Module 5 – Core PHP

PHP Syntax

THEORY EXERCISE:

• Discuss the structure of a PHP script and how to embed PHP in HTML.

* A **PHP script** starts with the opening <?php tag and ends with the closing ?> tag. Everything between these tags is interpreted as PHP code.

**. mbedding PHP in HTML**: PHP can be embedded within HTML to create dynamic web pages. The PHP code is executed on the server, and the output is sent to the client's browser as plain HTML.

• What are the rules for naming variables in PHP? LAB EXERCISE:

**2. Rules for Naming Variables in PHP:**

* Variables in PHP always start with a dollar sign ($), followed by the name of the variable.
* **Rules**:
  1. The variable name must start with a letter or an underscore (\_).
  2. The name can contain letters, numbers, and underscores, but **cannot start with a number**.
  3. Variable names are **case-sensitive** ($name and $Name are different variables).
  4. Reserved keywords (e.g., class, function) cannot be used as variable names.

**Examples**:

* 1. Valid: $name, $\_name, $name123
  2. Invalid: $123name, $na-me, $class

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

****

**3. PHP Variables**

**THEORY EXERCISE:**

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

**What are Variables?**

**In PHP, variables are used to store data. They act like containers that hold values (like numbers, strings, etc.). Key features of variables in PHP include:**

1. **Dollar Sign ($): A variable name in PHP always starts with a $.**

**2.Dynamic Typing: You don’t need to declare the type of a variable in PHP; it is determined automatically based on the value assigned to it.**

**3. Case-Sensitive: Variable names in PHP are case-sensitive.**

**Scope of Variables**

**The scope of a variable refers to the part of the script where the variable can be accessed or used. PHP has three main types of variable scope:**

1. **Global Scope:**
   * **Variables declared outside of a function are considered global.**
   * **These variables cannot be accessed directly inside a function unless explicitly declared as global**

**2.Local Scope:**

. Variables declared inside a function are local to that function and cannot be accessed outside it

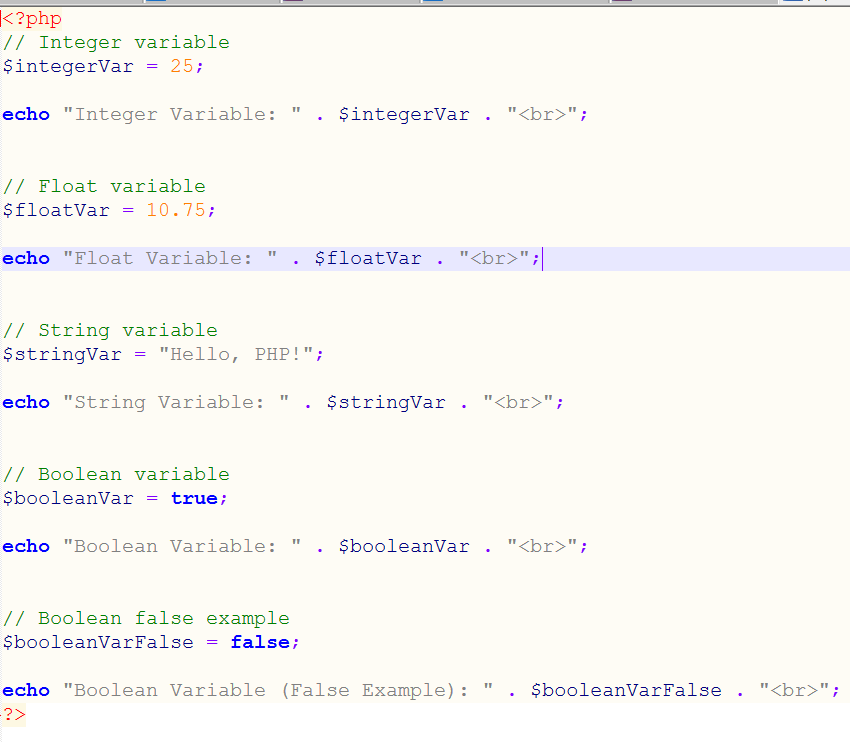
**3.Static Scope:**

* Static variables retain their value even after the function execution ends. They are initialized only once.

By understanding variables and their scope, you can control how data is accessed and manipulated within your PHP scripts effectively.

**LAB EXERCISE:**

**• Create a PHP script to declare and initialize different types of variables (integer, float, string, boolean). Display them using echo.**

****

**4. Super Global Variables**

**THEORY EXERCISE:** •

**. What are super global variables in PHP? List at least five super global arrays and their use.**

**Super Global Variables in PHP are predefined variables that are always accessible, regardless of scope. These variables are built-in arrays provided by PHP and can be accessed from any part of the script (inside functions, classes, etc.) without requiring a global declaration.**

**Characteristics of Super Global Variables:**

1. **They are automatically global in scope.**
2. **Used to access data from forms, URLs, sessions, cookies, and the server environment.**

**1.$\_GET**

* **Use: To collect data sent via HTTP GET method (query strings in the URL)**

**2.$\_POST**

**.Use**: To collect data sent via HTTP POST method (commonly used in form submissions).

**3**. **$\_REQUEST**

* **Use**: To collect data from both $\_GET, $\_POST, and $\_COOKIE. However, it is less secure and should be used cautiously.

**4.$\_SESSION**

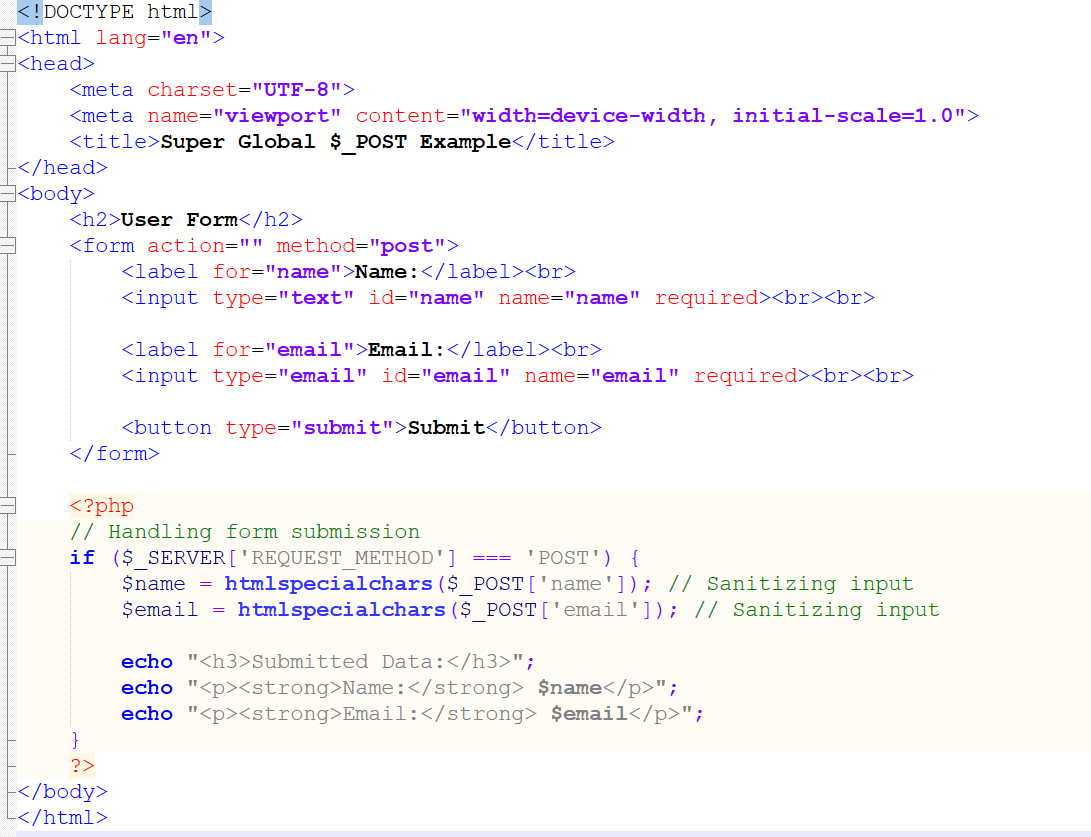
* **Use**: To store and access session variables. Data stored in a session is preserved across multiple page requests by the same user.

**5.$\_SERVER**

* **Use**: To access server-related information, such as headers, paths, and script locations.

**LAB EXERCISE:**

• **Create a form that takes a user's name and email. Use the $\_POST super global to display the entered data.**

****

**5. Practical Example: Multiple Tables and SQL Queries**

**LAB EXERCISE:**

**• Create multiple tables and perform queries using:**

**o SELECT, UPDATE, DELETE, INSERT**

**o WHERE, LIKE, GROUP BY, HAVING**

**o LIMIT, OFFSET, Subqueries, AND, OR, NOT, IN**

**1.create tables**

**CREATE TABLE Students (**

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

**Name VARCHAR(100),**

**Age INT,**

**Gender VARCHAR(10),**

**CourseID INT**

**);**

**INSERT INTO Students (Name, Age, Gender, CourseID)**

**VALUES**

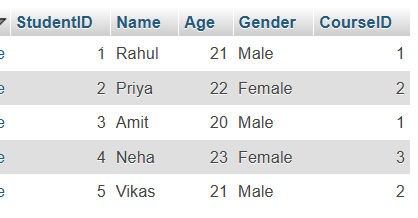
**('Rahul', 21, 'Male', 1),**

**('Priya', 22, 'Female', 2),**

**('Amit', 20, 'Male', 1),**

**('Neha', 23, 'Female', 3),**

**('Vikas', 21, 'Male', 2);**

****

**CREATE TABLE Courses (**

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

**CourseName VARCHAR(100),**

**Duration INT,**

**Fees DECIMAL(10, 2)**

**);**

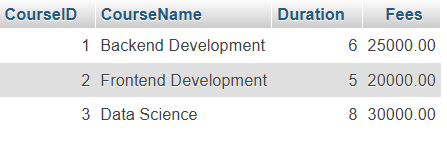
**INSERT INTO Courses (CourseName, Duration, Fees)**

