

Unit 10 –Website Design & Development

Lesson [11] – PHP Part I

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 browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do

- 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

Run a PHP File in XAMPP

- Install xampp server on your computer. After done installing xampp server start it by enabling Apache & MySQL server.
- Open your browser and type `http://localhost/`. you will got the welcome page.
- Create PHP script using any editor
- Save file as following... Create a directory inside `xampp/htdocs/myproject/`
- Open Your browser and write in url : `localhost/myproject/firstProg.php`

Basic PHP Syntax

- A PHP script is executed on the server, and the plain HTML result is sent back to the browser.
- A PHP script can be placed anywhere in the document.
- A PHP script starts with `<?php` and ends with `?>`:

```
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello ESOFTE";
?>
</body>
```



My first PHP page

Hello ESOFTE

PHP Variables

- 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)

PHP Variables

```
<body>

<h1><PHP Variables</h1>|
<?php
$x=789;
$y=456.123;
$name="Esoft";
$_test_="Welcome";
echo ($x) ;
echo ("<br>");
echo ($_test_." to ".$name." METRO CAMPUS");
?>
</body>
```

789

Welcome to Esoft METRO CAMPUS

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
- Conditional assignment operators

PHP Conditional Statements

- 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

PHP Conditional Statements

```
<body>
<?php
$x=96;
$y =78;
$z= 89;
if ($x>$y) {
    if ($x>$z){
        echo "Max Number is: ", $x;
    }
    else{
        echo "Max Number is: ", $z;
    }
} else {
    if ($y>$z){
        echo "Max Number is: ", $y;
    }
    else{
        echo "Max Number is: ", $z;
    }
}
?>
</body>
```

Max Number is: 96

PHP Loops

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

- 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

PHP Loops

```
<body>
<?php
$x = 1;
$sum = 0;
while($x <= 10) {
    $sum = $sum+$x;
    $x++;
}
echo "The sum is: $sum";
?>
</body>
```

```
<body>
<h1>The PHP for Loop</h1>
<?php
$sum=0;
for ($x = 0; $x <= 100; $x++) {
    echo "The number is: $x <br>";
    $sum = $sum+$x;
}
echo "The sum is: $sum";
?>
</body>
```

```
<body>
<h1>The PHP do...while Loop</h1>
<?php
$x = 100;
do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
</body>
```

PHP Functions

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.

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

Hello world!
Hello world!
Hello world!

PHP Functions

- Information can be passed to functions through arguments. An argument is just like a variable.

```
<body>
<?php
function familyName ($fname) {
    echo "$fname Gunawardhana.<br>";
}

familyName ("kasun");
familyName ("Nadith");
familyName ("Harsha");
familyName ("Gayan");
familyName ("Sadun");
?>
```

```
kasun Gunawardhana.
Nadith Gunawardhana.
Harsha Gunawardhana.
Gayan Gunawardhana.
Sadun Gunawardhana.
```

PHP Functions

```
<body>
<?php
function sum($x, $y) {
    $z = $x + $y;
    echo "sum is:". $z. "<br>";
}
sum(15, 45);
sum(15, 555);
?>

</body>
```

```
sum is:60
sum is:570
```

PHP Arrays

- An array is a special variable, which can hold more than one value at a time.

```
<body>
<h1>PHP Arrays</h1>
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
</body>
```

PHP Arrays

I like Volvo, BMW and Toyota.

The End...!!