PHP Mini Course: Intro, Syntax, Variables, Conditions, Loops

0. Setup & How to Run PHP

Before coding, set up a PHP environment. You can use XAMPP, WAMP, Laragon, MAMP, or PHP's built-in server with the command: php -S localhost:8000. Save your files with the .php extension and place them in the server directory (such as htdocs or your project folder).

1. First PHP File – Intro & Basic Syntax

1.1 Minimal PHP Script (hello.php)

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

This script prints 'Hello, World!' to the browser. The <?php tag starts PHP code and ?> ends it. The echo statement outputs text, and each statement ends with a semicolon (;).

1.2 PHP Inside HTML (intro.php)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>PHP Intro Page</title>
</head>
<body>
    <h1>Welcome to PHP</h1>

    <p>
        <?php
        echo "This text is generated by PHP.";
        ?>
    </p>

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

Here, PHP is embedded inside HTML. The server runs the PHP code and sends only the generated HTML to the browser.

2. PHP Syntax in Detail

2.1 Statements & Semicolons (syntax.php)

```
<?php
echo "First line.<br>";
echo "Second line.<br>";
```

```
echo "Third line.<br>";
?>
```

Each echo statement ends with a semicolon. The
 tag is an HTML line break.

2.2 Whitespace & Newlines

```
<?php
echo "This";
echo " is";
echo " on";
echo " one";
echo " line.";
?>

<?php
echo "This<br>";
echo "is<br>";
echo "on<br>";
echo "multiple<br>";
echo "lines.<br>";
?>
```

Whitespace and newlines in PHP code don't usually affect the output formatting; HTML tags control how content appears in the browser.

2.3 Case Sensitivity

```
<?php
$number = 10;

// This works
echo $number . "<br>";

// This will cause a notice/error (undefined variable)
echo $Number . "<br>";
?>
```

Variable names are case-sensitive. $number and $Number are treated as two different variables.

3. Comments & Output Functions

3.1 Comments (comments.php)

```
<?php
// This is a single-line comment
echo "Hello from PHP!<br>"; // Comment after code

# This is another single-line comment style
echo "This line is also executed.<br>";

/*
 This is a multi-line comment.
 You can write longer explanations here.
*/
echo "Multi-line comments are useful for documentation.<br>";
?>
```

Comments are ignored by the PHP engine and are used to explain code or temporarily disable lines.

3.2 echo vs print

```
<?php
echo "Using echo function.<br>";
print "Using print function.<br>";

$result = print "print returns 1, which you can store.<br>";
echo "The value returned by print is: " . $result . "<br>";
?>
```

Both echo and print output text. echo is slightly faster and does not return a value, while print always returns 1.

3.3 HTML in echo

```
<?php
echo "<h2>This is a heading printed by PHP</h2>";
echo "<p>This is a paragraph printed by PHP.</p>";
?>
```

4. Variables & Data Types

4.1 Declaring & Using Variables (variables_basic.php)

```
<?php
$name = "Emon";
$age = 22;
$height = 5.9;
$is_student = true;

echo "Name: " . $name . "<br>";
echo "Age: " . $age . "<br>";
echo "Height: " . $height . "<br>";
echo "Is student: " . $is_student . "<br>";
?>
```

Variables in PHP start with $, and PHP is loosely typed, so you don’t need to declare data types explicitly.

4.2 Variable Naming Rules

```
<?php
$firstName = "EliteDev";
$_count = 10;
$age2 = 25;
?>
```

These are valid variable names. Invalid examples include: $2age or $first-name.

4.3 Basic Data Types (data_types.php)

```
<?php
$fullName = "EliteDev Emon";   // string
$age = 25;                     // integer
$price = 19.99;                // float
$isAdmin = false;              // boolean

echo "Full Name: " . $fullName . "<br>";
echo "Age: " . $age . "<br>";
```

```
echo "Price: " . $price . "<br>";
echo "Is Admin: " . $isAdmin . "<br>";
?>
```

4.4 Inspecting Types with var_dump

```
<?php
$fullName = "EliteDev Emon";
$age = 25;
$price = 19.99;
$isAdmin = false;

var_dump($fullName);
echo "<br>";
var_dump($age);
echo "<br>";
var_dump($price);
echo "<br>";
var_dump($isAdmin);
?>
```

var_dump() is very helpful for debugging because it shows both the value and the data type of a variable.

4.5 String Concatenation & Interpolation (strings.php)

```
<?php
$name = "Emon";
$language = "PHP";

// Concatenation using .
echo "Hello, " . $name . "! Welcome to " . $language . ".<br>";

// Interpolation using double quotes
echo "Hello, $name! Welcome to $language.<br>";

// Single quotes (no interpolation)
echo 'Hello, $name! Welcome to $language.<br>';
?>
```

Use . to concatenate strings. Double-quoted strings support variable interpolation, while single-quoted strings do not.

