**MODULE 1- CORE PHP**

PHP SYNTAX

THEORY EXERCISE:

**(1)** Discuss the structure of a PHP script and how to embed PHP in HTML

**(2)** What are the rules for naming variables in PHP?

Structure of a php script

a php script is a set of instructions written in the php language to perform tasks like displaying content,handling forms,or connecting to databases.a php script typically has the following structure.

<?Php

echo “hello ,world!”;

?>

**Element of php script:**

1. **Opening and closing tag:**

Php code must be enclosed within <?Php and ?>.

1. **Statement**

Each php statement ends with a semicolon(;**)**

1. **Comments**

Single-line://comment or #comment

Multi-line /\* comment \*/

1. **Function and variables**

used to perform task and store data

1. **Output**

echo and print are commonly used to display content

**Embedding PHP in HTML**

PHP is often embedded inside HTML to make web pages dynamic. You can mix PHP with HTML using the PHP tags.

**<?php**

**echo "<h1>Welcome to my website</h1>";**

**echo "<p>Today is " . date("l") . "</p>";**

**?>**

1. Rules for Naming Variables in PHP

In PHP, variables are used to store data like strings, numbers, or arrays. To use variables correctly, we must follow certain naming rules:

**Basic rules:**

1. **start with $sign**

-> $name = "Shruti";

1. **Start with a letter or underscore (\_)**

-> The variable name must begin with a **letter (A–Z or a–z)** or an **underscore (\_)**, not a number.

Valid: $name, $\_count  
 Invalid: $1number

1. **Can contain letters, numbers, and underscores**  
   But **no spaces or special characters** (like @, #, -)

$user\_123  
 $user-name

1. **case -sensitive**$name and $Name are **different** variables in PHP.

LAB EXERCISE: 

Write a PHP script to print "Hello, World!" on a web page.

**<?php**

**echo "<h1>hello world</h1>";**

**?>**

**PHP Variables:**

1. **Explain the concept of variables in PHP and their scope.**

### What is a Variable?

### ->A **variable** in PHP is used to **store data** (like text, numbers, etc.) that you can use later in your script.

### Rules:

1. All variables start with a $ sign
2. Variable names are case-sensitive
3. You can assign a value using the =operator

**<?php**

**$name = "Shruti";**

**$age = 22;**

**echo $name;**

**?>**

Scope of Variables in PHP

1. local scope
2. Global scope
3. Static scope
4. **Local scope :**

· Variable declared **inside a function**.

· Only available **within** that function.

function test() {

$x = 10; // local scope

echo $x;

}

test();

1. **Global scope:**

Variable declared **outside all functions**.

Can’t be accessed directly **inside** a function.

$x = 20; function test() {

// echo $x;

test();

1. **Static scope**

A static variable keeps its value **even after the function ends**.

**function countCall() {**

**static $count = 0;**

**$count++;**

**echo $count . "<br>";**

**}**

**countCall(); // 1**

**countCall(); // 2**

**countCall(); // 3**

**Super Global Variables**

THEORY EXERCISE:

What are super global variables in PHP? List at least five super global arrays andtheiruse.

Super global variables are **predefined built-in variables** in PHP that are available in **all scopes** of a script.  
They are **accessible from anywhere** — inside functions, classes, or files — **without using the** global **keyword**.

These variables are mostly **associative arrays** and are used to **store information** such as:

(1) $\_GET:

**Use:** To collect form data sent via **URL (GET method)**.

1. $\_post:

**Use:** To collect **secure form data** sent using **POST method**.

1. $\_REQUEST:

**Use:** To collect data from **both GET and POST**, and also from cookies.

1. $\_SESSION:

**Use:** To store **user-specific data temporarily** (until browser is closed).

1. $\_SERVER:

**Use:** To get information about **server, file path, browser**, etc.

1. **Conditions, Events, and Flows**

Explain how conditional statements work in PHP.

Conditional statements are used in PHP to **make decisions** in a program.  
They allow the program to **execute certain blocks of code** depending on whether a given **condition is true or false**.

Types of Conditional Statements in PHP:

1. **if statement**

It checks a condition. If the condition is **true**, the code inside the block runs.

1. **If else statement**

This checks a condition. If the condition is **true**, one block runs; otherwise, the **else** block runs.

1. **If else if else Statement**

It is used to check **multiple conditions**. The first condition that is true will execute its block, and the rest will be skipped.

1. **Switch statement**

It is used to **test one variable** against multiple possible values. It is a cleaner alternative to multiple if...elseif conditions.

1. **If Condition and If-Else If**

LAB EXERCISE: 

Write a PHP program to determine if a number is even or odd using if conditions.

