

Bachelor Of Engineering In Information Technology

Semester Six, Third Year(Even semester)

1st June 2022

24th Offline Lecture

Padre Conceicao College of Engineering
Verna Goa 403722 India

Web Technology

RC 2019-20

Unit 3

Controlling Program Flow

Chapter 5

- In every programming or scripting language , by default, the program executes in a sequential order. You can change the flow of the program by using conditional and looping statements. These statements are used when you want to specify some conditions or parameters in the program. The program executes only if the specified conditions are fulfilled. You can use the conditional and looping statements in the following conditions:
- Specified condition or value is true
- Specified condition or value is not true
- Execute a program or block of code repeatedly till the condition is fulfilled
- Transfer control of the program from one block to another

Conditional Statements

- Conditional Statements help the program to decide whether the condition given in the script is true and false. The conditions provided in the statements can be single or multiple and it allows to branch the path of execution in a script based on whether a single or multiple conditions evaluate to true or false.

Exploring different types of Conditional Statements

- In PHP, there are four types of conditional statements, which are
 1. The if Statement
 2. The if-else Statement
 3. The if-else if-else Statement
 4. The switch -case Statement

The if Statement

- The if Statements always begin with if and followed by a condition provided in parentheses. If the condition is evaluated and found true, the statement or statements immediately following the condition are executed. If the condition is found false, no change takes place and you get a blank browser window when the script is run.
- The Syntax for an if statement is as follows

```
if(expression)  
    statement
```

Script of the if statement

```
<?php  
$x=1;  
if($x==1)  
print '$x is equal to 1';  
?>
```

Output:\$x is equal to 1

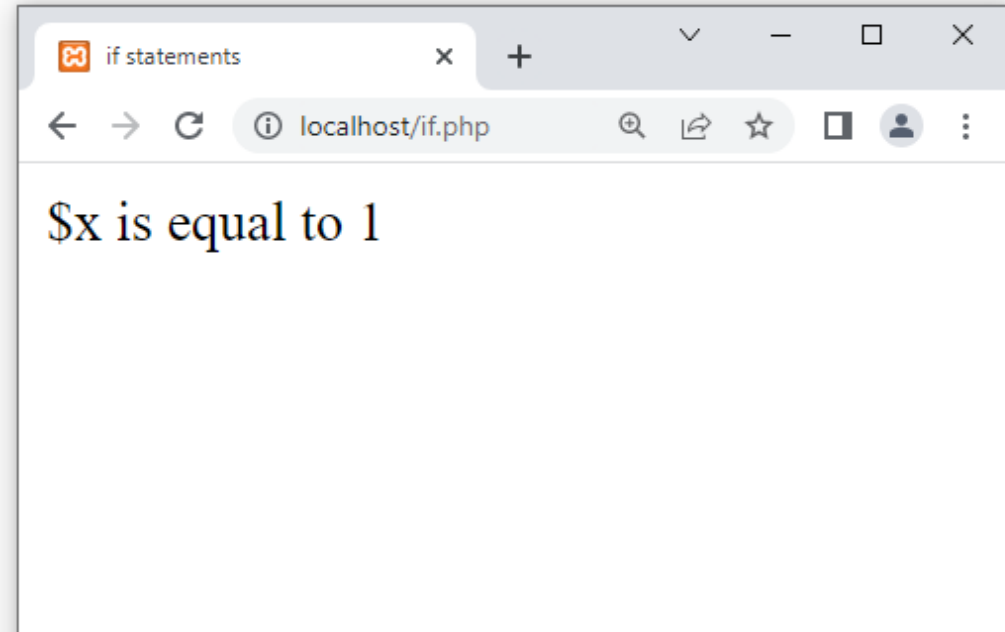
If the condition stated in the if block is, then the message \$x is equal to 1 gets displayed. If the condition is false, then the output gets the blank Web page. When you have more than one statement to be executed within a condition, it is necessary to provide them in brackets as shown

Demonstration of the if statement

if - Notepad

File Edit Format View Help

```
<html>
<head><title>if statements</title></head>
<body>
<?php
$x=1;
if($x==1)
print '$x is equal to 1';
?>
</body>
</html>
```