5. Conditions (if, else, elseif)

5.1 Basic if (if_basic.php)

```
<?php
$age = 20;

if ($age >= 18) {
    echo "You are an adult.";
}
?>
```

5.2 if ... else (if_else.php)

```
<?php
$age = 16;

if ($age >= 18) {
    echo "You are allowed to vote.";
} else {
    echo "You are not allowed to vote.";
}
?>
```

5.3 if ... elseif ... else (if_elseif.php)

```
<?php
$score = 75;

if ($score >= 80) {
    echo "Grade: A";
} elseif ($score >= 60) {
    echo "Grade: B";
} elseif ($score >= 40) {
    echo "Grade: C";
} else {
    echo "Grade: F";
}
?>
```

5.4 Comparison & Logical Operators

```
<?php
$a = 10;
$b = 20;

if ($a == 10) {
    echo "$a is equal to 10<br>";
}

if ($b != 10) {
    echo "$b is not equal to 10<br>";
}

if ($b > $a) {
    echo "$b is greater than $a<br>";
}

$age = 20;
$hasID = true;

if ($age >= 18 && $hasID) {
    echo "You are allowed to enter.<br>";
}

$member = false;
```

```php
if ($age >= 18 || $member) {
    echo "You can join the event.<br>";
}

$isBlocked = false;

if (!$isBlocked) {
    echo "You are not blocked.<br>";
}
?>
```

5.5 Nested Conditions (nested_if.php)

```php
<?php
$age = 22;
$country = "Bangladesh";

if ($country == "Bangladesh") {
    if ($age >= 18) {
        echo "You are an adult in Bangladesh.";
    } else {
        echo "You are a minor in Bangladesh.";
    }
} else {
    echo "You are from another country.";
}
?>
```

6. Loops (for, while, do-while, foreach)

6.1 for Loop (for_loop.php)

```php
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "Number: $i <br>";
}
?>
```

6.2 while Loop (while_loop.php)

```php
<?php
$x = 1;

while ($x <= 5) {
    echo "Value: $x <br>";
    $x++;
}
?>
```

6.3 do ... while Loop (do_while.php)

```php
<?php
$y = 6;

do {
    echo "This runs at least once. Value: $y <br>";
    $y++;
```

```
} while ($y <= 5);
?>
```

6.4 foreach Loop with Indexed Array (foreach_basic.php)

```
<?php
$fruits = ["Apple", "Mango", "Banana", "Orange"];

foreach ($fruits as $fruit) {
    echo $fruit . "<br>";
}
?>
```

6.5 foreach with Associative Array (foreach_assoc.php)

```
<?php
$person = [
    "name" => "Emon",
    "age" => 22,
    "country" => "Bangladesh"
];

foreach ($person as $key => $value) {
    echo $key . " : " . $value . "<br>";
}
?>
```

6.6 break and continue (break_continue.php)

```
<?php
echo "Using break:<br>";

for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        break;
    }
    echo $i . "<br>";
}

echo "<br>Using continue:<br>";

for ($j = 1; $j <= 10; $j++) {
    if ($j == 5) {
        continue;
    }
    echo $j . "<br>";
}
?>
```

7. Full Example – Student Grade Report (student_grades.php)

```
<?php
$students = [
    ["name" => "Rahim", "marks" => 85],
    ["name" => "Karim", "marks" => 72],
    ["name" => "Hasan", "marks" => 65],
    ["name" => "Imran", "marks" => 40],
```

```
    ["name" => "Sakib", "marks" => 30]
];

function getGrade($marks)
{
    if ($marks >= 80) {
        return "A+";
    } elseif ($marks >= 70) {
        return "A";
    } elseif ($marks >= 60) {
        return "A-";
    } elseif ($marks >= 50) {
        return "B";
    } elseif ($marks >= 40) {
        return "C";
    } else {
        return "F";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Student Grades</title>
</head>
<body>
    <h1>Student Grade Report</h1>

    <table border="1" cellpadding="8" cellspacing="0">
        <tr>
            <th>Name</th>
            <th>Marks</th>
            <th>Grade</th>
            <th>Status</th>
        </tr>
        <?php foreach ($students as $student): ?>
            <?php
            $grade = getGrade($student["marks"]);

            if ($student["marks"] >= 40) {
                $status = "Passed";
            } else {
                $status = "Failed";
            }
            ?>
            <tr>
                <td><?php echo $student["name"]; ?></td>
                <td><?php echo $student["marks"]; ?></td>
                <td><?php echo $grade; ?></td>
                <td><?php echo $status; ?></td>
            </tr>
```

```
        <?php endforeach; ?>
    </table>
</body>
</html>
```

8. Exercises

Exercise 1 – Personal Info Page (info.php)

```
<?php
$name = "Emon";
$country = "Bangladesh";
$age = 22;
$gpa = 3.75;

echo "My name is $name, I am $age years old, I'm from $country and my GPA is $gpa.";
?>
```

Exercise 2 – Temperature Checker (temperature.php)

```
<?php
$temperature = 28;

if ($temperature > 30) {
    echo "It's Hot";
} elseif ($temperature > 20) {
    echo "Normal Weather";
} else {
    echo "Cold";
}
?>
```

Exercise 3 – For Loop 1–10 (numbers.php)

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo $i . "<br>";
}
?>
```

Exercise 4 – Countdown with While (countdown.php)

```
<?php
$number = 10;

while ($number >= 1) {
    echo $number . "<br>";
    $number--;
}
?>
```

Exercise 5 – Colors with foreach (colors.php)

```
<?php
$colors = ["Red", "Green", "Blue", "Yellow"];

foreach ($colors as $color) {
    echo $color . "<br>";
```

```
}
?>
```

Exercise 6 – Pass/Fail for Multiple Students (pass_fail.php)

```
<?php
$students = [
    ["name" => "A", "marks" => 75],
    ["name" => "B", "marks" => 35],
    ["name" => "C", "marks" => 50],
    ["name" => "D", "marks" => 20]
];

foreach ($students as $student) {
    $name = $student["name"];
    $marks = $student["marks"];

    if ($marks >= 40) {
        echo "$name passed with $marks marks.<br>";
    } else {
        echo "$name failed with $marks marks.<br>";
    }
}
?>
```

9. Recap

In this mini-course, you learned how to set up PHP, write basic scripts, embed PHP in HTML, use variables and data types, control program flow with conditions, and repeat tasks using loops. With these fundamentals, you can start building dynamic web pages and small PHP applications.