to be executed if condition is true;}

The if...else Statement

The if..... else statement executes some code if a condition is true and another code if that condition is false.

Syntax

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

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

The if.... elseif... else statement executes different codes for more than two conditions.

Syntax

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

The switch Statement

- The switch statement is used to perform different actions based ondifferent conditions.

Use the switch statement to **select one of many blocks of code to be executed**.

Syntax

```
switch (n)
    { case
    label1:
        code to be executed if n=label1;
        brea
    k; case
    label2:
        code to be executed if n=label2;
        brea
    k; case
    label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once. The value of the expression is then compared with the values for

each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically.The default statement is used if no match is found.

PHP Loops

Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal code-lines in a script, we can use loopsto perform a task like this.

In PHP, we have the following looping statements:

- while- loops through a block of code as long as the specified condition is true
- do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- for- loops through a block of code a specified number of times
- foreach- loops through a block of code for each element in an array

The PHP while Loop

- The while loop executes a block of code as long as the specified condition is true.

Syntax

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

The PHP do...while Loop

The do...while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

Syntax

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

for Loops

- PHP for loops execute a block of code a specified number of times.

The PHP for Loop

- The for loop is used when you know in advance how many times the scriptshould run.

Syntax

```
for (init counter; test counter; increment counter) {
    code to be executed;
}
```

Parameters:

- *init counter*: Initialize the loop counter value
- *test counter*: Evaluated for each loop iteration. If it evaluates to TRUE, theloop continues. If it evaluates to FALSE, the loop ends.
- *increment counter*: Increases the loop counter value

The PHP foreach Loop

- The foreach loop works only on arrays, and is used to loop through eachkey/value pair in an array.

Syntax

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

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

Class: - B.Com, B.Com (Hons), BBA & BAJMC II Year **Subject: Web Development Using PHP & MySQL**

Unit-2

Functions

The real power of PHP comes from its functions; it has more than 1000 built-in functions. Functions are blocks of codes that perform specific task in php. It makes the codes reusable and shorthand. it can be categorized into two groups.

- Predefined functions
- User defined functions

Predefined Functions

Predefined function (plural predefined functions) (computing) **Any of a set of subroutines that perform standard mathematical functions included in a programming language**; either included in a program at compilation time, or called when a program is executed. Predefined functions are the inbuilt functions of php. These functions can be subdivided into multiple categories as stated below.

- string functions
- numeric functions
- date and time functions
- array functions
- directory functions

String Function

strtolower(); -> converts all characters of the string to lower case
strtoupper(); -> converts all characters of the string to upper case
ucfirst(); -> converts first letter to upper case
ucwords(); -> converts first letter of each word to upper case
strlen(); -> counts number of characters in a string and returns integer value
trim(); -> trims unnecessary spaces
ltrim(); -> trims unnecessary spaces from left
rtrim(); -> trims unnecessary spaces from right
sprintf("%s,%s,%s",$a,$b,$c); -> placeholder to keep the value
str_word_count() -> count the number of words
$count = str_word_count($x,1) -> returns array
strstr(); echo strstr($str,U,true);
stristr(); case insensitive strstr
str_replace() -> search and replace characters from the string
str_repeat() -> repeats the string
substr(int,int) -> prints a string from defined initial character number to defined last number
strpos() -> finds position of the string

Numeric Function

abs() -> returns positive value of a number
sqrt() -> returns square root of a number
round() -> rounds a floating number
floor() -> rounds a number down to a nearest integer
ceil() -> rounds a number up to a nearest integer
rand() -> generates a random integer
mt_rand() -> generates random number between defined inital and end number
pow() -> returns x raised to the power of y
pi() -> returns the value of pi
min() -> returns the lowest value from an array
max() -> returns the highest value from an array
fmod() -> returns the remainder from x/y {%}
bindec() -> converts a binary number to a decimal number
decbin() -> converts a decimal number to a binary number
deg2rad() -> converts a degree value to a radian value

Array Functions

array() -> creates an array
sizeof() -> counts the number of values in an array
sort() -> sorts an indexed array in ascending order
in_array() -> checks if a specified value is in array

list() -> keep array values in variable
compact() -> keeps variable values in associative array
arsort() -> sorts an associative array in descending order according to the value
array_unique() -> removes duplicate values from an array
explode() -> converts string to array
implode() -> converts array to string
extract() -> converts array to variable
array_sum() -> returns the sum of values in an array
array_shift() -> removes the first element of an array and returns the first value from the array
array_pop() -> deletes the last element from an array
array_merge() -> merges multiple arrays
array_search() -> searches for a defined value in an array and returns the key for that value
array_reverse() -> returns an array in reverse order
array_keys() -> returns all the keys from an array

PHP Date/Time Functions

PHP date() Function: The PHP date() function converts timestamp to a more readable date and time format.

Syntax:

date(format, timestamp)

- The format parameter in the date() function specifies the format of returned date and time.
- The timestamp is an optional parameter, if it is not included then the current date and time will be used.

Formatting options available in date() function: The format parameter of the date() function is a string that can contain multiple characters allowing to generate the dates in various formats. Date-related formatting characters that are commonly used in the format string:

- d: Represents day of the month; two digits with leading zeros (01 or 31).
- D: Represents day of the week in the text as an abbreviation (Mon to Sun).
- m: Represents month in numbers with leading zeros (01 or 12).
- M: Represents month in text, abbreviated (Jan to Dec).
- y: Represents year in two digits (08 or 14).
- Y: Represents year in four digits (2008 or 2014).

The parts of the date can be separated by inserting other characters, like hyphens (-), dots (.), slashes (/), or spaces to add additional visual formatting.

PHP time() Function: The time() function is used to get the current time as a Unix timestamp (the number of seconds since the beginning of the Unix epoch: January 1, 1970, 00:00:00 GMT).
The following characters can be used to format the time string:

- h: Represents hour in 12-hour format with leading zeros (01 to 12).
- H: Represents hour in 24-hour format with leading zeros (00 to 23).
- i: Represents minutes with leading zeros (00 to 59).
- s: Represents seconds with leading zeros (00 to 59).
- a: Represents lowercase antemeridian and post meridian (am or pm).
- A: Represents uppercase antemeridian and post meridian (AM or PM)

checkdate()	Validates a Gregorian date
date_diff()	Returns the difference between two dates
date_format()	Returns a date formatted according to a specified format

date_time_set()	Sets the time
getdate()	Returns date/time information of a timestamp or the current local date/time

PHP User Defined Functions

- Besides the built-in PHP functions, we can create our own functions.
- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute immediately when a page loads.
- A function will be executed by a call to the function.

Defining a Function in PHP

A user-defined function declaration starts with the word function:

Syntax

```
function functionName() {
  code to be executed;
}
```

- Function names are NOT case-sensitive.

In the example below, we create a function named "writeMsg()". The opening curly brace ({) indicates the beginning of the function code and the closing curly brace (}) indicates the end of the function. The function outputs "Hello world!". To call the function, just write its name:

Example

```
<html>
<body>
<?php
function writeMsg()echo "Hello world!";
}
writeMsg();
?>
</body>
</html>
```

PHP Function Arguments

- Information can be passed to functions through arguments. An argument is just like a variable.
- Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

PHP Default Argument Value

The following example shows how to use a default parameter. If we call the function setHeight() without arguments it takes the default value as

argument:

Example

```
<html>
<body>
<?php
function setHeight($minheight = 50) {
   echo "The height is : $minheight
   <br>";
}
setHeight(3
50);
setHeight()
;
setHeight(1
35);
setHeight(8
0);
?>
</body>
</html>
```

PHP Functions - Returning values

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

****