

# Introduction to PHP

---

CO226 – Lab 03

Kalani Udayanthi  
kalaniu@eng.pdn.ac.lk

# Outline

- PHP
- Basic PHP Syntax
- Comments in PHP
- Variables in PHP
- Concatenation Operation
- Conditional Statements in PHP
- Loops in PHP
- Arrays in PHP
- Functions in PHP

# PHP

- **PHP : Hypertext Preprocessor (Personal Home Page)**
- Server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- Used for form handling, file processing, and database access

- PHP scripts are executed on the server
- PHP supports many databases
  - MySQL
  - Informix
  - Oracle
  - Sybase
  - Solid
  - PostgreSQL
- PHP is an open source software, free to download and use

# What is a PHP File?

- PHP files may contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- File extensions
  - ".php", ".php3", or ".phtml"
- PHP runs on different platforms (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource:

# Basic PHP Syntax

```
<?php  
    //Your code goes here  
?>
```

```
<html>  
<body>  
  
<h1>My first PHP page</h1>  
  
<?php  
echo "Hello World!";  
?>  
  
</body>  
</html>
```

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

Hello World!

# Comments in PHP

```
<html>  
<body>
```

```
<?php
```

```
// This is a single-line comment
```

```
# This is also a single-line comment
```

```
/*  
This is a multiple-lines comment  
block that spans over multiple  
lines  
*/
```

```
?>
```

```
</body>  
</html>
```



```
<!DOCTYPE html>
<html>
<body>

<?php

echo "Comments are as follows"

// This is a single-line comment

# This is also a single-line comment

/*This is a multiple-lines comment block
that spans over multiple
lines*/
?>

</body>
</html>
```

Comments are as follows

# Variables in PHP

- 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**
  - \$name ≠ \$NAME

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

# Output Variables

```
<?php
$name= "Kumara";
echo "My name is $name";
?>
```

```
<?php
$name = 'Kumara';    // Assign the value Kumara' to $name
$temp = &$name;      // Reference $name via $temp.

$temp = "My name is $temp";    // Alter $temp...
echo "$temp <br>";
echo $name;    // $name is altered too.
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$number = 9;

$newNumber = &$number;

echo "Number : $number<br>";
echo "New Number : $newNumber<br>";

echo "<br>After changing the newNumber variable <br>";

$newNumber = 11;
echo "Number : $number<br>";
echo "New Number : $newNumber<br>";

?>

</body>
</html>
```

Number : 9  
New Number : 9

After changing the newNumber variable  
Number : 11  
New Number : 11

# Concatenation Operation

```
<?php  
$txt1="Hello ,";  
$txt2="Have a nice day!";  
echo $txt1 . " " . $txt2  
//outputs "Hello , Have a nice day!"  
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php

$txt1="Hello ,";
$txt2="Have a nice day!";
echo $txt1 . " " . $txt2

?>

</body>
</html>
```

Hello , Have a nice day!

# Conditional Statements in PHP

- **if statement** - use this statement to execute some code only if a specified condition is true
- **if...else statement** - use this statement to execute some code if a condition is true and another code if the condition is false
- **if...elseif....else statement** - use this statement to select one of several blocks of code to be executed
- **switch statement** - use this statement to select one of many blocks of code to be executed

# if

```
if (condition) {  
    code to be executed if condition is true;  
}
```

```
<?php  
$t = date("H");  
  
if ($t < "20") {  
    echo "Hi";  
}  
?>
```



```
<!DOCTYPE html>
<html>
<body>

<?php

$a = 18;
$b = 7;

if ($a > $b)
    echo "$a is bigger than $b";

?>

</body>
</html>
```

18 is bigger than 7

# If else

```
if (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```

```
<?php  
$t = date("H");  
  
if ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$a = 13;
```

```
$b = 7;
```

```
if ($a > $b)
```

```
    echo "$a is bigger than $b";
```

```
else
```

```
{echo "$a is smaller than $b";}
```

```
?>
```

```
</body>
```

```
</html>
```

13 is bigger than 7

# If elseif

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if first condition is false and this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

```
<?php  
$t = date("H");  
  
if ($t < "10") {  
    echo "Have a good morning!";  
} elseif ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

```
<body>

<?php

$age = 75;

if ($age <= 19)
    echo "This one is in Teen Age ";

elseif ($age <= 55)
{echo "This one is in Middle Age ";}

else
{
    echo "This one is in Old Age <br> ";
    echo "We need to respect them!";
}

?>

</body>
</html>
```

This one is in Old Age  
We need to respect them!

