**Module1–CorePHP**

**1. Structure of a PHP Script and Embedding PHP in HTML**

Structure of a PHP Script:

- A PHP script starts with <?php and ends with ?>.

- It contains:

1. PHP Tags: Code is written inside <?php ... ?>.

2. Statements & Functions: Like echo, if-else, loops, etc.

3. Comments: // for single-line, /\* for multi-line.

\*/

/\* Example: \*/

echo "Hello, World!";

<!DOCTYPE html>

<html>

<body>

<h1>Welcome</h1>

<p>Today's date is: <?php echo date("Y-m-d"); ?></p>

</body>

</html>

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

**Answer:**

PHP Variable Naming Rules:

Must start with $. ($name)

Begin with a letter or \_ (Not a number).

Can contain letters, numbers, and \_ (No spaces or -).

Case-sensitive ($User ≠ $user).

Cannot use PHP reserved keywords ($echo ❌).

✅ Example: \*/

$valid\_name = "Zaid"; // ✅ Valid

// $1user = "John"; ❌ Invalid

### **PHP Variables and Their Scope**

#### **Variable in PHP :** A **variable** in PHP is used to store data (like numbers, strings, etc.). It starts with a **$** sign.

Example:<?php

$name = "Zaid";

echo $name; // Output: Zaid ?>

**✅ Types of Variable Scope in PHP:**

**Local Scope** – Defined inside a function, accessible only inside it.  
function test() {

$x = 10; // Local variable

echo $x; }

**test(); // Output: 10**

**Global Scope – Defined outside a function, not accessible inside it unless global is used.**

$y = 20; // Global variable

function demo() { global $y; echo $y; }

**demo(); // Output: 20**

**3. What Are Superglobal variables in PHP?List At Least five super global arrays and their use ?**

### **Superglobal Variables in PHP** Superglobal variables are **predefined global arrays** in PHP that can be accessed from anywhere in the script, including functions and classes. They are mainly used to **store and retrieve** data like form inputs, session details, and server information.

1️⃣ **$\_GET** → Stores **URL parameters** (sent via query strings).

* Example: example.com?name=Zaid & age=22

echo $\_GET['name']; // Output: Zaid

echo $\_GET['age']; // Output: 22

2️⃣ **$\_POST** → Stores **form data** sent via POST method.

echo $\_POST['email']; // Access submitted email from form

3️⃣ **$\_SESSION** → Stores **user session data** (remains until session ends).

session\_start();

$\_SESSION['user'] = "Zaid"; // Store user info

echo $\_SESSION['user']; // Output: Zaid

4️⃣ **$\_COOKIE** → Stores **user data in cookies** (remains after closing browser). echo $\_COOKIE['theme']; // Access saved theme preference

5️⃣ **$\_SERVER** → Stores **server and request information**.

echo $\_SERVER['HTTP\_HOST']; // Get domain name echo $\_SERVER['REQUEST\_METHOD']; // Check request type (GET/POST)

**<Step 1: Create Tables**

**CREATE TABLE customers (**

**id INT PRIMARY KEY AUTO\_INCREMENT,**

**name VARCHAR(100),**

**email VARCHAR(100),**

**city VARCHAR(50)**

**);**

**CREATE TABLE orders (**

**id INT PRIMARY KEY AUTO\_INCREMENT,**

**customer\_id INT,**

**product VARCHAR(100),**

**quantity INT,**

**price DECIMAL(10,2),**

**order\_date DATE,**

**FOREIGN KEY (customer\_id) REFERENCES customers(id)**

**);**

**Step 2: Insert Data**

**INSERT INTO customers (name, email, city) VALUES**

**('John Doe', 'john@example.com', 'New York'),**

**('Alice Smith', 'alice@example.com', 'Los Angeles'),**

**('Bob Johnson', 'bob@example.com', 'Chicago');**

**INSERT INTO orders (customer\_id, product, quantity, price, order\_date)**

**VALUES**

**(1, 'Laptop', 1, 1000.00, '2024-02-01'),**

**(2, 'Phone', 2, 500.00, '2024-02-03'),**

**(3, 'Tablet', 1, 300.00, '2024-02-05');**

**Step 3: Perform SQL Queries**

**1. Select All Customers**

**SELECT \* FROM customers;**

**2. Select Orders Where Price > 400**

**SELECT \* FROM orders WHERE price > 400;**

**3. Update Customer Email**

**UPDATE customers SET email = 'newemail@example.com' WHERE id = 1;**

**4. Delete an Order**

**DELETE FROM orders WHERE id = 3;**

**5. Select Orders with LIKE (Find Products Containing 'Phone')**

**SELECT \* FROM orders WHERE product LIKE '%Phone%';**

**6. Group Orders by Customer and Count Orders**

**SELECT customer\_id, COUNT(\*) AS total\_orders**

**FROM orders GROUP BY customer\_id;**

**7. Use HAVING to Show Customers with More Than 1 Order**

**SELECT customer\_id, COUNT(\*) AS total\_orders**

**FROM orders GROUP BY customer\_id HAVING total\_orders > 1;**

**8. Limit Results (Show Only 2 Customers)**

**SELECT \* FROM customers LIMIT 2;**

**9. Offset Results (Skip First Record, Show Next 2)**

**SELECT \* FROM customers LIMIT 2 OFFSET 1;**

**10. Subquery: Select Customers Who Placed Orders**

**SELECT \* FROM customers WHERE id IN (SELECT customer\_id FROM orders);**

**11. Use AND, OR, NOT**

**SELECT \* FROM orders WHERE quantity > 1 AND price < 800;**

**SELECT \* FROM customers WHERE city = 'New York' OR city = 'Chicago';**

**SELECT \* FROM customers WHERE NOT city = 'Los Angeles';**

**6.Conditions,Events,andFlows**

Conditional statements **decision-making** ke liye use hote hain. Yeh check karte hain **agar (if), warna (else), ya multiple conditions (elseif)**.