**VALUES**

**('Backend Development', 6, 25000.00),**

**('Frontend Development', 5, 20000.00),**

**('Data Science', 8, 30000.00);**

****

**3.select**

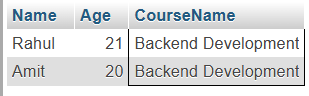
**SELECT \* FROM Students;**

**SELECT S.Name, S.Age, C.CourseName**

**FROM Students S**

**JOIN Courses C ON S.CourseID = C.CourseID**

**WHERE C.CourseName = 'Backend Development';**

****

**4.update**

**UPDATE Students**

**SET Age = 22**

**WHERE Name = 'Rahul';**

**UPDATE Courses**

**SET Fees = Fees + 2000**

**WHERE CourseName = 'Frontend Development';**

**5.delete queries**

**DELETE FROM Students**

**WHERE Name = 'Vikas';**

**DELETE FROM Courses**

**WHERE CourseName = 'Data Science';**

**6.WHERE,LIKE,AND,OR,NOT**

**1.WHERE**

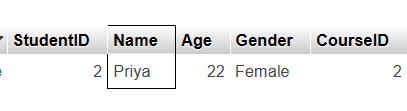
**SELECT \* FROM Students**

**WHERE Age = 21;**

**2.LIKE**

**SELECT \* FROM Students**

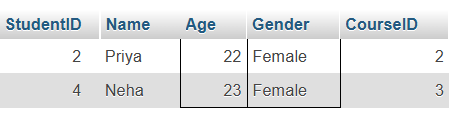
**WHERE Name LIKE 'Pri%';**

****

**3.AND,OR**

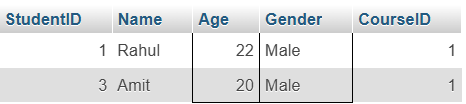
**SELECT \* FROM Students**

**WHERE Age > 20 AND Gender = 'Female';**

****

**SELECT \* FROM Students**

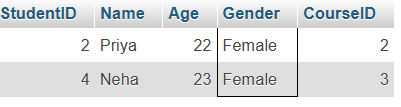
**WHERE Age < 21 OR Gender = 'Male';**

****

**4.NOT**

**SELECT \* FROM Students**

**WHERE NOT Gender = 'Male';**

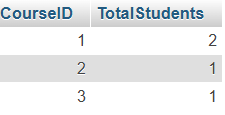
****

**7..GROUP BY, HAVING**

**SELECT CourseID, COUNT(\*) AS TotalStudents**

**FROM Students**

**GROUP BY CourseID;**

****

**SELECT CourseID, COUNT(\*) AS TotalStudents**

**FROM Students**

**GROUP BY CourseID**

**HAVING COUNT(\*) > 1;**

**8.LIMIT,OFFSET**

**SELECT \* FROM Students**

**LIMIT 2;**

**SELECT \* FROM Students**

**LIMIT 2 OFFSET 1;**

**9.SUBQUERIES**

**SELECT S.Name, C.CourseName**

**FROM Students S**

**JOIN Courses C ON S.CourseID = C.CourseID**

**WHERE C.Fees = (SELECT MAX(Fees) FROM Courses);**

**6. Conditions, Events, and Flows**

**THEORY EXERCISE:**

**• Explain how conditional statements work in PHP.**

**What is a Conditional Statement in PHP?**

**Last Updated : 16 Feb, 2024**

**A conditional statement in**[**PHP**](https://www.geeksforgeeks.org/php-tutorial/)**is a programming construct that allows you to execute different blocks of code based on whether a specified condition evaluates to true or false. It enables you to create dynamic and flexible code logic by controlling the flow of execution based on various conditions.**

**Conditional statements in PHP include:**

**PHP conditional statements, like if, else, elseif, and switch, control code execution based on specified conditions, enhancing code flexibility and logic flow.**

**if statement:**

[**PHP if statement**](https://www.geeksforgeeks.org/php-decision-making/)**executes a block of code if a specified condition is true.**

**if (condition) {**

**// Code to execute if condition is true**

**}**

**else statement:**

**PHP else statement executes a block of code if the condition of the preceding if statement evaluates to false.**

**if (condition) {**

**// Code to execute if condition is true**

**} else {**

**// Code to execute if condition is false**

**}**

**else if statement:**

**PHP else if statement allows you to evaluate multiple conditions sequentially and execute the corresponding block of code if any condition is true.**

**if (condition1) {**

**// Code to execute if condition1 is true**

**} elseif (condition2) {**

**// Code to execute if condition2 is true**

**} else {**

**// Code to execute if both condition1 and condition2 are false**

**}**

**switch statement:**

**PHP switch statement provides an alternative to multiple elseif statements by allowing you to test a variable against multiple possible values and execute different blocks of code accordingly.**

**switch (expression) {**

**case value1:**

**// Code to execute if expression equals value1**

**break;**

**case value2:**

**// Code to execute if expression equals value2**

**break;**

**// More cases can be added as needed**

**default:**

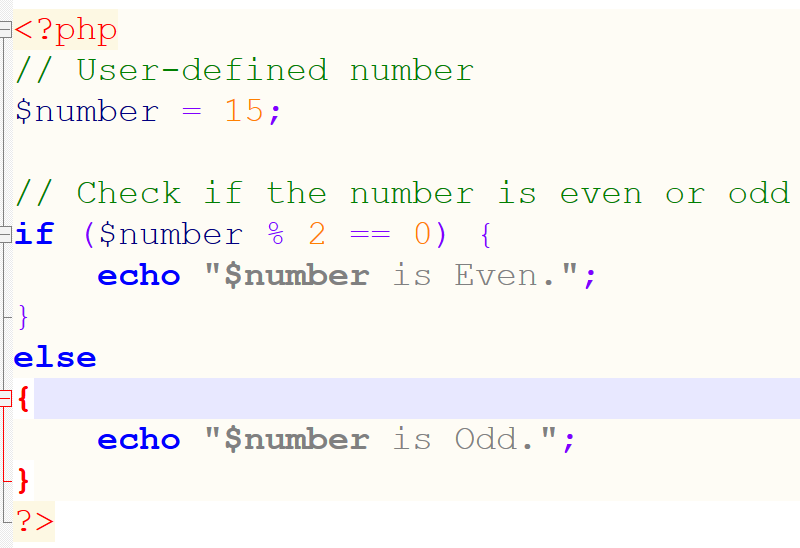
**// Code to execute if expression doesn't match any case**

**}**

**7. If Condition and If-Else If**

**LAB EXERCISE:**

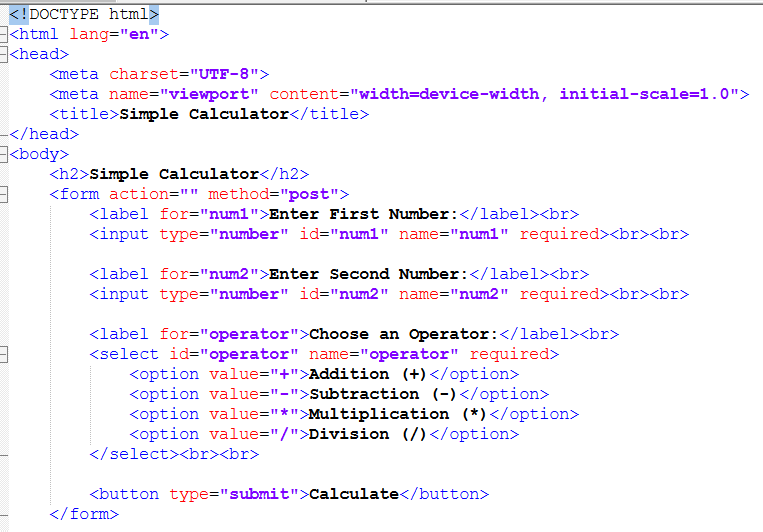
**• Write a PHP program to determine if a number is even or odd using if conditions**

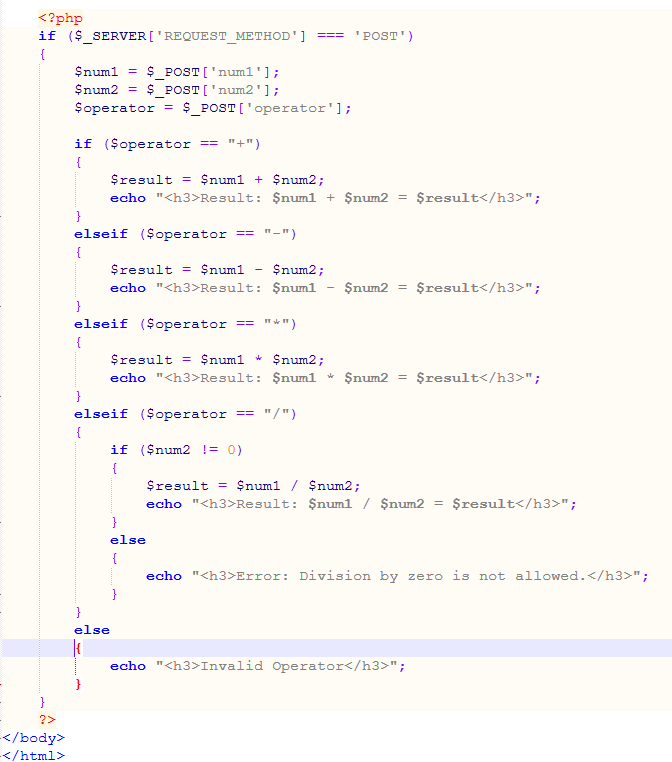
****

**8. Practical Example: Calculator and Day Finder**

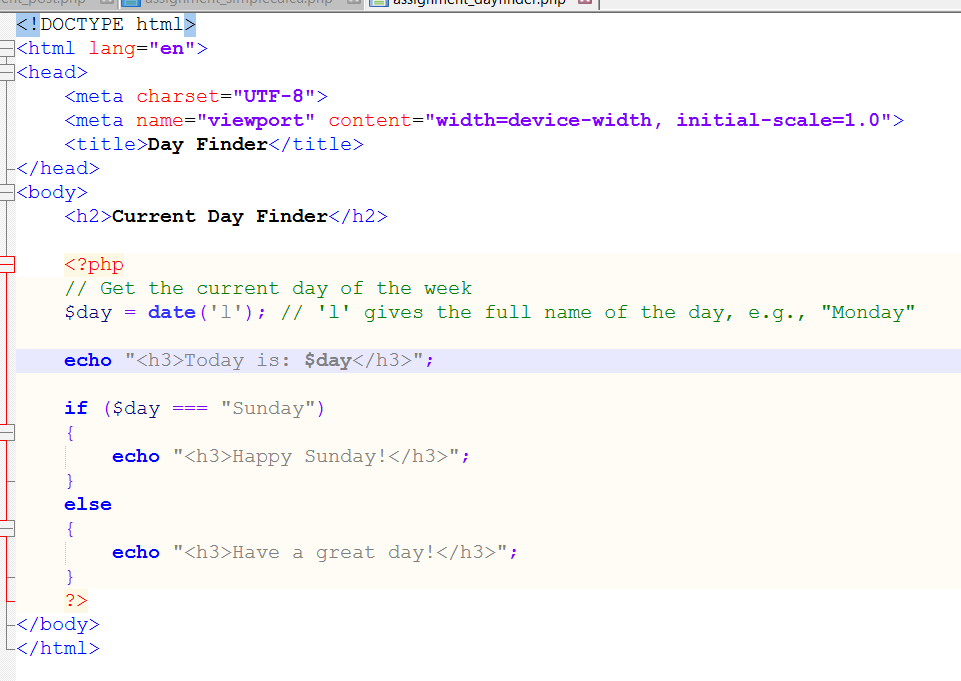
**LAB EXERCISE:**

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

****

****

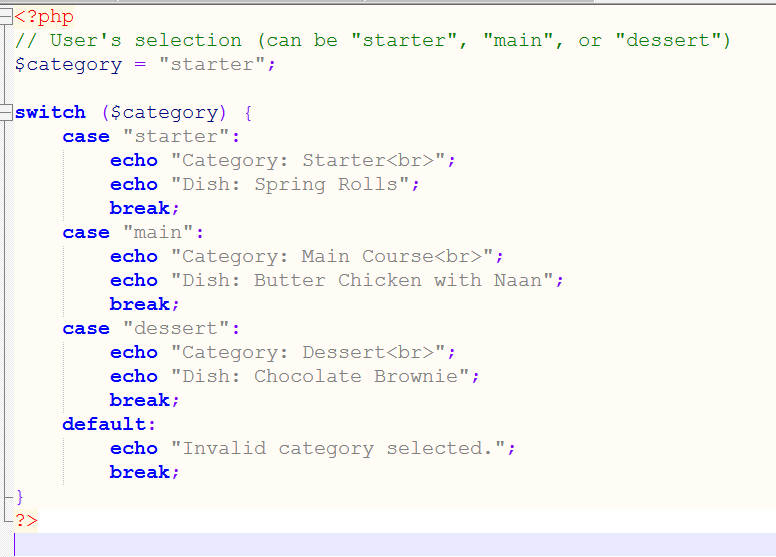
**2. Day Finder: Write a script that finds the current day. If it is Sunday, print "Happy Sunday."**