If condition with more than one statement

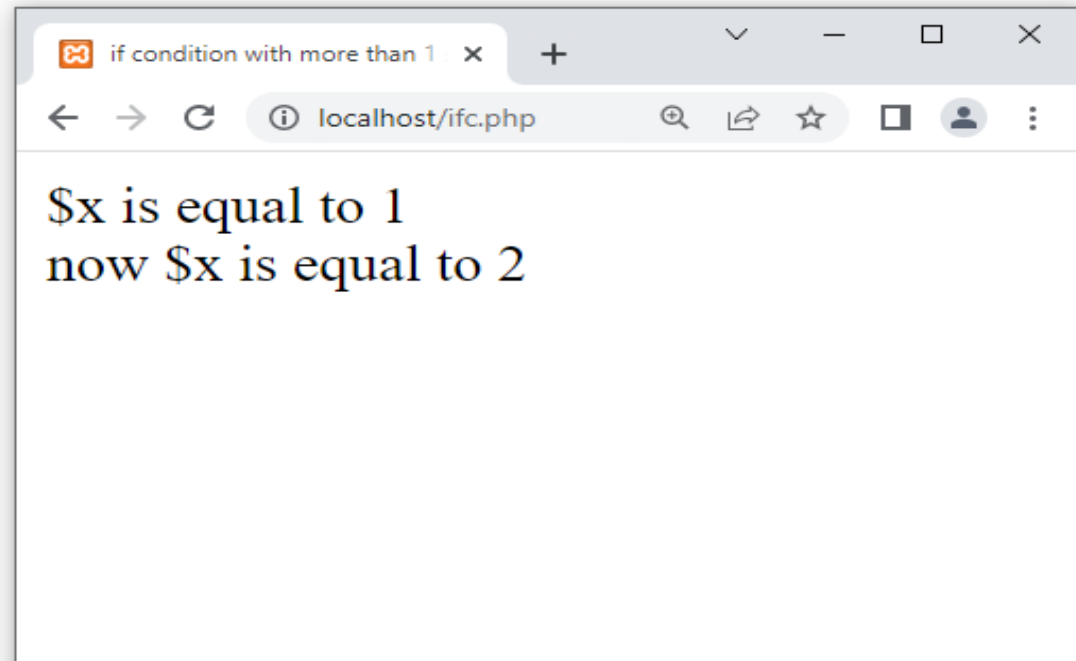
```
<?php
$x=1;
if($x==1)
{
Print '$x is equal to 1 <br>';
$x++;
Print 'now $x is equal to 2';
}
?>
```

Demonstration of if condition with more than one statement

ifc - Notepad

File Edit Format View Help

```
<html>
<head><title>if condition with more than 1 statement</title></head>
<body>
<?php
$x=1;
if($x==1)
{
print '$x is equal to 1 <br>';
$x++;
print 'now $x is equal to 2';
}
?>
</body>
</html>
```



- Output :

\$x is equal to 1

Now \$x is equal to 2

If the condition stated in the if block is true, then two messages, \$x is equal to 1 and \$x is equal to 2, get displayed. First , the PHP parser executes the \$x is equal to 1 statement, then increments the value of \$x by 1, and then executes the Now \$x is equal to 2 statement. If condition is false, then only a blank web page is shown as the output. You can write the conditional statement in multiple ways.

PHP parser checks whether or not the condition stated in the if block is true. In case the condition is true, the message gets displayed on the web page.

Demonstration of if statement in Different ways

```
<?php
$x=1;
If ($x==1) print '$x is equal to 1 <br>';
If ($x==1) {print '$x is equal to 1 <br>';}
If ($x==1) {
print '$x is equal to 1 <br>';}
?>
```