**<?php**

**$number=7;**

**if ($number % 2 == 0) {**

**echo $number . “ is an Even number.";**

**} else {**

**echo $number.” is an Odd number.";**

**}**

**?>**

**Output:** 7 is an Odd number.

**Practical Example: Calculator and Day Finder**

1. Simple Calculator: Create a calculator using if-else conditions that takes twoinputsand an operator (+, -, \*, /).

**<?php**

**$number1=10;**

**$number2=5;**

**$operator="+”;**

**if($operator=="+"){**

**$result = $number1 + $number2;**

**echo"Result: $number1 + $number2 = $result";**

**}**

**elseif($operator=="-"){**

**$result=$number1-$number2;**

**echo"Result:$number1-$number2 = $result";**

**}**

**elseif($operator=="\*"){**

**$result=$number1\*$number2;**

**echo"Result:$number1\*$number2 = $result";**

**}**

**elseif ($operator == "/") {**

**if ($number2 != 0) {**

**$result = $number1 / $number2;**

**echo "Result: $number1 / $number2 = $result";**

**} else {**

**echo "Error: Cannot divide by zero!";**

**}**

**}**

**else {**

**echo "Invalid operator!";**

**}**

**?>**

1. Day Finder: Write a script that finds the current day. If it is Sunday, print "HappySunday."

**<?php**

**$day = date("l");**

**echo "Today is: $day<br>";**

**if ($day == "Sunday") {**

**echo "Happy Sunday.";**

**}**

**?>**

Switch Case and Ternary Operator

1. Restaurant Food Category Program: Use a switch case to display the category(Starter/Main Course/Dessert) and dish based on user selection.

**<?php**

**$choice = 5; // Change this to 1, 2, or 3**

**switch ($choice) {**

**case 1:**

**echo "Category: Starter<br>";**

**echo "Dish: Spring Rolls";**

**break;**

**case 2:**

**echo "Category: Main Course<br>";**

**echo "Dish: Veg Biryani with Raita";**

**break;**

**case 3:**

**echo "Category: Dessert<br>";**

**echo "Dish: ice-cream";**

**break;**

**default:**

**echo "Invalid selection. Please choose 1, 2, or 3.";**

**}**

**?>**

1. Ternary Operator Example: Write a script using the ternary operator to displayamessage if the age is greater than 18.

**<?php**

**$age = 20;**

**$message = ($age > 18) ? "You are eligible" : "You are not eligible”;**

**echo $message;**

**?>**

**(3)**Color Selector: Write a program to display the name of a color based on user input (red, green, blue).

**<?php**

**$color="red";**

**$color=strtolower($color);**

**switch ($color){**

**case "red":**

**echo"You selected RED.";**

**break;**

**case "green":**

**echo"you selected GREEN.";**

**break;**

**case "blue":**

**echo"you selected BLUE.";**

**break;**

**default:**

**echo"invalid color selected.";**

**}**

?>

**Loops:**

**Do-While, For Each, For Loop**

Discuss the difference between for loop, foreach loop, and do-while loop in PHP.

1. **For loop**

A for loop is used when the number of iterations is **known or fixed**. It includes initialization, condition, and increment/decrement in one line.

1. **Foreach loop**

A foreach loop is used to iterate through arrays. It works only with arrays and gives each element one by one.

1. **Do while loop**

A do-while loop executes the code block **at least once**, and then checks the condition.  
It runs the code **first**, then checks the condition.

**LAB EXERCISE**

(1)For Loop: Write a script that displays numbers from 1 to 10 on a single line.

**<?php**

**for ($i = 1; $i <= 10; $i++) {**

**echo $i . " ";**

**}**

**?>**

**(2)**For Loop (Addition): Add all integers from 0 to 30 and display the total.

**<?php**

**$sum = 0;**

**for ($i = 0; $i <= 30; $i++) {**

**$sum += $i; // same as $sum = $sum + $i;**

**}**

**echo "The total sum from 0 to 30 is: $sum";**

**?>**

1. Various Patterns: Generate different patterns using loops.

**<?php**

**for ($i = 1; $i <= 4; $i++) {**

**for ($j = 1; $j <= $i; $j++) {**

**echo "\* ";**

**echo "<br>";**

**}**

**>?**

1. Chessboard Pattern: Use a nested loop to create a chessboard pattern (8x8 grid).

**<html>**

**<head>**

**<title>Chessboard Pattern</title>**

**<style>**

**table {**

**border-collapse: collapse;**

**}**

**td {**

**width: 60px;**

**height: 60px;**

**}**

**</style>**

**</head>**

**<body>**

**<h2>8x8 Chessboard Pattern</h2>**

**<table border="1">**

**<?php**

**for ($row = 1; $row <= 8; $row++) {**

**echo "<tr>";**

**for ($col = 1; $col <= 8; $col++) {**

**$total = $row + $col;**

**if ($total % 2 == 0) {**

**echo "<td style='background-color: white;'></td>";**

**} else {**

**echo "<td style='background-color:black;'></td>";**

**}**

**}**

**echo "</tr>";**

**}**

**?>**

**</table>**

**</body>**

**</html>**

**PHP Array and Array Functions**

1. **Define arrays in PHP. What are the different types of arrays**

An **array** in PHP is a special variable that allows you to store **multiple values in a single variable**  
Instead of creating separate variables for each value, an array stores them all together and can be accessed using an index or key

Types of array php

PHP supports **three main types of arrays**:

1. **Indexed array:**

These arrays use **numerical indexes** (starting from 0).