****

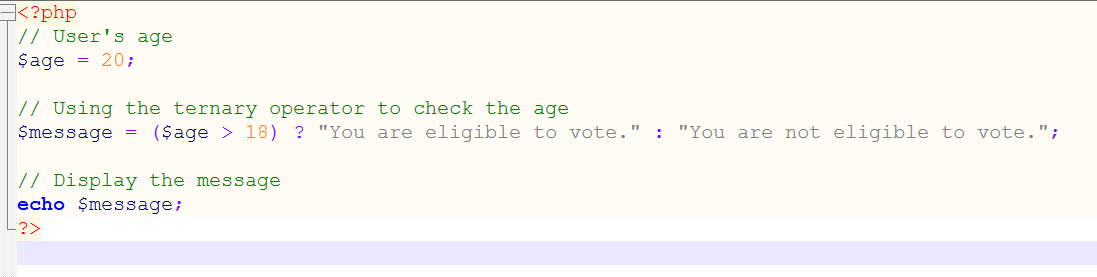
**9. Switch Case and Ternary Operator**

**LAB EXERCISE:**

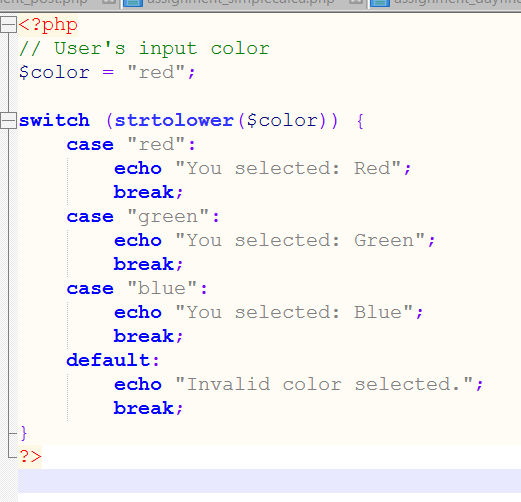
* 1. **Restaurant Food Category Program: Use a switch case to display the category (Starter/Main Course/Dessert) and dish based on user selection.**

****

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

****

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

****

**10. Loops: Do-While, For Each, For Loop**

**THEORY EXERCISE:**

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

* 1. **For Loop:**
* **Syntax: for (initialization; condition; increment/decrement) { //code }**
* **Used when the number of iterations is known**

**. Suitable for numeric ranges or fixed iteration counts.**

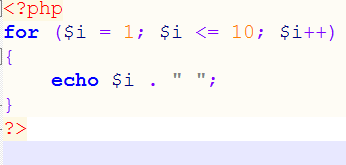
1. **Foreach Loop:**
2. **Syntax: foreach ($array as $value) { //code }**
3. **Iterates over elements of an array.**
4. **Best for working with arrays or collections.**
5. **Do-While Loop:**

**Executes the block of code at least once before checking the condition.**

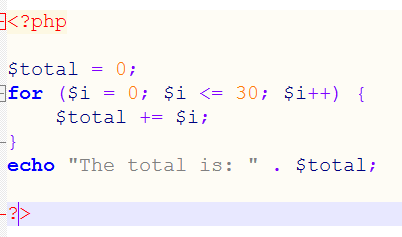
**Useful when the code must run at least once regardless of the condition.**

**LAB EXERCISE:**

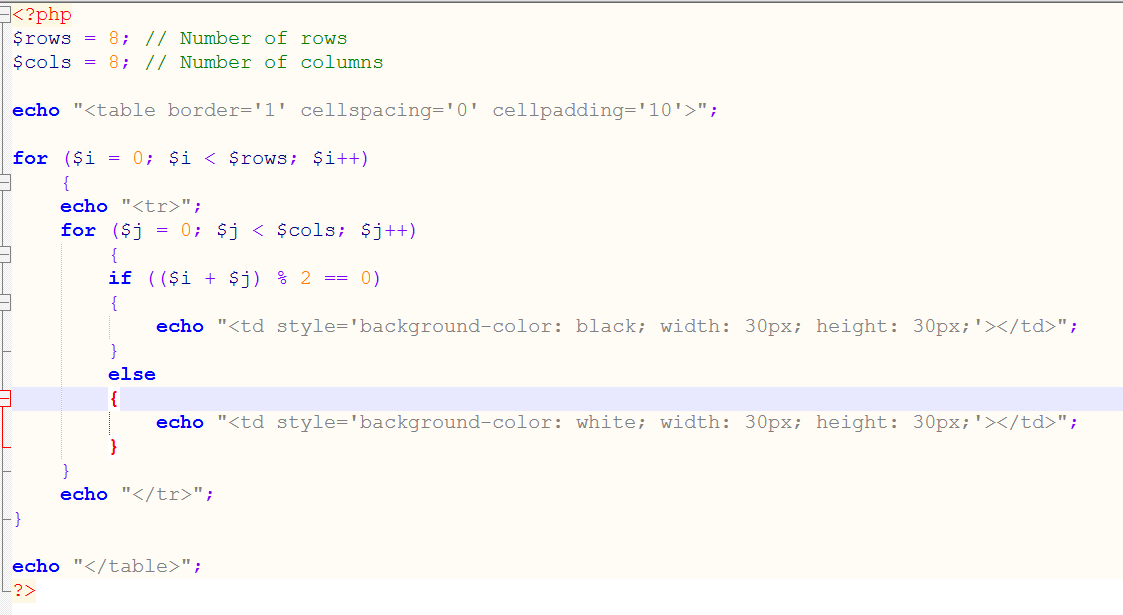
1. **For Loop: Write a script that displays numbers from 1 to 10 on a single line.**



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

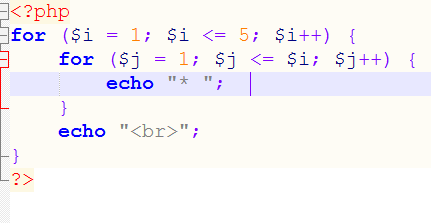
****

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

****

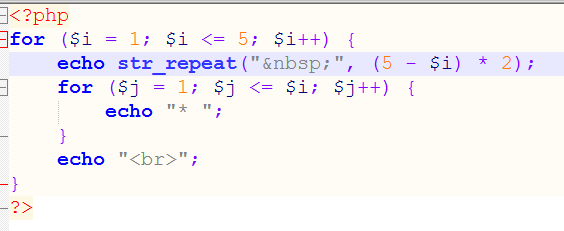
**4. Various Patterns: Generate different patterns using loops.**

**Right-Angled Triangle**

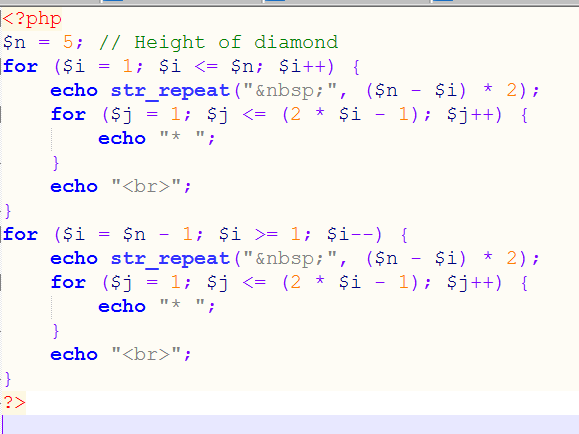
****

**Inverted Triangle Pattern:**

**Pyramid Pattern:**

****

**Diamond Pattern:**

****

**11. PHP Array and Array Functions**

**THEORY EXERCISE:**

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

**Arrays in PHP:**

**An array in PHP is a data structure that can hold multiple values in a single variable. Arrays allow storing, accessing, and manipulating collections of data in an organized way.**

**Types of Arrays:**

1. **Indexed Array:**
   * **Stores values with numeric indices starting from 0.**
2. **Associative Array:**

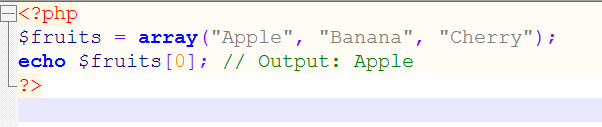
* **Stores values with named keys (key-value pairs).**

1. **Multidimensional Array:**

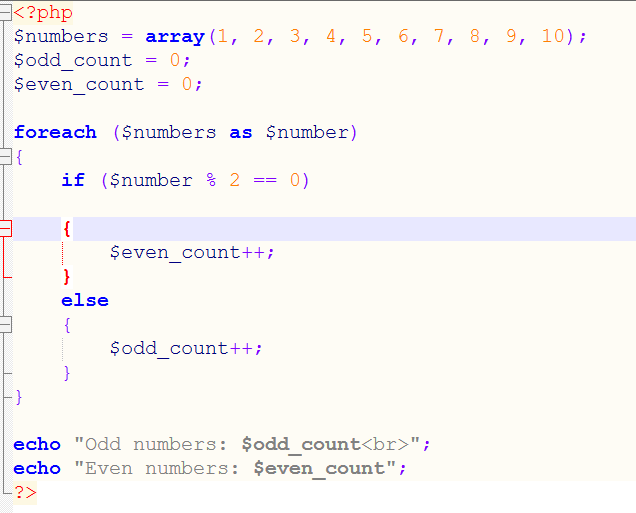
* **Contains arrays as its elements (array of arrays).**

**LAB EXERCISE:**

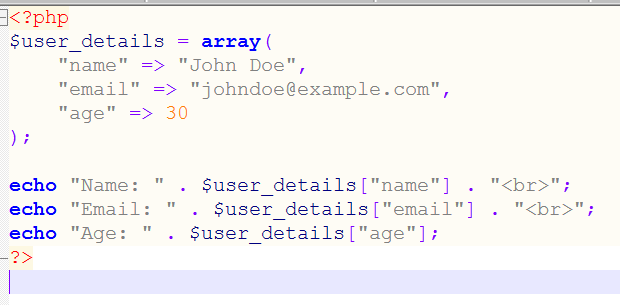
* 1. **Display the value of an array.**

****

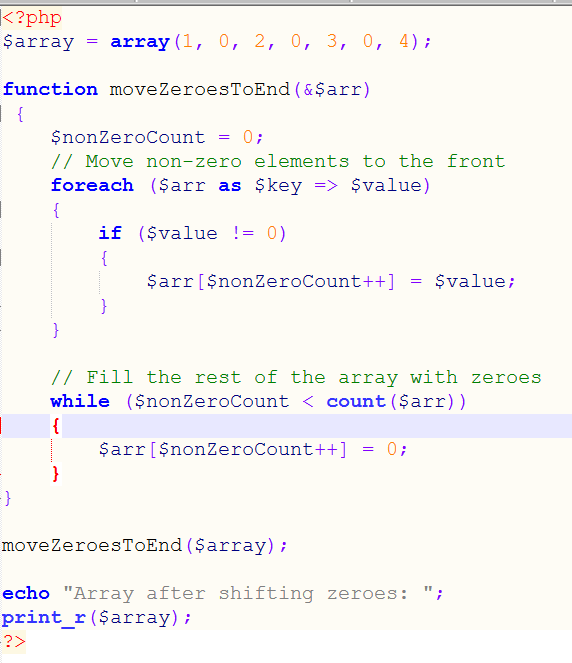
* 1. **Find and display the number of odd and even elements in an array.**