Output

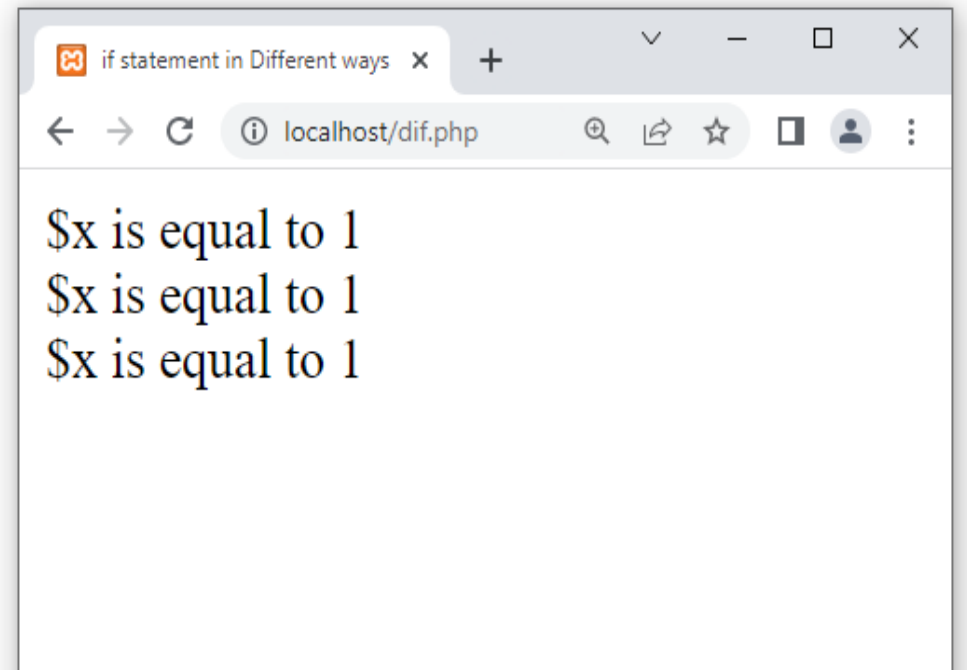
```
$x is equal to 1
$x is equal to 1
$x is equal to 1
```

Demonstration of if statement in Different ways

dif - Notepad

File Edit Format View Help

```
<html>
<head><title>if statement in Different ways</title></head>
<body>
<?php
$x=1;
if ($x==1) print '$x is equal to 1 <br>';
if ($x==1) {print '$x is equal to 1 <br>';}
if ($x==1) {
print '$x is equal to 1 <br>';}
?>
</body>
</html>
```



The if-else Statement

- The if-else Statements are used to make two types of decisions, such as true and false, based on a condition. When the condition is true, the if statement is executed and when the condition is false , the else statement is executed.
- The syntax for the if-else statement is as follows

```
if(condition)
```

```
{    statements_1}
```

```
else
```

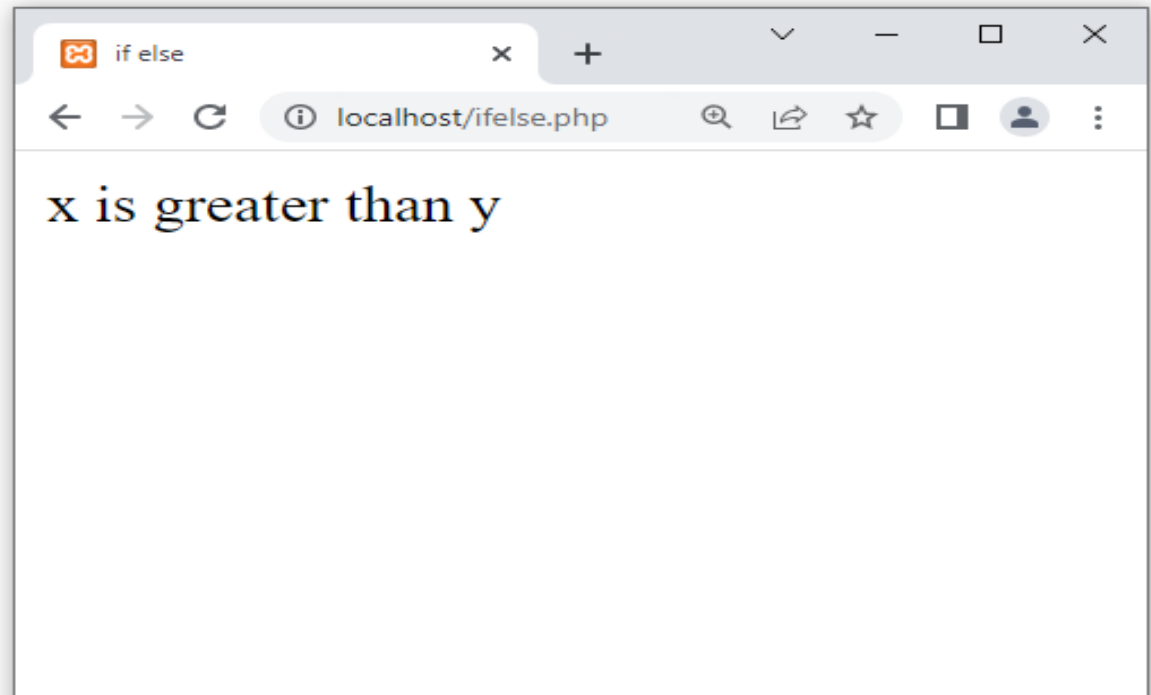
```
{    statements_2}
```

Showing the script of the if-else statement

ifelse - Notepad

File Edit Format View Help

```
<html>
<head><title>if else</title></head>
<body>
<?php
$x=91;
$y=12;
if($x > $y)
{
echo "x is greater than y";
}
else
{
echo "x is not greater than y";
}
?>
</body>
</html>
```