Values are accessed by their numeric index.

$colors = array("Red", "Green", "Blue");

echo $colors[0]; // Output: Red

1. **Associative Array:**

· These arrays use **named keys** instead of numeric indexes.

· Each key is associated with a value (key => value).

$student = array("name" => "Shruti", "age" => 22);

echo $student["name"]; // Output: Shruti

### ****Multidimensional Array****

· These are **arrays containing one or more arrays**.

· Used to store complex data like tables or records.

$marks = array(

"Shruti" => array(90, 85, 88),

"Riya" => array(78, 82, 91)

);

echo $marks["Shruti"][1]; // Output: 85

LAB EXERCISE:

1. Display the value of an array.

**<?php**

**$colors=array("red","orange","blue");**

**foreach($colors as $color){**

**echo$color."<br>";**

**}**

**?>**

1. Find and display the number of odd and even elements in an array

**<?php**

**$number = array(1, 3, 5, 7, 9, 11, 13, 15);**

**$evencount = 0;**

**$oddcount = 0;**

**foreach ($number as $num) {**

**if ($num % 2 == 0) {**

**$evencount++;**

**} else {**

**$oddcount++;**

**}**

**}**

**echo "Total even numbers: " . $evencount . "<br>";**

**echo "Total odd numbers: " . $oddcount . "<br>";**

**?>**

1. Create an associative array for user details (name, email, age) and display them.

**<?php**

**$user=array(**

**"name"=>"shruti jain",**

**"email"=>"shruti@gmail.com",**

**"age"=>21**

**);**

**echo "Name: " . $user["name"] . "<br>";**

**echo "Email: " . $user["email"] . "<br>";**

**echo "Age: " . $user["age"] . "<br>";**

**?>**

1. Write a script to shift all zero values to the bottom of an array.

**<?php**

**$array = [0, 5, 0, 3, 0, 1];**

**$nonZero = [];**

**$zeroCount = 0;**

**foreach ($array as $value) {**

**if ($value == 0) {**

**$zeroCount++;**

**} else {**

**$nonZero[] = $value;**

**}**

**}**

**for ($i = 0; $i < $zeroCount; $i++) {**

**$nonZero[] = 0;**

**}**

**print\_r($nonZero);**

**?>**

1. **PHP Date-Time Function**

Write a script to display the current date and time in different formats.

**<?php**

**date\_default\_timezone\_set("Asia/Kolkata");**

**echo "<h2>Current Date and Time in Different Formats:</h2>";**

**echo "Format 1: " . date("d-m-Y H:i:s") . "<br>";**

**echo "Format 2: " . date("m/d/Y") . "<br>";**

**echo "Format 3: " . date("l, d F Y") . "<br>";**

**echo "Format 4: " . date("h:i A") . "<br>";**

**echo "Format 5: " . date("c") . "<br>";**

**echo "Format 6: " . date("r") . "<br>";**

**?>**

. **Header Function**

THEORY EXERCISE:

What is the header function in PHP and how is it used?

The header() function in PHP is used to send raw HTTP headers to the browser **before any output** is sent.

It is mainly used

· Redirecting pages

· Setting content type (like JSON, PDF)

· Controlling caching

· Forcing file downloads

LAB EXERCISE:  Redirect users to another page using the header() function.

<?php

header("Location: page2.php");

exit();

?>

<html>

<head>

<title>Redirected Page</title>

</head>

<body>

<h1>You have been redirected successfully!</h1>

<p>This is Page 2.</p>

</body>

</html>

**Include and Require**

THEORY EXERCISE:

Explain the difference between include and require in PHP.

1. **Include:**

· It is used to include a file in a PHP script.

· If the file is **not found**, PHP shows a **warning**, but the script **continues to run**.

· Best used for **optional** files (like sidebar, ads, or footer).

1. **require:**

· It is also used to include a file in a PHP script.

· If the file is **not found**, PHP shows a **fatal error**, and the script **stops running.**

· Best used for **essential** files (like database connection or configuration files).

