CHAPTER – 6 – PHP CONDITIONAL STATEMENTS

12th Computer Application – Study Materials

1. Define: Conditional statement in PHP ?

- PHP also allows you to write code that perform different actions based on the results of a logical or comparative test conditions at run time.
- This means, you can create test conditions in the form of expressions that evaluates to either true or false and based on these results you can perform certain actions.

2.Define if statement in PHP .

- The if statement is used to execute a block of code only if the specified condition evaluates to true.
- This is the simplest PHP's conditional statements and can be written like:

```
if(condition){
 // Code to be executed
   }
```

The following example will output "Have a nice weekend!" if the current day is Friday:

Example

```
<?php
$d = date("D");
if($d == "Fri"){
    echo "Have a nice weekend!";
}
?>
```

3.What is if else statement in PHP ?

- The if...else statement allows you to execute one block of code if the specified condition is evaluates to true and another block of code if it is evaluates to false. It can be written, like this:

```
if(condition){
    // Code to be executed if condition is true
} else{
    // Code to be executed if condition is false
}
```

The following example will output "Have a nice weekend!" if the current day is Friday, otherwise it will output "Have a nice day!"

Example

```
<?php
$d = date("D");
if($d == "Fri"){
    echo "Have a nice weekend!";
} else{
    echo "Have a nice day!";
}
?>
```

4.List out Conditional statements in PHP ?

There are several statements in PHP that you can use to make decisions:

- The if statement
- The if...else statement
- The if...elseif....else statement
- The switch...case statement

5.Write a syntax for If else Statement in PHP.

```
if(condition){
    // Code to be executed if condition is true
} else{
    // Code to be executed if condition is false
}
```

6.Define if...elseif....else statement in PHP .

- The if...elseif...else a special statement that is used to combine multiple if...else statements.

```
if(condition1){
   // Code to be executed if condition1 is true
} elseif(condition2){
   // Code to be executed if the condition1 is false and condition2 is true
} else{
   // Code to be executed if both condition1 and condition2 are false
}
```

7.Usage of Switch statement in PHP .

- Switch statement work the same as if statements but they can check for multiple values at a time .

8.Write a syntax of Switch statement .

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

9.Compare if and if else statement

if	If else

The if statement executes some code if one condition is true.
if (condition)
{
code to be executed if condition istrue;
}
	The if....else statement executes some code if a condition is true and another code if that condition is false
	if(condition) {
	code to be executed if condition is true;
	} else
	{
	code to be executed if condition isfalse;

	}

II. Explain in Details

1.Write the features Conditional statement in PHP

- 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 - Switch statement work the same as if statements but they can check for multiple values at a time .

2.Explain Switch statement with example

Use the switch statement to select one of many blocks of code to be executed.

Syntax

```
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;
}
```

This is how it works: First we have a single expression n (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically. The default statement is used if no match is found.

Example

```
<?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!";
}
?>
```

3.Explain the concept of if elseif else statement in PHP ?

The if...elseif....else Statement

The if....elseif...else statement executes different codes for more than two conditions.

Syntax

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

The example below will output "Have a good morning!" if the current time is less than 10, and "Have a good day!" if the current time is less than 20. Otherwise it will output "Have a good night!":

Example

```
<?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!";
}
?>
```

CHAPTER – 7 – PHP LOOPING STRUCTURE

12th Computer Application – Study Materials

1. Define Looping structure in PHP .

- **PHP** is used to execute a statement or a block of statements, multiple times until and unless a specific condition is met. T
- This can be accomplished using looping statements .

2.Explain For loop in PHP. (Or) Explain concepts of for loop with example ?

- The for loop is used when you know in advance how many times the script should run.

Syntax

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

Parameters:

- *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

The example below displays the numbers from 0 to 10:

Example

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

2. What is For each loop in PHP ? (or) Discuss in detail about For each loop.

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

Syntax

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

For every loop iteration, the value of the current array element is assigned to $value and the array pointer is moved by one, until it reaches the last array element.

The following example demonstrates a loop that will output the values of the given array ($colors):

Example

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

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

3. List out the Looping structure in PHP ? (or) Explain Looping structure in PHP

In PHP, we have the following looping statements:

- 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

4. Explain the process of Do while loop.

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

Syntax

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

- The example below first sets a variable $x to 1 ($x = 1).
- Then, the do while loop will write some output, and then increment the variable $x with 1.
- Then the condition is checked (is $x less than, or equal to 5?), and the loop will continue to run as long as $x is less than, or equal to 5:

Example

```
<?php
$x = 1;

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

Notice that in a do while loop the condition is tested AFTER executing the statements within the loop. This means that the do while loop would execute its statements at least once, even if the condition is false the first time.

5. Explain array concepts in looping structure ?

- The **foreach loop** is mainly used for looping through the values of an array.
- It loops over the array, and each value for the current array element is assigned to $value, and the array pointer is advanced by one to go the next element in the array

Syntax

```
<?php foreach (array as $value){

  //code to be executed; } ?>
```

Example:

```
<?php

$salary[]=2000;

$salary[]=3000;

$salary[]=5000;

foreach($salary as $value){

  echo "Salary: $value<br>";}?>
```

Program Output:

```
Salary: 2000


Salary: 3000


Salary: 5000
```

6. Usage of for each loop in PHP

- The **foreach loop** is mainly **used** for **looping** through the values of an array.
- It **loops** over the array, and **each** value for the current array element is assigned to $value, and the array pointer is advanced by one to go the next element in the array

7.Compare For loop and For each loop

- These are the **difference between Foreach** and for **loop**.
- for **loop** is used if we know already that how many times the script should be run but**in the** case of **foreach loop**, we don't have any idea about the number of iteration.
- Also **foreach loop** is used to iterate only arrays and objects.

8.Differentiate For each and While loop

- **while** − **loops** through a block of code once, and then repeats the **loop** as long as a special condition is true.
- **foreach** − **loops** through a block of code **for** each element **in an** array.

9. Differentiate While and Do while loop

- In **while loop** the condition **is** checked at the starting of the **loop** and if it **is** true then the code inside the **loop is** executed.
- So even if the condition **is** false, the script inside the **loop is** executed at least once.
- This **is** the basic **difference between do while loop** and **while loop**