****

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

****

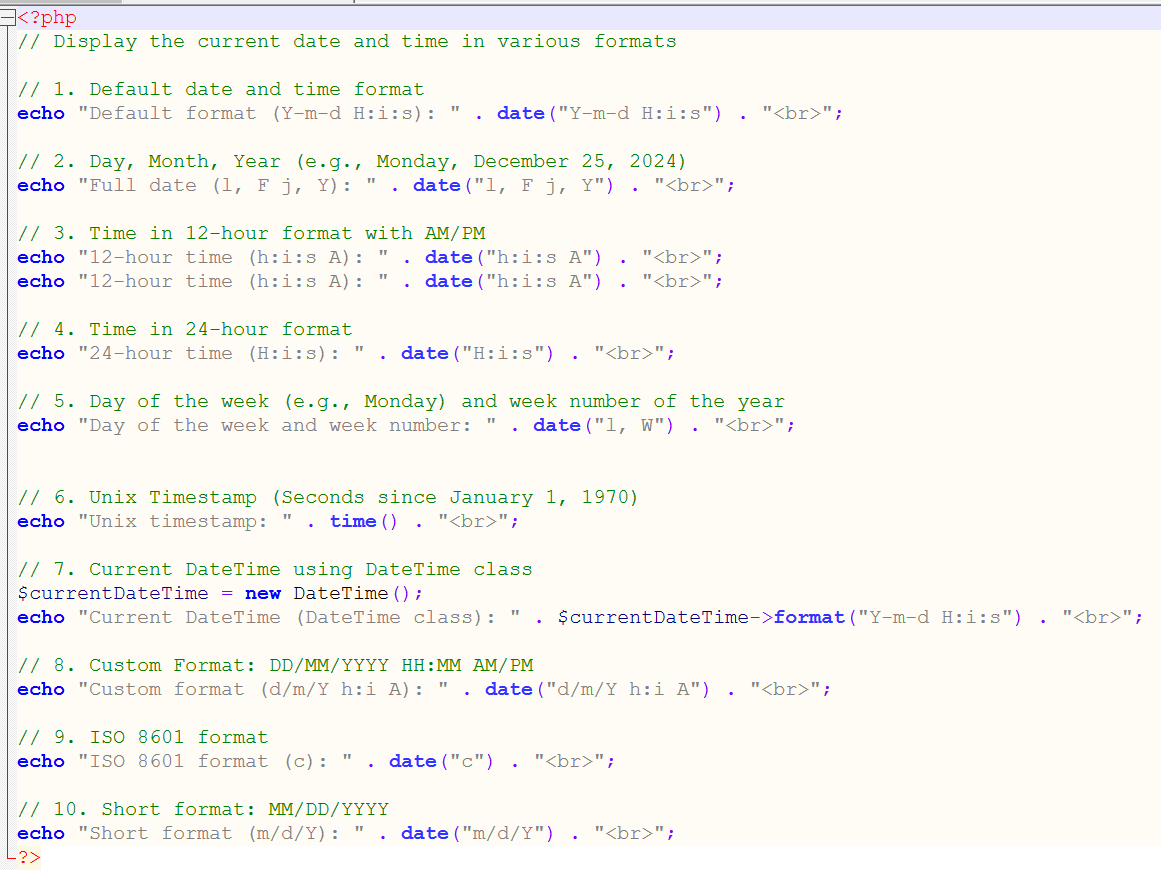
* 1. **Write a script to shift all zero values to the bottom of an array.**

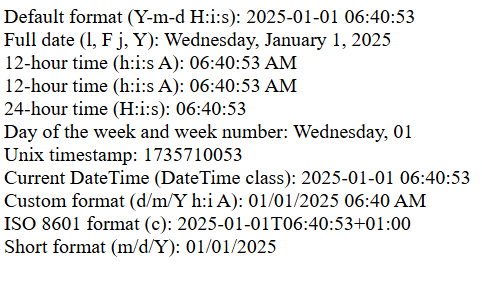
****

**12. PHP Date-Time Function**

**LAB EXERCISE:**

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

****

****

**13. 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 commonly used for tasks such as:**

1. **Redirecting to another page: Send a Location header to instruct the browser to navigate to a different page.**
2. **Setting content type: Define the content type of the response, e.g., text/html, application/json, etc.**
3. **Managing caching: Send headers to control caching behavior.**
4. **Custom headers: Add custom headers for advanced use cases.**

**Important Notes:**

* **The header() function must be called before any output (e.g., echo or HTML tags) is sent to the browser.**
* **Sending headers after output results in a "Headers already sent" error.**

**Syntax:**

**php**

**Copy code**

**header(string $header, bool $replace = true, int $response\_code = 0);**

**Parameters:**

* **$header: The header string to be sent.**
* **$replace: Whether to replace a previous similar header. Defaults to true.**
* **$response\_code: An optional HTTP response status code.**

**LAB EXERCISE:**

**• Redirect users to another page using the header() function.**

**<?php**

**// Redirect to another page using header()**

**header("Location: https://www.example.com");**

**exit; // Ensure no further code execution after redirection**

**?>**

**14. Include and Require**

**THEORY EXERCISE:**

**• Explain the difference between include and require in PHP.**

**Difference Between include and require in PHP**

**Both include and require are used to include external PHP files into a script. However, they handle errors differently:**

1. **include:**
   * **If the file cannot be included (e.g., file not found), PHP throws a warning, and the script continues to execute.**
   * **Used when the file is optional, and the application can still function without it.**

**require:**

* **If the file cannot be included, PHP throws a fatal error, and the script stops execution.**
* **Used for essential files that the application depends on.**
* **Key Differences:**

| **Feature** | **include** | **require** |
| --- | --- | --- |
| Error Handling | Generates a **warning** | Generates a **fatal error** |
| Script Execution | Continues execution | Stops execution |
| Use Case | Non-critical or optional files | Essential or required files |

**LAB EXERCISE:**

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

**15. Practical Example:**

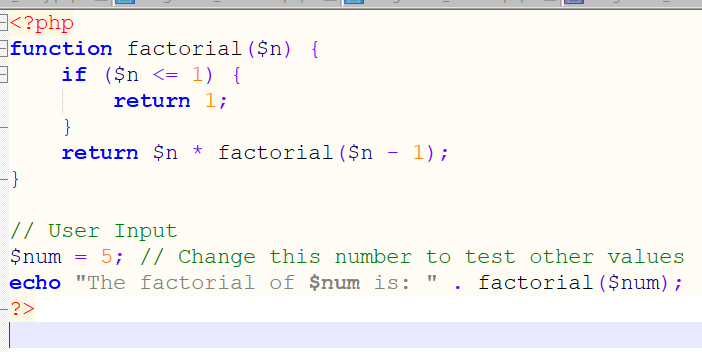
**Calculator, Factorial, String Reverse**

**LAB EXERCISE:**

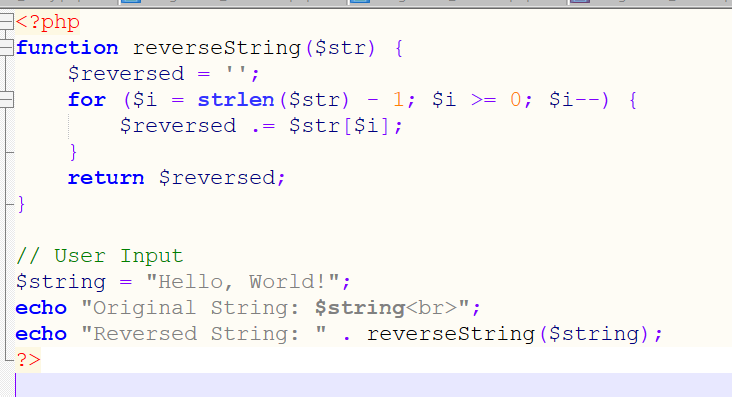
**Calculator: Create a calculator using user-defined functions.**

**?>** ****

**Factorial: Write a function that finds the factorial of a number using recursion.**

****

**String Reverse: Reverse a string without using built-in functions.**

****

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

**16. PHP Expressions, Operations, and String Functions**

**THEORY EXERCISE:**

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

**What are PHP Expressions?**

**PHP expressions are the smallest units of code that can be evaluated to produce a value. These include variables, constants, function calls, or a combination of operators and values.**

**Examples of Expressions:**

1. **Simple values:**

**$x = 5; // $x is an expression**

1. **Arithmetic operation:**

**$y = $x + 10; // $x + 10 is an expression**

1. **Logical operation:**

**$result = ($x > 0 && $x < 10); // Evaluates to true or false**

**Arithmetic Operations in PHP**

**PHP supports common arithmetic operators to perform mathematical calculations:**

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| **+** | **Addition** | **5 + 3** | **8** |
| **-** | **Subtraction** | **5 - 3** | **2** |
| **\*** | **Multiplication** | **5 \* 3** | **15** |
| **/** | **Division** | **6 / 3** | **2** |
| **%** | **Modulus (remainder)** | **5 % 3** | **2** |

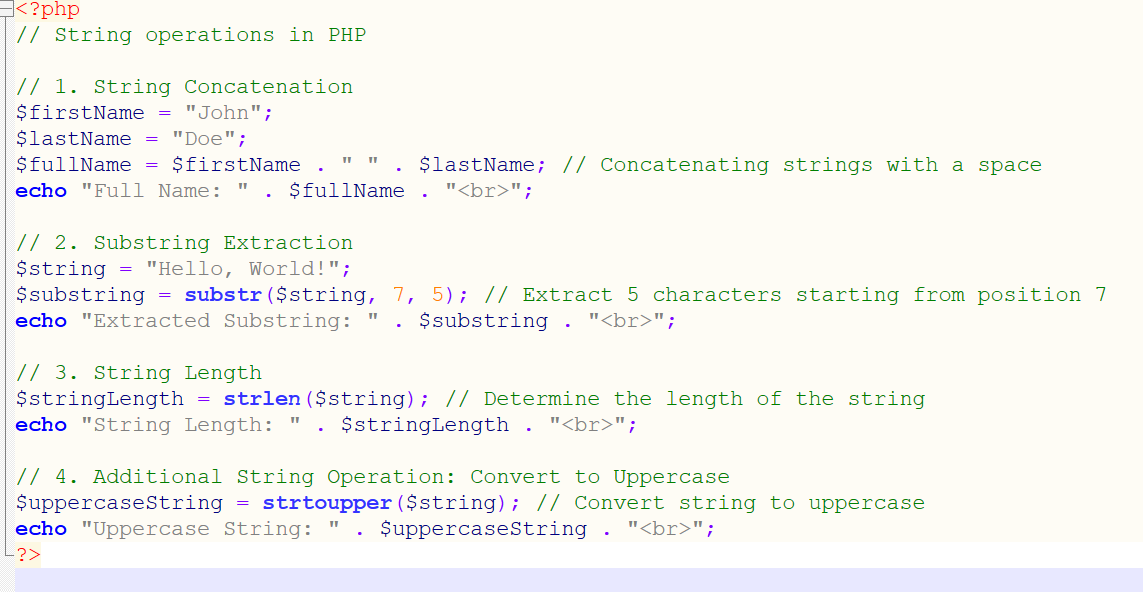
**Logical Operations in PHP**

**Logical operators are used to perform boolean logic:**

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| **&&** | **Logical AND** | **(true && false)** | **false** |
| **`** |  | **`** | **Logical OR** |
| **!** | **Logical NOT** | **!true** | **false** |