The if-else if-else Statement

- A combination of if-else if-else Statements is evaluated in a sequence. When the condition within the if statement is false then the elseif condition is checked and if it is found true, elseif statements are executed .Else condition is executed when all the elseif conditions are false.

The syntax for if-else if-else Statement

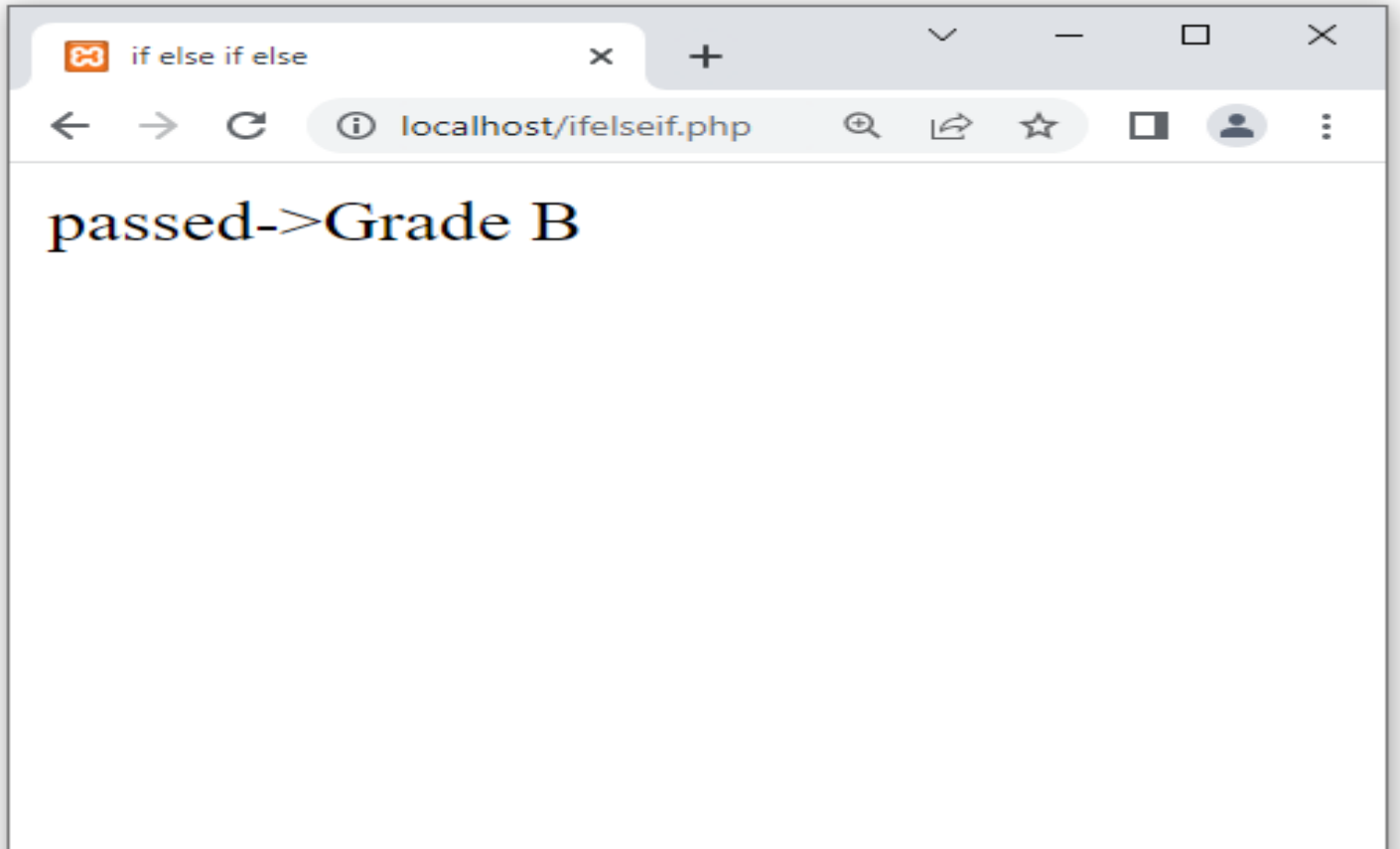
```
if(condition_1)
{
    statements_1}
elseif(condition_2)
{
    statements_2}....
elseif(condition_n_1)
{
    statements_n_1}
else
{
    statements_n}
```