# Switch

```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all labels;  
}
```

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor
        green!";
}
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$favcolor = "blue";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red,
blue, nor green!";
}
?>

</body>
</html>
```

Your favorite color is blue!



# Loops in PHP

- **while** - loops through a block of code while a specified condition is true
- **do...while** - loops through a block of code once, and then repeats the loop as long as a 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

# While loop

- Executes a block of code as long as the specified condition is true.

```
while (condition is true) {  
    code to be executed;  
}
```

- Print numbers from 1 to 10

```
<?php  
$x = 1;  
  
while($x <= 10) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>

</body>
</html>
```

The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5

# Do while

- always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

```
do {  
    code to be executed;  
} while (condition is true);
```

```
<?php  
$x = 1;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>

</body>
</html>
```

The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5

# For loop

- used when you know in advance how many times the script should run.

```
for (init counter; test counter;  
    increment counter) {  
    code to be executed for each iteration;  
}
```

***init counter*** : Initialize the loop counter value

***test counter*** : Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

***increment counter*** : Increases the loop counter value

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
for ($x = 0; $x <= 100; $x+=10) {
    echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0  
The number is: 10  
The number is: 20  
The number is: 30  
The number is: 40  
The number is: 50  
The number is: 60  
The number is: 70  
The number is: 80  
The number is: 90  
The number is: 100

# Foreach loop

- works only on arrays, and is used to loop through each key/value pair in an array.

```
foreach ($array as $value) {  
    code to be executed;  
}
```

- Print the elements in the array “colors”

```
<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as value){  
    echo "$value <br>";  
}  
?>
```



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$colors = array("red", "green",  
"blue", "yellow");
```

```
foreach ($colors as $value) {  
    echo "$value <br>";  
}
```

```
?>
```

```
</body>
```

```
</html>
```

red

green

blue

yellow

# Arrays in PHP

```
array();
```

- **Numeric / Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays

# Indexed/Numeric Array

- assigned automatically,

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

- assigned manually,

```
$colors[0] = " Red ";  
$colors[1] = " Orange ";  
$ colors[2]= " Blue ";  
$ colors[3]= " Green ";
```

```
<?php  
$colors= array("Red", "Orange", "Blue", "Green");  
echo "I like " . $colors[0] . ", " . $colors[3] . " and " . $ colors[3] . ".";  
?>
```

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

I like Volvo, BMW and Toyota.

# Associative Array

- arrays that use named keys that you assign to them.

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

```
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
echo "Peter is " . $age['Peter'] . " years old."  
?>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$age = array("Peter"=>"35", "Ben"=>"37",
"Joe"=>"43");

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>

</body>
</html>
```

Key=Peter, Value=35  
Key=Ben, Value=37  
Key=Joe, Value=43

# Multidimensional Arrays

- an array containing one or more arrays.

```
$cars = array  
(  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```

```
$families = array (  
    "Griffin"=>array ( "Peter", "Lois", "Megan" ),  
    "Quagmire"=>array( "Glenn", "Aristotle", "Fox"),  
    "Brown"=>array ( "Cleveland", "Loretta", "Junior")  
);
```

```
<?php  
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2]."<br>;  
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2]."<br>;  
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2]."<br>;  
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2]."<br>;  
?>
```

```

<!DOCTYPE html>
<html>
<body>

<?php
$cars = array (
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);

for ($row = 0; $row < 4; $row++) {
    echo "<p><b>Row number $row</b></p>";
    echo "<ul>";
    for ($col = 0; $col < 3; $col++) {
        echo "<li>".$cars[$row][$col]."</li>";
    }
    echo "</ul>";
}
?>

</body>
</html>

```

#### Row number 0

- Volvo
- 22
- 18

#### Row number 1

- BMW
- 15
- 13

#### Row number 2

- Saab
- 5
- 2

#### Row number 3

- Land Rover
- 17
- 15



# Functions in PHP

- A function
  - is a block of statements that can be used repeatedly in a program.
  - will not execute automatically when a page loads.
  - will be executed by a call to the function.
- A user-defined function declaration starts with the word function:

```
function functionName() {  
    code to be executed;  
}
```

- A function name must start with a letter or an underscore.  
Function names are NOT case-sensitive.

# A simple function

```
<!DOCTYPE html>
<html>
<body>

<?php
function printHello() {
    echo "Hello world!";
}

    printHello();
?>

</body>
</html>
```

Hello world!

# Function with arguments

```
<!DOCTYPE html>
<html>
<body>

<?php
function addNumbers(int $a, int $b)
{
    return $a + $b;
}
echo  addNumbers(5, 6);

?>

</body>
</html>
```

11

# Summary

- PHP
- Basic PHP Syntax
- Comments in PHP
- Variables in PHP
- Concatenation Operation
- Conditional Statements in PHP
- Loops in PHP
- Arrays in PHP
- Functions in PHP