**LAB EXERCISE:**

**• Write a script to perform various string operations like concatenation, substring extraction, and string length determination.**

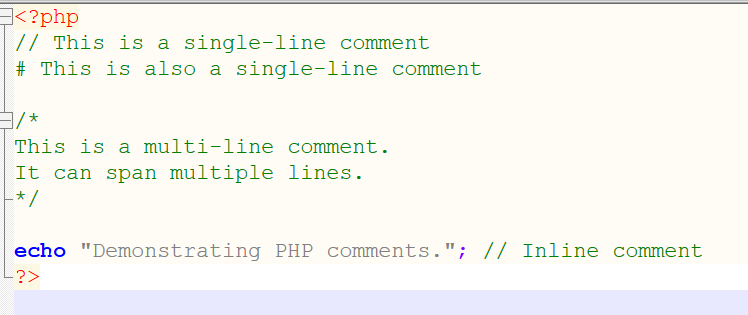
****

**Extra LAB EXERCISES for Core PHP**

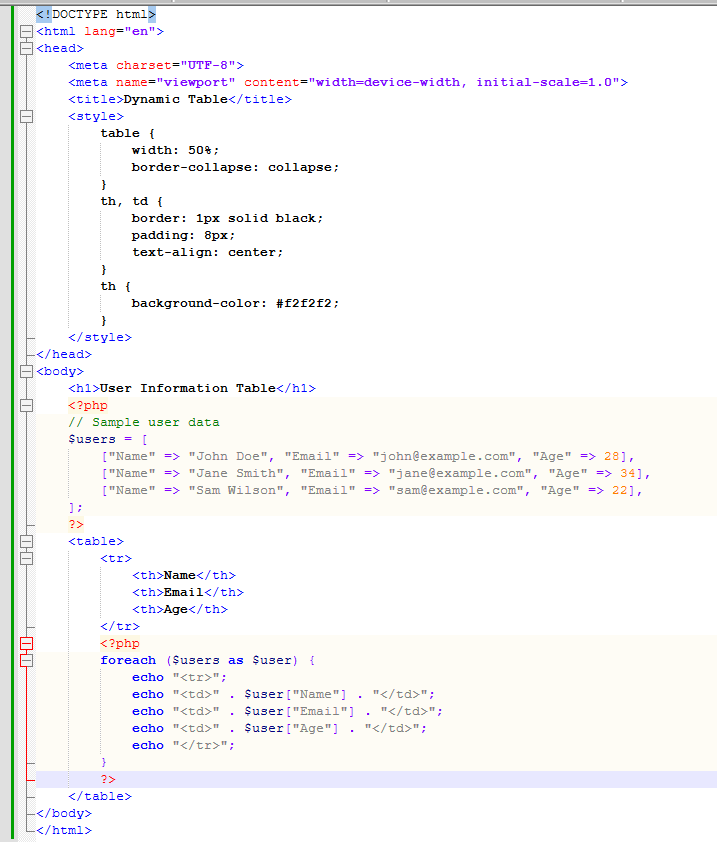
* 1. **PHP Syntax Extra**
  2. **LAB EXERCISES:**

• **PHP Comments:**

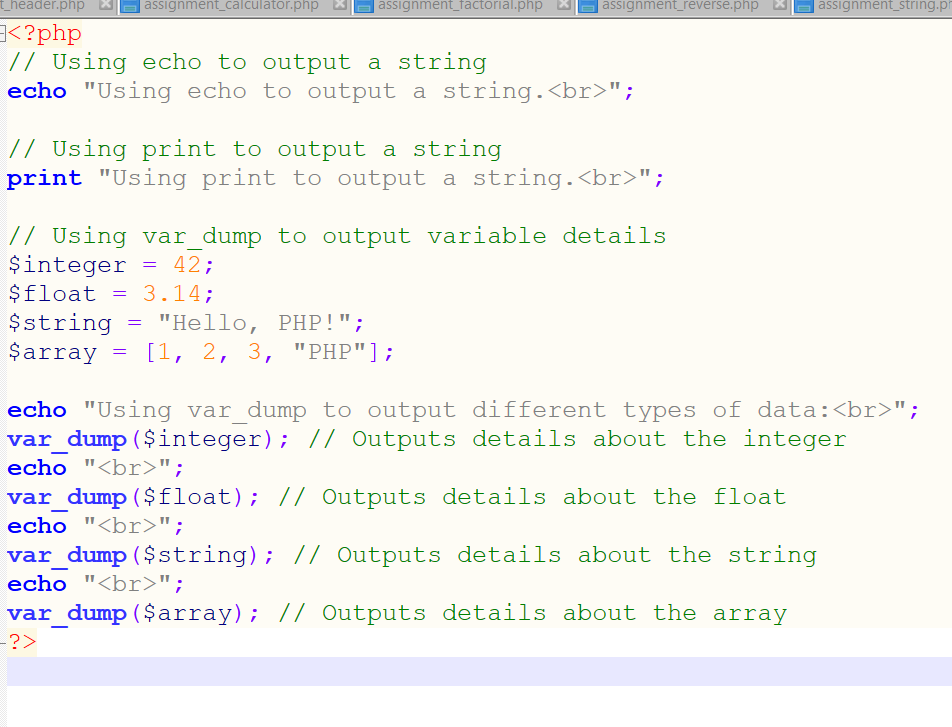
**Write a PHP script that demonstrates the use of single-line (//), multi-line (/\* \*/), and inline (#) comments.**

****

**.Embedding HTML and PHP: Create a web page that uses PHP to dynamically generate HTML content (e.g., a table with user information using PHP).**

****

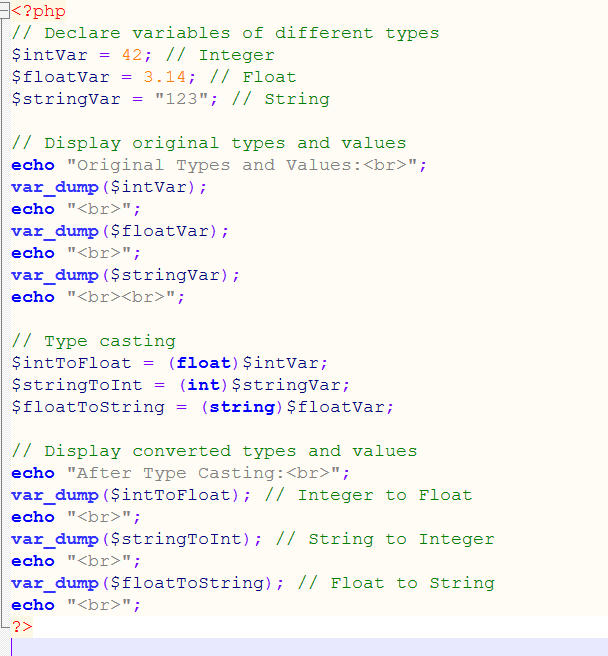
• **Output Statements: Experiment with echo, print, and var\_dump.**

****

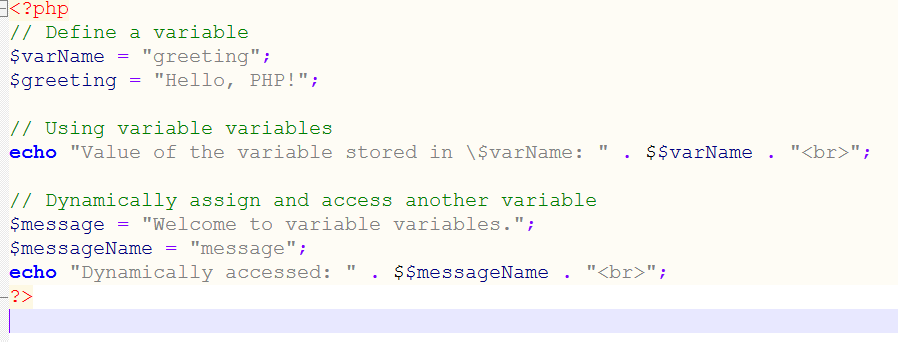
* 1. **PHP Variables**

**Extra LAB EXERCISES:**

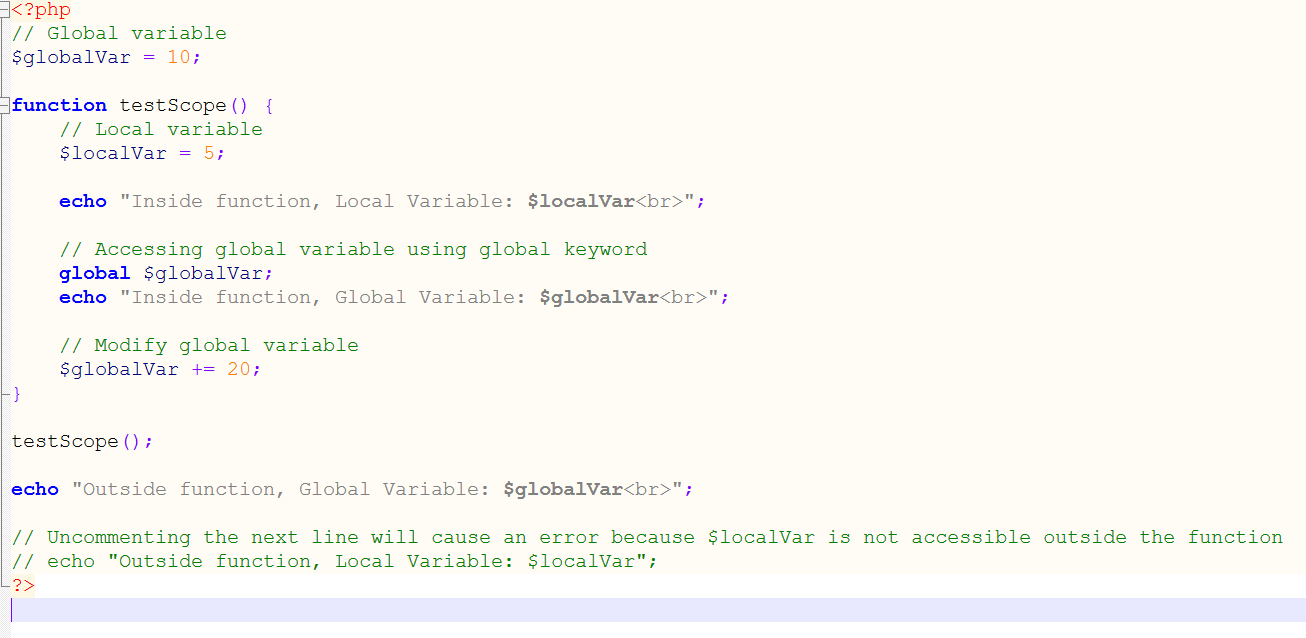
• **Type Casting: Write a script that declares variables of different types and converts them into other types (e.g., integer to float, string to integer). Display the type and value before and after the conversion.**

****

• **Variable Variables: Demonstrate the use of variable variables in PHP. Write a script where a variable name is stored in another variable, and then use it to print the value.**

****

• **Global and Local Scope: Write a script that shows how global and local variables work. Use the global keyword inside a function to access a global variable.**