Script of the if-else if-else Statement

ifelseif - Notepad

File Edit Format View Help

```
<html>
<head><title>if else if else</title></head>
<body>
<?php
$result=70;
if($result >=75)
{
echo "passed->Grade A <br/>";
}
elseif($result >=60)
{
echo "passed->Grade B <br/>";
}
elseif($result >=45)
{
echo "passed->Grade C <br/>";
}
else
{
echo "Failed <br/>";
}
?>
</body>
</html>
```



The switch Statement

- The switch Statement allows a program to evaluate a condition and match the condition's value to the statements given in case labels shown in the following syntax. If a match is found, the program executes the associated statement.
- First the program looks for a statement in the case labels that matches with the value of condition and execute the associated statements. If no matching label is found, the program looks for the optional default statement. In the absence of default statement, the program executes the statement given in the end of the switch

The syntax for the switch statement

```
Switch(condition)
{
    case label_1:
        statements_1
        break;
    case label_2:
        statements_2
        break;
    ....
    default:
        statements_n
        break;
}
```

Demonstration of switch statement

```
switch - Notepad
File Edit Format View Help
<html>
<head><title>switch statement</title></head>
<body>
<?php
$flower="rose";
switch($flower)
{
case "rose":
echo $flower."costs Rs.2.50";
break;
case "daisy":
echo $flower."costs Rs.1.25";
break;
case "lily":
echo $flower."costs Rs.1.50";
break;
default:
echo "There is no such flower in our shop";
break;
}
?>
</body>
</html>
```



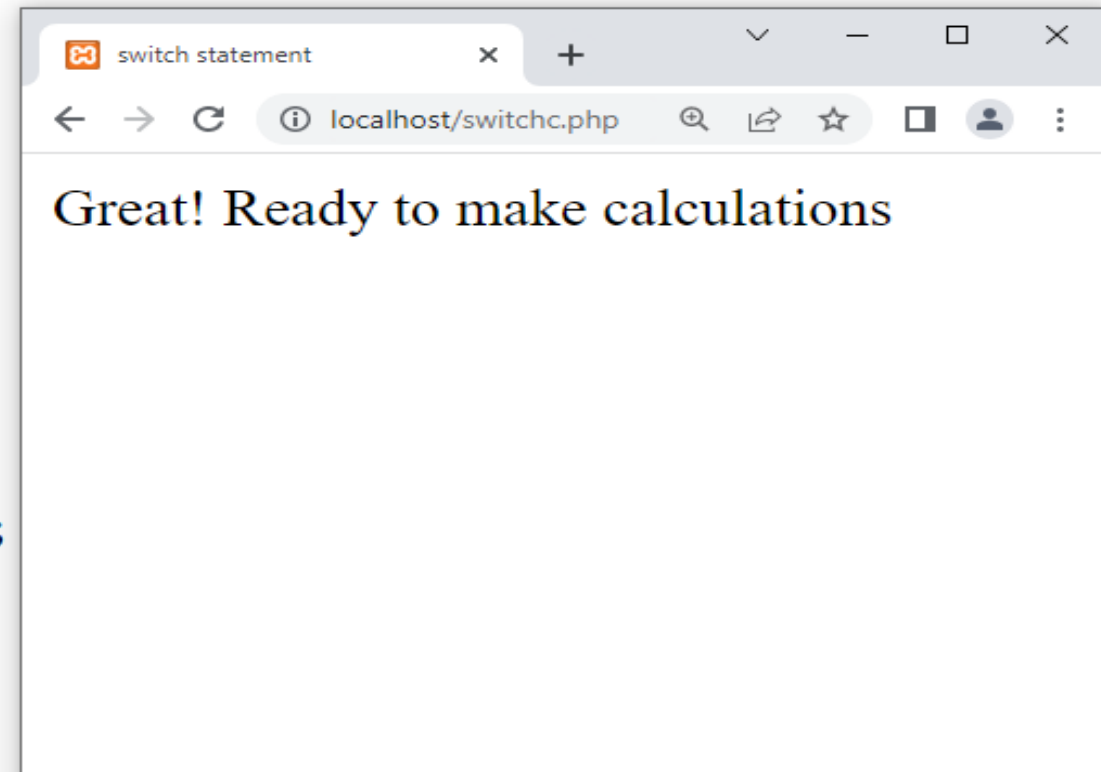
- The \$flower variable is assigned a value, rose, and then the switch condition (value of \$flower) is matched with the case value given in the switch block.
- If the case value matches with the switch condition, then the control flow is transferred to the case statement.
- In addition to match the conditions to the case labels, you can test for other conditions, such as greater than and less than relationships. The condition against which the expression is tested must be repeated in the case statement.