### **Types of Conditional Statements:**

**if Statement – Condition true ho to code chalega.**  
$age = 18;

if ($age >= 18) {

echo "You can vote.";

}

**if-else Statement – True hone pe ek code, warna dusra code.**  
if ($age >= 18) {

echo "You can vote.";

} else {

echo "You cannot vote.";

}

**if-elseif-else Statement – Multiple conditions check karta hai.**if ($age < 18) {

echo "Too young to vote.";

} elseif ($age == 18) {

echo "Just eligible to vote!";

} else {

echo "You can vote.";

}

**switch Statement – Jab multiple cases compare karne ho.**$day = "Monday";

switch ($day) {

case "Monday":

echo "Start of the week!";

break;

case "Friday":

echo "Weekend is near!";

break;

default:

echo "Normal day.";

}

✅ **Conditional statements ka use logic implement karne aur decisions lene ke liye hota hai!** 🚀

**10.Loops: Do-While ,ForEach, ForLoop**

### **Difference Between For, Foreach, and Do-While Loops in PHP**

| **Loop Type** | **Usage & Condition** | **Best For** |
| --- | --- | --- |
| **For Loop** | Runs when the number of iterations is known. Checks condition before execution. | Fixed number of iterations |
| **Foreach Loop** | Iterates over arrays, no explicit condition needed. | Arrays & objects |
| **Do-While Loop** | Executes at least once, checks condition after execution. | At least one guaranteed iteration |

### 

### **Examples:**

#### **For Loop:**

for ($i = 1; $i <= 5; $i++) { echo $i . " "; }

**Output:** 1 2 3 4 5

#### **Foreach Loop:**

$fruits = ["Apple", "Banana", "Cherry"];

foreach ($fruits as $fruit) { echo $fruit . " "; }

**Output:** Apple Banana Cherry

#### **Do-While Loop:**

php

CopyEdit

$i = 1;

do { echo $i . " "; $i++; } while ($i <= 5);

**Output: 1 2 3 4 5**

**For** → Fixed count | **Foreach** → Arrays | **Do-While** → At least once execution

**11.PHP Array and Array Functions**

**Arrays in PHP** An **array** in PHP is a data structure that stores multiple values in a **single variable**. Each value in an array is accessed using an index or key.

### **Types of Arrays in PHP**

1. **Indexed Array (Numeric Array)**
   * Stores values with numeric indexes starting from 0.
   * Used for storing a list of values.

**Example:**  
$fruits = ["Apple", "Banana", "Cherry"];

echo $fruits[0]; // Output: Apple

1. **Associative Array**
   * Uses **named keys** instead of numeric indexes.
   * Helps in mapping meaningful keys to values.

**Example:**  
$person = ["name" => "John", "age" => 25, "city" => "New York"];

echo $person["name"]; // Output: John

1. **Multidimensional Array**
   * An array inside another array (nested structure).
   * Used for storing complex data like tables.

**Example:**  
$students = [

["John", 25, "New York"],

["Alice", 22, "London"]

];

echo $students[0][0]; // Output: John

### **Summary:**

| **Array Type** | **Description** | **Example** |
| --- | --- | --- |
| **Indexed** | Uses numeric indexes | $arr = ["A", "B", "C"]; |
| **Associative** | Uses named keys | $arr = ["name" => "John", "age" => 25]; |
| **Multidimensional** | Nested arrays | $arr = [["A", 1], ["B", 2]]; |

**13.HeaderFunction What is the heade rfunction in PHP and how is it used?**

### **Header Function in PHP**

The header() function in PHP is used to send **raw HTTP headers** to the browser before any output is sent. It allows control over **redirections, content type, caching, and more**.

### **Uses of header() Function:**

**Redirect to Another Page**  
header("Location: homepage.php");

**14. Explain the difference between include and require PHP ?**

Both include and require are used to import external PHP files into a script, but they behave differently when an error occurs.

1. include

* Includes the file, but if the file is missing or has an error, it **shows a warning and continues execution**.
* Used when the file is optional.

**Example:**include "header.php";

echo "Page content continues even if header.php is missing.";

2. require

* Includes the file, but if the file is missing or has an error, it **shows a fatal error and stops execution**.
* Used when the file is essential for the script to work.

**Example:**require "config.php";

echo "This will not execute if config.php is missing.";

**16.Explain what PHP expressions are and give examples of arithmetic and logical operations?**

**PHP Expressions :** PHP expressions are combinations of values, variables, and operators that produce a result.

1️⃣ Arithmetic Expressions (Math operations)

📌 Used for calculations

$a = 10;

$b = 5;

$sum = $a + $b; // Addition (10 + 5 = 15)

$sub = $a - $b; // Subtraction (10 - 5 = 5)

$mul = $a \* $b; // Multiplication (10 \* 5 = 50)

$div = $a / $b; // Division (10 / 5 = 2)

$mod = $a % $b; // Modulus (Remainder) (10 % 5 = 0)

2️⃣ Logical Expressions (True/False conditions) 📌 Used for decision making

$x = true;

$y = false;

$and = $x && $y; // AND (Both true? ❌ false)

$or = $x || $y; // OR (At least one true? ✅ true)

$not = !$x; // NOT (Opposite of true? ❌ false)