****

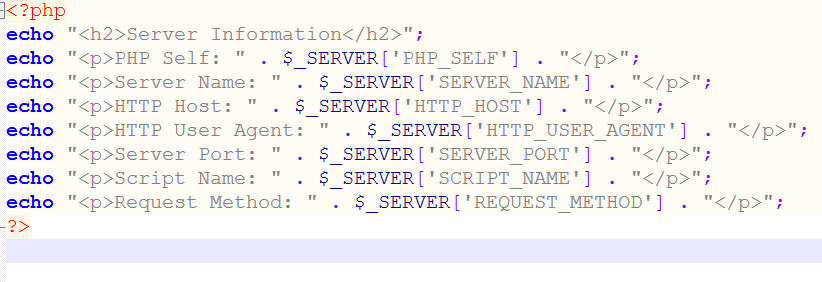
1. **Super Global Variables**

**Extra LAB EXERCISES:**

• **$\_GET and $\_POST: Create two separate forms: one that uses the $\_GET method and one that uses $\_POST. Display the difference in the URL and how data is passed.**

****

• **$\_SERVER: Write a script to display various details of the server environment using $\_SERVER (like PHP\_SELF, SERVER\_NAME, HTTP\_USER\_AGENT, etc.).**

****

• **$\_FILES: Create a form that allows users to upload a file. Handle the uploaded file using the $\_FILES super global and display information about the file.**

****

1. **Practical Example:**

**Multiple Tables and SQL Queries**

**Extra LAB EXERCISES:**

• **Complex Joins: Create a PHP script that connects two or more tables using INNER JOIN, LEFT JOIN, and RIGHT JOIN. Display data from these tables based on specific conditions.**

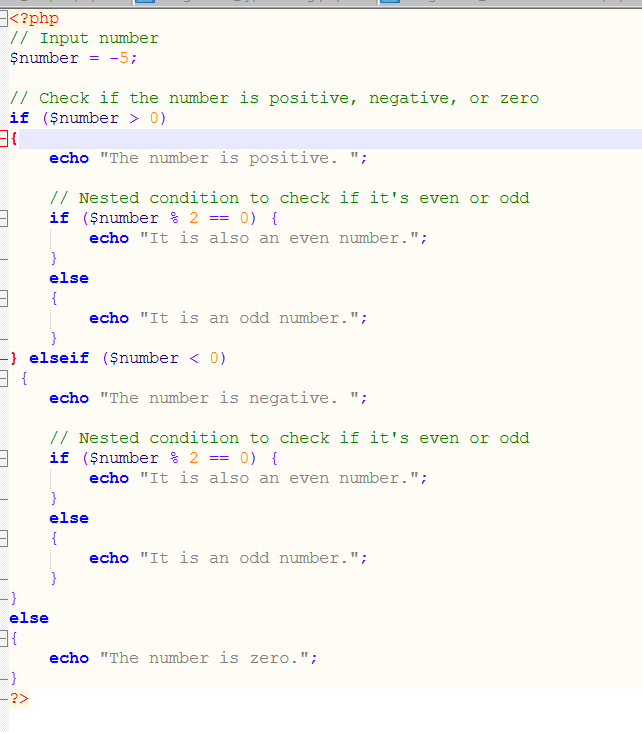
• **Prepared Statements: Implement SQL queries using prepared statements with placeholders to prevent SQL injection in SELECT, INSERT, UPDATE, and DELETE.**

• **Transaction Management: Write a PHP script that uses SQL transactions to insert data into multiple tables, ensuring data integrity in case of an error.**

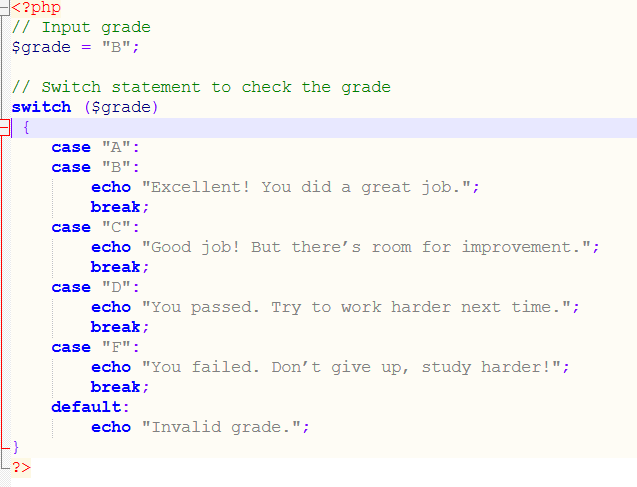
1. **Conditions, Events, and Flows Extra**

**LAB EXERCISES:**

• **Nested Conditions: Write a script that uses nested if-else conditions to categorize a number as positive, negative, or zero, and also check if it's an even or odd number.**

****

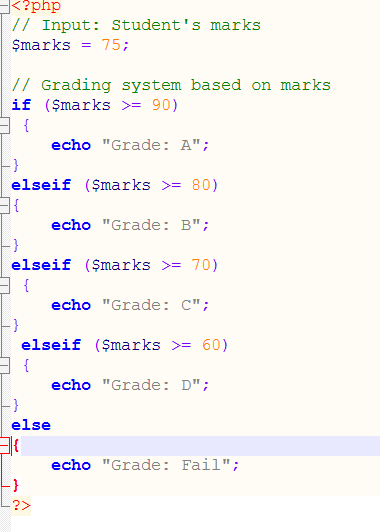
• **Switch Case with Multiple Cases: Write a script that accepts a grade (A, B, C, D, F) and displays a message using a switch statement. Handle multiple cases that fall under the same logic (e.g., A and B show "Excellent").**

****

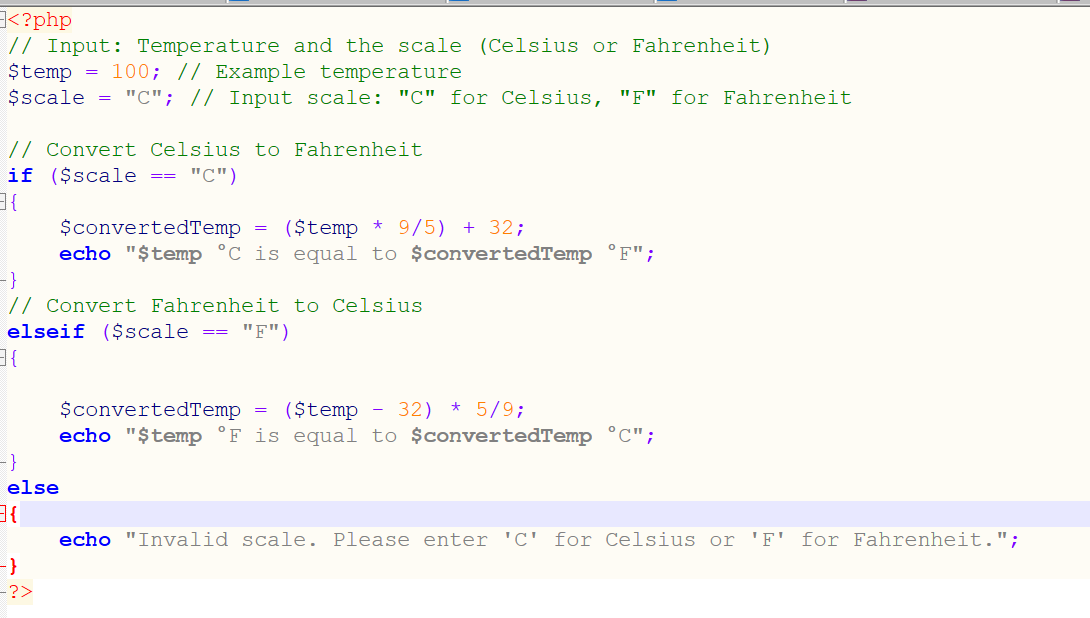
1. **If Condition and If-Else If Extra**

**LAB EXERCISES:**

• **Grading System: Write a PHP program that accepts a student’s marks and outputs their grade using if-else conditions (A, B, C, D, Fail based on score).**

****

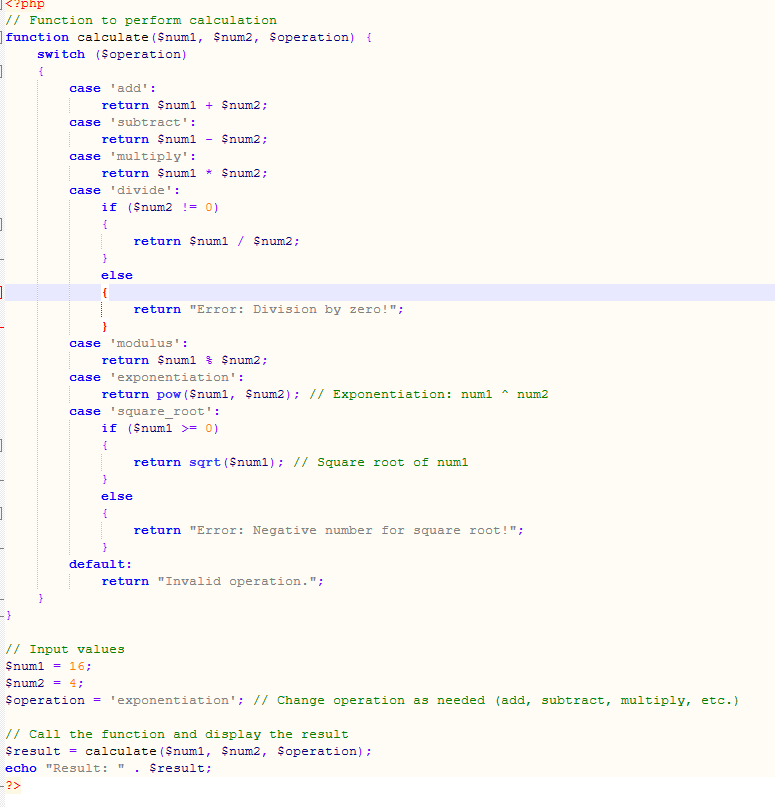
• **Temperature Converter: Write a script that takes temperature in Celsius or Fahrenheit as input and converts it to the other format using if conditions.**

****

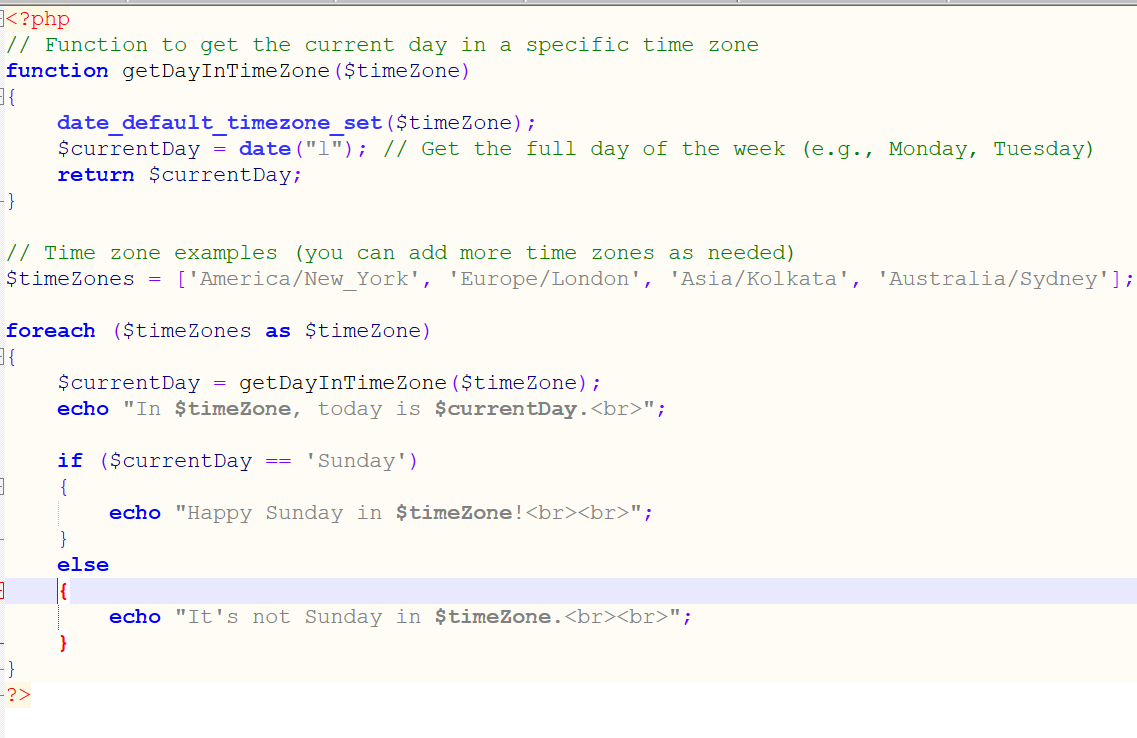
**8.Practical Example: Calculator and Day Finder**