Demonstration of switch statement

switchc - Notepad

File Edit Format View Help

```
<html>
<head><title>switch statement</title></head>
<body>
<?php
$myNo=5;
switch($myNo)
{
case 0:
echo "zero is not a valid value";
break;
case $myNo<0:
echo "Negative numbers are not allowed";
break;
default:
echo "Great! Ready to make calculations";
break;
}
?>
</body>
</html>
```



- The \$myNo variable is assigned a value 5. The switch condition (value of \$myNo) is matched with the case value given in the switch block. If the case value matches with the switch condition, then the control flow is transferred to the case statement.
- The case statement is executed and the message is displayed.
- If the switch condition is not matched with any case value, then the control flow is transferred to the default keyword.
- The statement within the default keyword is executed and the message , "Great! Ready to make calculations", is displayed as output. If an expression matches with the values specified in more than 1 case statement, only the first one should be executed.
- When a program makes a match , PHP stops looking for more matches.

Defining the Nested If-else statements

- Conditional statements are also useful because they can be nested within each other.
- The syntax is

```
if(condition)
```

```
{
```

```
    statement;
```

```
    if(condition)
```

```
    {        Statement;    }
```

```
    else
```

```
    {        Statement;    }
```

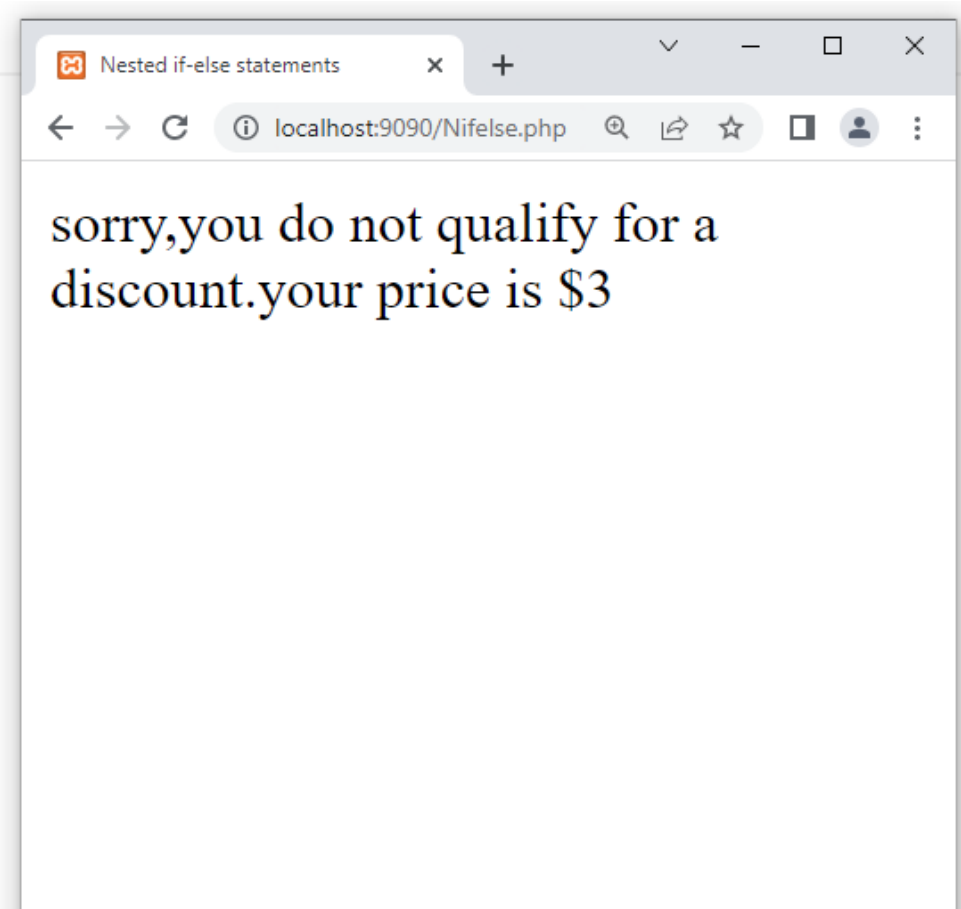
```
    else
```

```
    {        Statement;    }
```

```
}
```