LAB EXERCISE: 

Use include and require to insert common header and footer files into multiple PHP pages.

header.php

**<html>**

**<head>**

**<title>My Website</title>**

**</head>**

**<body>**

**<h1>Welcome to My Website</h1>**

**<nav>**

**<a href="home.php">Home</a> |**

**<a href="about.php">About</a>**

**</nav>**

**<hr>**

footer.php

**<hr>**

**<p>© 2025 Shruti Jain. All rights reserved.</p>**

**</body>**

**</html>**

home.php

**<?php**

**require("header.php");**

**?>**

**<h2>This is the Home Page</h2>**

**<p>Welcome to our website!</p>**

**<?php**

**include("footer.php");**

**?>**

**About.php**

**<?php**

**require("header.php");**

**?>**

**<h2>About Us</h2>**

**<p>This is the about page of the website.</p>**

**<?php**

**include("footer.php");**

**?>**

**OUTPUT:**

**Welcome to My Website**

**Home | About**

**------------------------**

**This is the Home Page**

**Welcome to our website!**

**------------------------**

**© 2025 Shruti Jain. All rights reserved.**

1. Practical Example:

Calculator, Factorial, String Reverse LAB EXERCISE:

1. Calculator: Create a calculator using user-defined

**<?php**

**function add($a, $b) {**

**return $a + $b;**

**}**

**function subtract($a, $b) {**

**return $a - $b;**

**}**

**function multiply($a, $b) {**

**return $a \* $b;**

**}**

**function divide($a, $b) {**

**if ($b == 0) {**

**return "Cannot divide by zero!";**

**}**

**return $a / $b;**

**}**

**$num1 = 10;**

**$num2 = 5;**

**$operator = "+";**

**echo "<h2>Calculator Output</h2>";**

**switch ($operator) {**

**case "+":**

**echo "Addition: " . add($num1, $num2);**

**break;**

**case "-":**

**echo "Subtraction: " . subtract($num1, $num2);**

**break;**

**case "\*":**

**echo "Multiplication: " . multiply($num1, $num2);**

**break;**

**case "/":**

**echo "Division: " . divide($num1, $num2);**

**break;**

**default:**

**echo "Invalid operator";**

**}**

**?>**

(2)Factorial: Write a function that finds the factorial of a number using

**<?php**

**function factorial($n) {**

**if ($n == 0 || $n == 1) {**

**return 1;**

**} else {**

**return $n \* factorial($n - 1);**

**}**

**}**

**$num = 5;**

**echo "Factorial of $num is: " . factorial($num);**

**?>**

(3)String Reverse: Reverse a string without using built-in functions.

**<?php**

**function reverseString($str) {**

**$reversed ="";**

**$length = strlen($str);**

**for ($i =$length -1; $i>=0;$i--) {**

**$reversed.=$str[$i];**

**}**

**return $reversed;**

**}**

**$input ="shruti";**

**echo"original string: $input <br>";**

**echo"Reversed string:".reverseString($input);**

**?>**

Download File: Create a button that allows users to download a file.

**PHP Expressions, Operations, and String Functions THEORY EXERCISE:**

1. Explain what PHP expressions are and give examples of arithmetic and logical operations.

In PHP, an expression is a combination of **values, variables, operators, and functions** that gives a **single result**.

Expressions are used to perform tasks like **calculations**, **comparisons**, and **decision-making** in programs. For example, $a + $b is an expression that adds two values.

They are very important in PHP because they help the program to **process data and perform actions**.

1. Arithmetic Expressions:

Arithmetic expressions use **arithmetic operators** to perform mathematical operations such as addition, subtraction, multiplication, etc.

**<?php**

**$a = 20;**

**$b = 4;**

**echo "Addition: " . ($a + $b); // 24**

**echo "Subtraction: " . ($a - $b); // 16**

**echo "Multiplication: " . ($a \* $b); // 80**

**echo "Division: " . ($a / $b); // 5**

**echo "Modulus: " . ($a % $b); // 0**

**?>**

1. **Logical Expressions:**

Logical expressions are used to **evaluate conditions** and return a boolean value (true or false).

**<?php**

**$x = true;**

**$y = false;**

**echo "AND: " . ($x && $y); // false (0)**

**echo "OR: " . ($x || $y); // true (1)**

**echo "NOT: " . (!$x); // false (0)**

**?>**

LAB EXERCISE: 

Write a script to perform various string operations like concatenation, substringextraction, and string length determination.

**<?php**

**$firstName = "shruti";**

**$lastName = "jain";**

**$fullName = $firstName . " " . $lastName;**

**echo "Full Name (concatenation): " . $fullName . "<br>";**

**$sub = substr($fullName, 0, 6);**

**echo "Substring (First 6 letters): " . $sub . "<br>";**

**$length = strlen($fullName);**

**echo "Length of Full Name: " . $length . "<br>";**

**?>**