**Extra LAB EXERCISES:**

• **Enhanced Calculator: Modify the calculator to handle more complex operations such as exponentiation (^), modulus (%), and square root (√).**

****

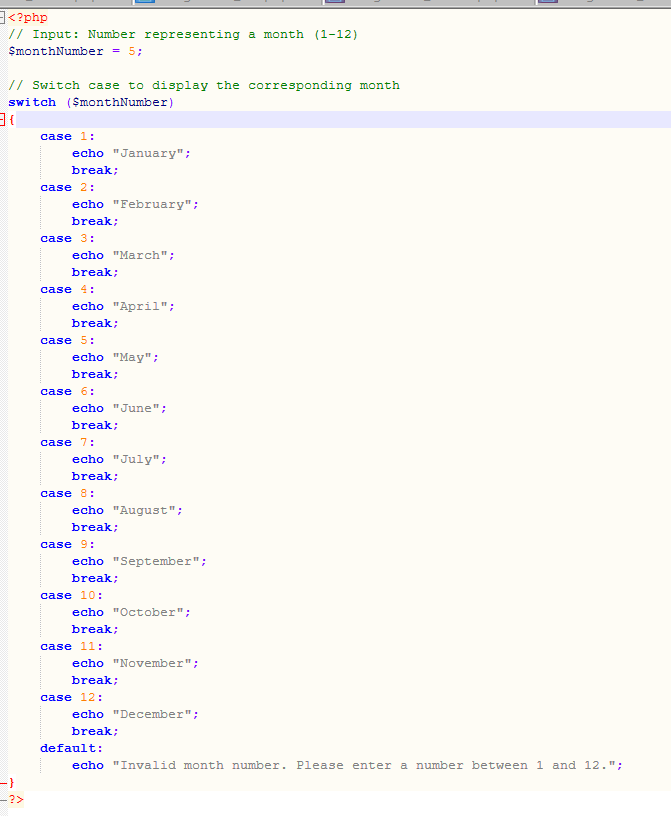
• **Date Finder with Time Zone: Write a script that finds the current day and prints "Happy Sunday" if it's Sunday, but also adjusts for different time zones.**

****

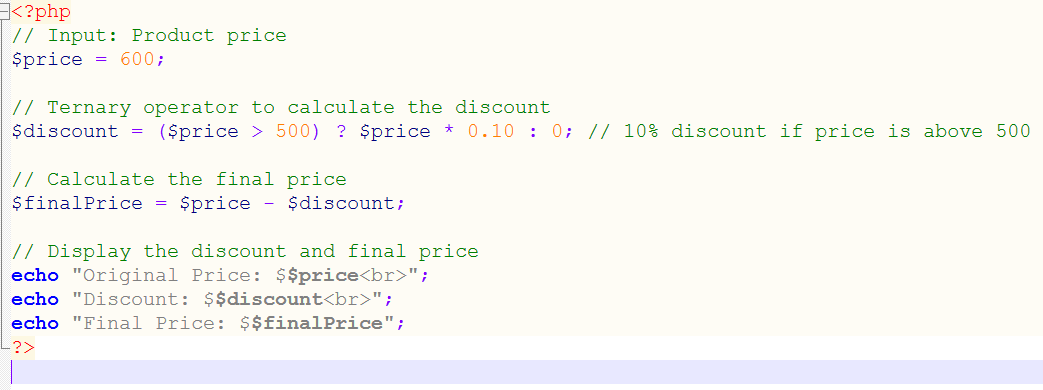
1. **Switch Case and Ternary Operator**

**Extra LAB EXERCISES:**

• **Month Display: Create a program using switch case that takes a number (1-12) and displays the corresponding month.**

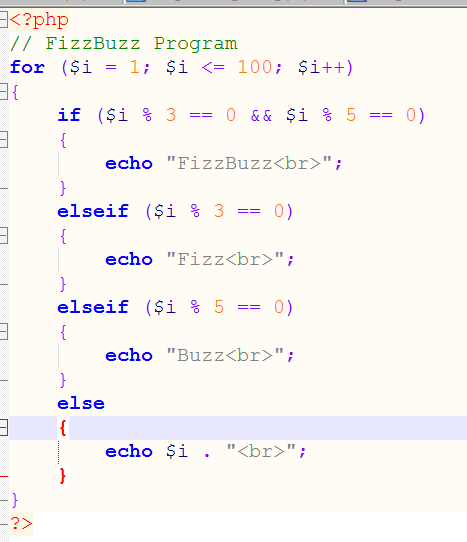
****

• **Discount Calculation (Ternary Operator): Write a script that calculates and displays the discount on a product based on a user-defined price. If the price is above 500, give a 10% discount; otherwise, no discount (use the ternary operator).**

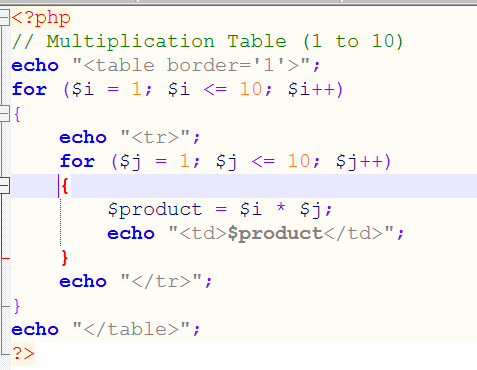
**9. Loops: Do-While, For Each, For Loop**

**Extra LAB EXERCISES:**

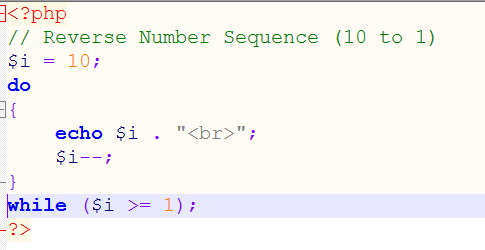
**• FizzBuzz Program: Write a program using a for loop that prints numbers from 1 to 100. But for multiples of 3, print "Fizz" instead of the number, for multiples of 5 print "Buzz", and for multiples of both 3 and 5 print "FizzBuzz".**

****

**• Multiplication Table: Write a PHP script using a nested for loop to generate a multiplication table from 1 to 10.**

****

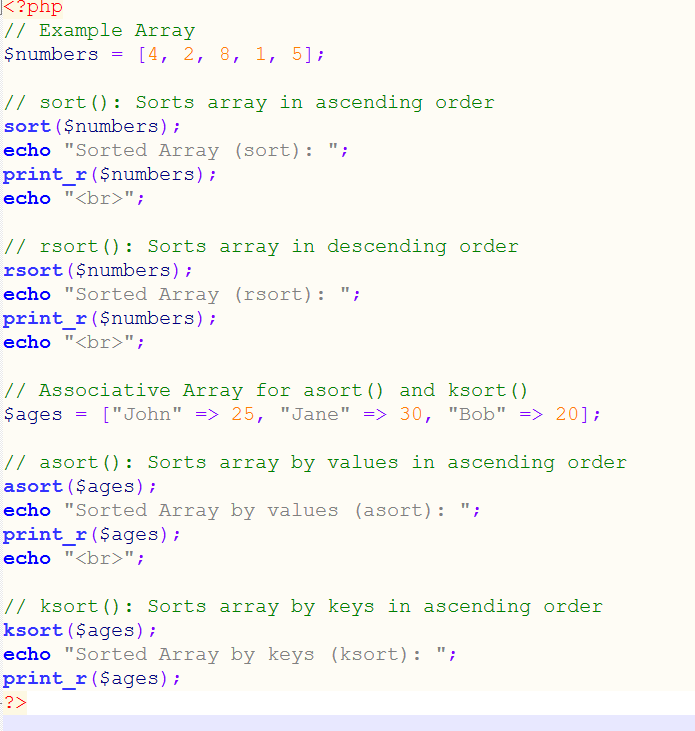
**• Reverse Number Sequence: Write a script using a do-while loop that displays numbers from 10 to 1.**

****

1. **PHP Array and Array Functions Extra**

**LAB EXERCISES:**

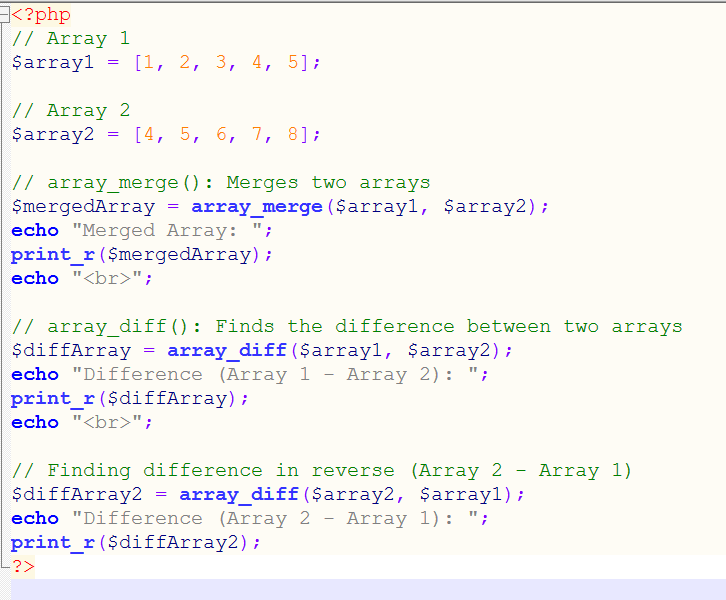
• **Sorting Arrays: Write a script that demonstrates the use of sort(), rsort(), asort(), and ksort() functions to sort arrays.**

****

• **Multi-dimensional Array: Create a multi-dimensional array to store information about products (name, price, and stock). Write a script to display the information in a tabular format.**



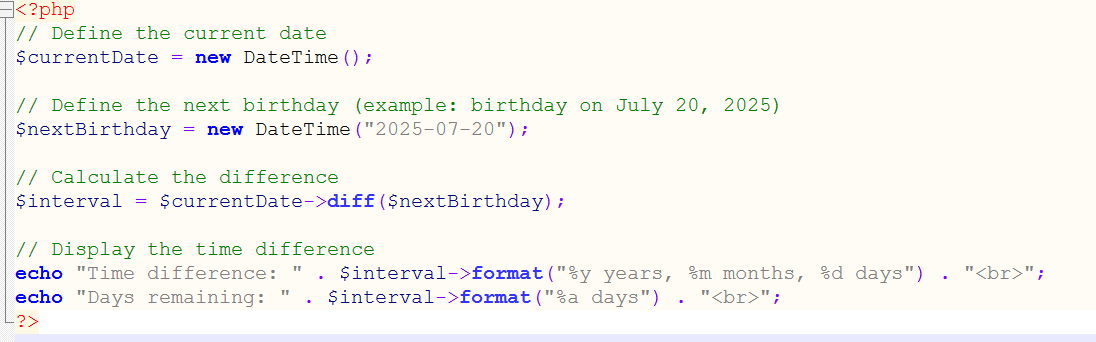
• **Array Merge and Diff: Write a PHP script that merges two arrays and finds the difference between them using array\_merge() and array\_diff().**