Demonstration of Nested if-else statements

```
Nifelse - Notepad
File Edit Format View Help
<html>
<head><title>Nested if-else statements</title></head>
<body>
<?php
$age=30;
$price=3.00;
if($age > 65)
{
$discount=.90;
echo "You have received our senior's discount.your price is $".$price*$discount;
}
else
{
if($age < 18)
{
$discount=.95;
echo "You have received our student's discount.your price is $".$price*$discount;
}
else
{
echo "sorry,you do not qualify for a discount.your price is $".$price;
}
}
?>
</body>
</html>
```



- The \$age and \$price variables are assigned the 30 and 3.00 values, respectively. The parser checks whether or not the if condition(value of \$age) is true. If it is true, the associated block of statements is executed and the message is displayed. If the condition is not true, the control is transformed to the else block, executing the statements within the else block. This entire process is repeated for the next if-else block that is nested in the else block.

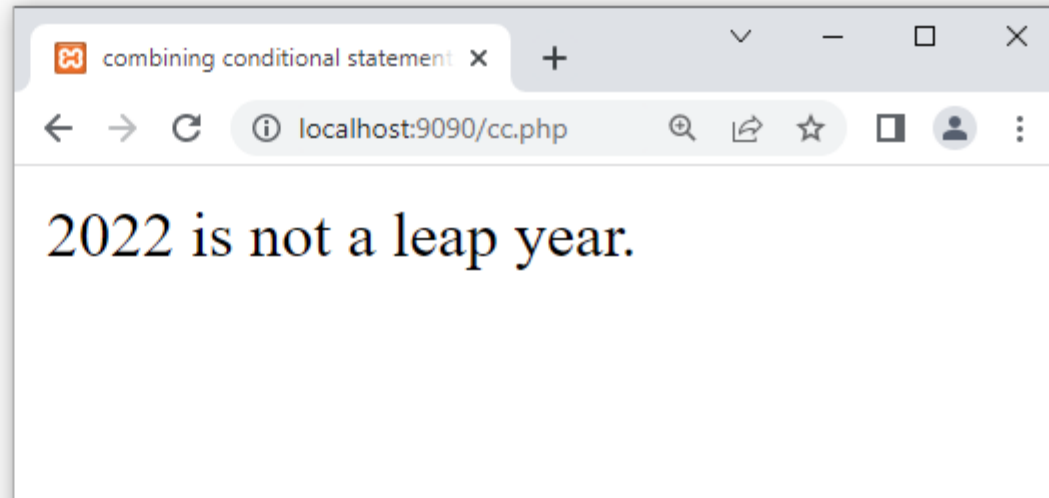
Combining Conditional Statements

- You can also combine conditional statements by using logical operator, such as the && or || operators.

cc - Notepad

File Edit Format View Help

```
<html>
<head><title>combining conditional statements</title></head>
<body>
<?php
$year=2022;
//leap year are divisible by 400 or by 4 but not 100
if(($year % 400 == 0) || ($year % 100 != 0) && ($year % 4 ==0))
{
echo "$year is a leap year.";
}
else
{
echo "$year is not a leap year.";
}
?>
</body>
</html>
```



- The value 2022 is assigned to the \$year variable. The parser checks whether or not the if condition (value of \$year) is true. The block of statements under the if statement gets executed in case the condition is true; otherwise, the control is passed to the else block. In this example, the if condition is false and the message, 2022 is not a leap year , is displayed as an output.

Looping Statements

- In programming ,it is necessary to repeat the same block of statements a given number of times, or until certain conditions are fulfilled.This can be accomplished by using looping statements. Loops execute a block of statements a specified number of times provided it is true.