****

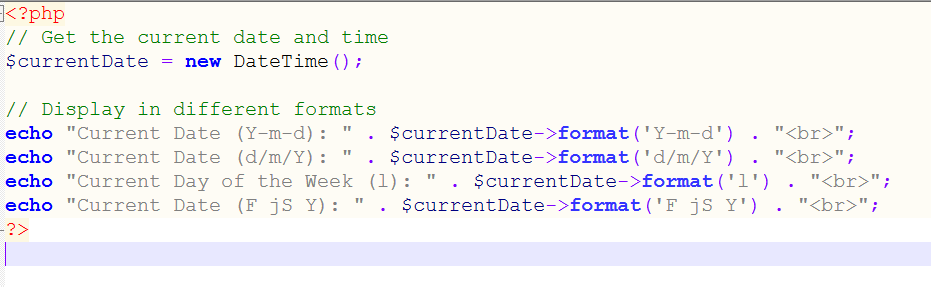
1. **PHP Date-Time Function**

**Extra LAB EXERCISES:**

• **Time Difference: Write a script that calculates the time difference between two dates (e.g., "today" and "next birthday").**

****

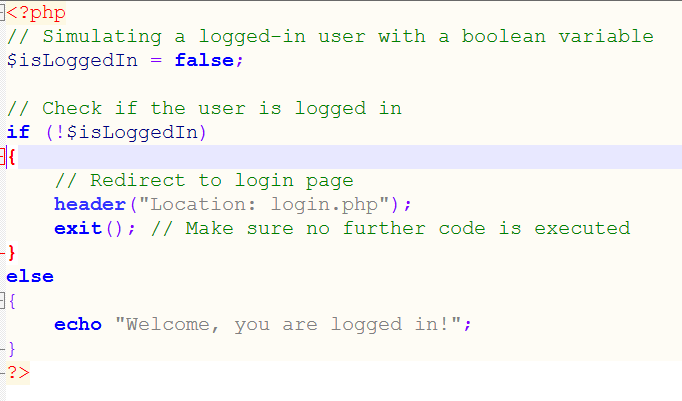
• **Custom Date Formats: Create a script that displays the current date in different formats (e.g., Y-m-d, d/m/Y, l, F jS Y).**

****

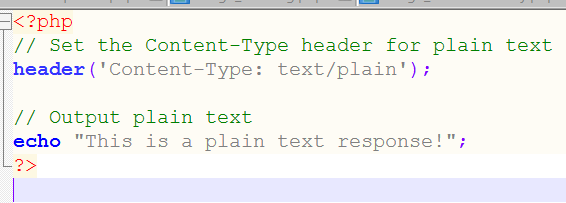
1. **Header Function**

**Extra LAB EXERCISES:**

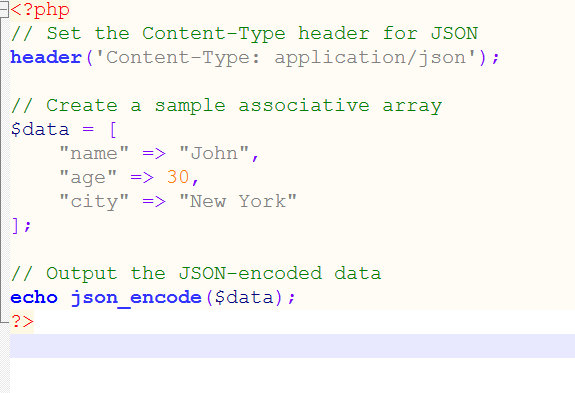
• **Page Redirect Based on Condition: Write a script that checks if a user is logged in (use a boolean variable). If not, use the header() function to redirect them to a login page.**

****

• **Content-Type Header: Write a script that sets the Content-Type header to return a plain text file or a JSON response**

**Code Example 1: Content-Type for Plain Text**

**Code Example 2: Content-Type for JSON Response**

****

1. **Include and Require**

**Extra LAB EXERCISES:**

• **Template System: Write a PHP script that includes header, navigation, and footer files in multiple web pages to create a basic template system.**

**1. Template System**

**Create a simple template system using include. This system will have a header, navigation, and footer that can be reused across multiple pages.**

**Steps:**

1. **Create separate PHP files for header.php, navigation.php, and footer.php.**
2. **Use the include function to include these files in the main pages.**

**Code Example:**

* **header.php**

**php**

**Copy code**

**<!DOCTYPE html>**

**<html>**

**<head>**

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

**</head>**

**<body>**

**<header>**

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

**</header>**

* **navigation.php**

**php**

**Copy code**

**<nav>**

**<ul>**

**<li><a href="index.php">Home</a></li>**

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

**<li><a href="contact.php">Contact</a></li>**

**</ul>**

**</nav>**

* **footer.php**

**php**

**Copy code**

**<footer>**

**<p>&copy; 2025 My Website. All Rights Reserved.</p>**

**</footer>**

**</body>**

**</html>**

* **index.php**

**php**

**Copy code**

**<?php**

**include 'header.php';**

**include 'navigation.php';**

**?>**

**<main>**

**<h2>Home Page</h2>**

**<p>Welcome to the homepage of our website.</p>**

**</main>**

**<?php include 'footer.php'; ?>**

* **about.php**

**php**

**Copy code**

**<?php**

**include 'header.php';**

**include 'navigation.php';**

**?>**

**<main>**

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

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

**</main>**

**<?php include 'footer.php'; ?>**

• **File Not Found Handling: Use require to include a critical file. If the file doesn't exist, display a custom error message instead of the default PHP error.**

**2. File Not Found Handling**

**Use the require function to include a critical file. If the file doesn't exist, display a custom error message.**

**Steps:**

1. **Use require to include the file.**
2. **Use try-catch or a custom logic to handle errors.**

**Code Example:**

* **critical\_file.php**

**php**

**Copy code**

**<?php**

**echo "This is a critical file.";**

**?>**

* **main.php**

**php**

**Copy code**

**<?php**

**$file = 'critical\_file.php';**

**if (file\_exists($file)) {**

**require $file;**

**} else {**

**echo "<p style='color: red;'>Error: The required file <b>$file</b> is missing. Please contact the administrator.</p>";**

**}**

**// The rest of the code**

**echo "<p>Rest of the application works as expected.</p>";**

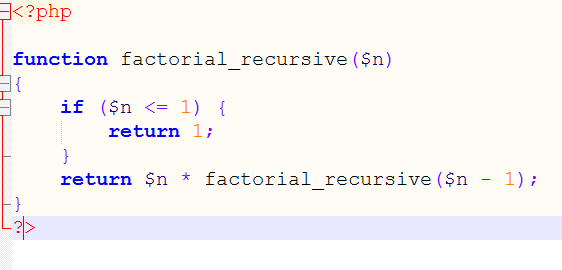
**?>**

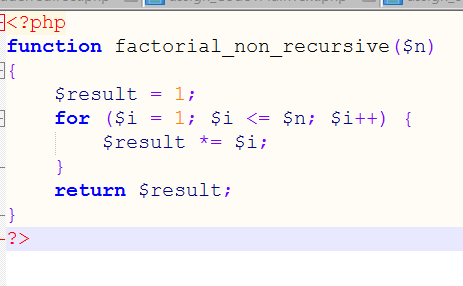
1. **Practical Example: Calculator, Factorial, String Reverse**

**Extra LAB EXERCISES:**

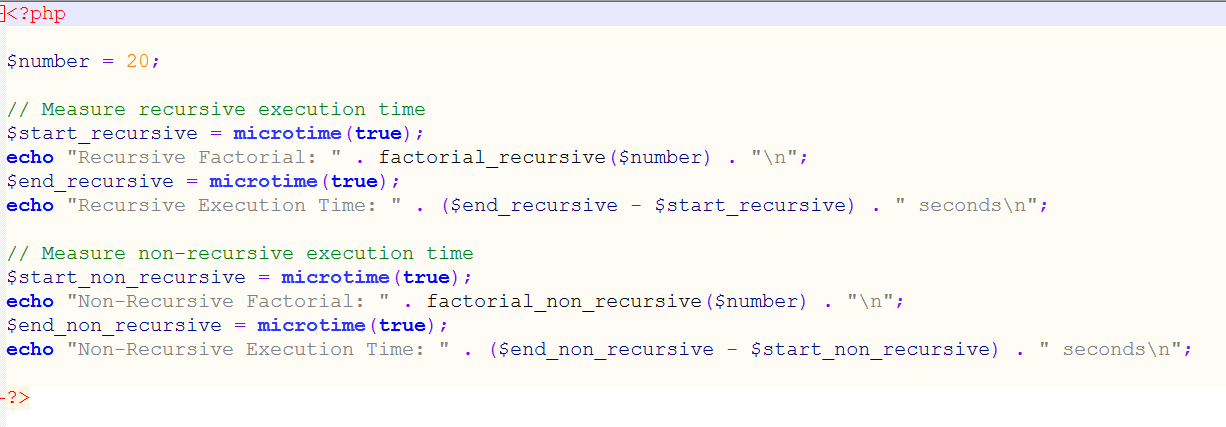
• **Enhanced Factorial: Write a recursive and non-recursive function to calculate the factorial of a number. Compare their performance for large numbers.**

**Recursive Factorial Function**

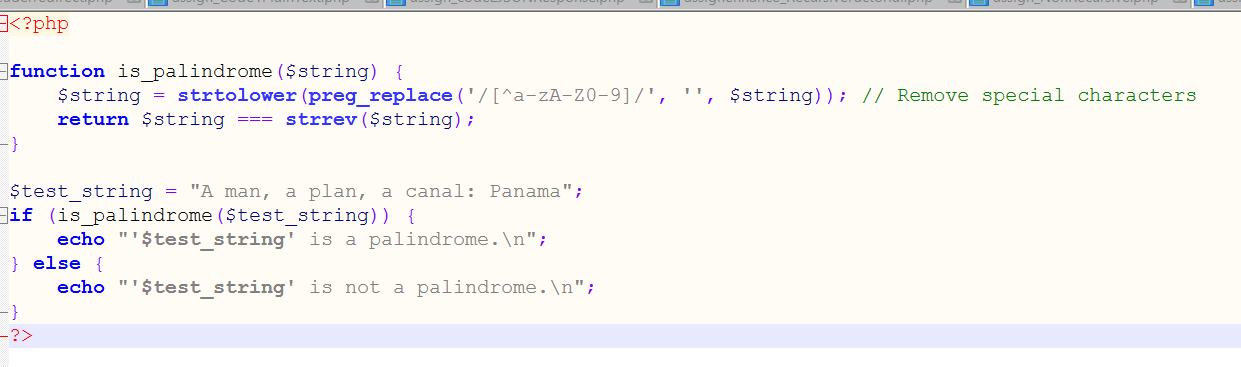
****

****

**Performance Comparison**

****

• **Palindrome Checker: Create a function that checks if a given string is a palindrome.**

****

• **File Upload: Create a form that allows users to upload a file. Upon submission, download the file using a button click and display the file’s details (name, type, size).**

****