Exploring Different types of Looping Statements

- In PHP , there are 4 types of looping statements
 1. The While Loop
 2. The do-while Loop
 3. The for Loop
 4. The foreach Loop

The While Loop

- The While Loops are the simplest type of loops in PHP. It executes the statements repeatedly, as long as the while condition is evaluated and found TRUE. The value of this condition is checked each time at the beginning of the loop, so even if this value changes during the execution of the nested statements, execution does not stop until the end of the process. Sometimes, if the while condition is evaluated and found FALSE from the very beginning the statements are not executed.
- Syntax is

```
while(condition)
{
statement;
}
```

Demonstration of the While Loop

while - Notepad

File Edit Format View Help

```
<html>
<head><title>While Loop</title></head>
<body>
<?php
$i=1;
while($i<=5)
{
echo "The number is ".$i."<br/>";
$i++;
}
?>
</body>
</html>
```

combining conditional statement x +

localhost:9090/while.php

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

- The while condition evaluates whether i is less than or equal to 5. The loop starts with $i=1$. The value of i gets increased by 1 each time the loop runs. When the value of i becomes greater than 5, the statement in the braces is skipped and the loop exits without performing any tasks. If the value of i is not greater than 5, the statement in the braces is executed and the loop returns to the while statement. The process repeats until i is greater than 5.

The do-while Loop

- The do-while Loops always executes the block of statements at least once and then checks the condition, and repeats the loop if the condition is true.
- The syntax for the do-while loop is as follows:

```
do  
{  
Statement;  
}  
while (condition);
```

Demonstration of the do-while Loop

dowhile - Notepad

File Edit Format View Help

```
<html>
<head><title>do while loop</title></head>
<body>
<?php
$i=1;
do
{
$i++;
echo "The number is ".$i."<br/>";}
while($i<=5);
?>
</body>
</html>
```

combining conditional statement x +

localhost:9090/dowhile

The number is 2
The number is 3
The number is 4
The number is 5
The number is 6

- The value 1 is assigned to the \$i variable. The value of i is incremented by 1 each time the loop runs. If the value of i is already greater than 5, then the loop exits. If \$i is not greater than 5, the statement in the braces executes and the loop returns to the do-while statement. The process repeats until \$i is greater than 5, and the message is display as an output.

Bachelor Of Engineering In Information Technology

Semester Six, Third Year(Even semester)

6th June 2022

28th Offline Lecture

Padre Conceicao College of Engineering
Verna Goa 403722 India

The for Loop

- The for Loop is used when you know how many times you want to execute a statement or a list of statements. For this reason, the for loop is also known as definite loop. The syntax of for loop is a bit more complex as compared to while loop. The for loop syntax is as follows:

```
for (initialization;condition;increment/decrement)
{
Statement;
}
```


Demonstration of the for Loop

for - Notepad

File Edit Format View Help

```
<html>
<head><title>for loop</title></head>
<body>
<?php
for($i=1;$i<=5;$i++)
{
echo "The number is ".$i."<br/>";
}
?>
</body>
</html>
```

Gmail x combining x +

localhost:9090/for.php

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

- The for statement takes three expressions inside its parentheses, separated by semi-colons, which are as follows:
 1. The initializing expression is executed. This expression usually initializes one or more loop counters, but the syntax allows an expression of any degree of complexity.
 2. The condition expression is evaluated. If the value of condition is true, the loop statements execute. If the value of condition is false, the for loop terminates.
 3. The update expression increment/decrement executes.
 4. The statements execute, and control returns to step 2.

In the example code, the value 1 is assigned to \$i variable. The loop iterates as long as the value of the \$i variable is less than or equal to 5. After each iteration, the value of \$i variable is incremented by 1. The statement inside the loop is executed until the condition is true.

The foreach Loop

- The foreach Loop is a variation of the for loop and allows you to iterate over elements in an array. There are two different versions of the foreach loop, which are given in the syntax

The foreach Loop syntax version 1

```
foreach (array as value)  
{  
statement;  
}
```

Demo of foreach Loop syntax version 1

foreach - Notepad

File Edit Format View Help

```
<html>
<head><title>foreach loop v1</title></head>
<body>
<?php
$email = array('PCCE Semester 6','TE IT');
foreach($email as $value)
{
echo "Processing ".$value."<br/>";
}
?>
</body>
</html>
```

foreach loop

localhost:9090/foreach.php

Processing PCCE Semester 6
Processing TE IT

The foreach Loop syntax version 2

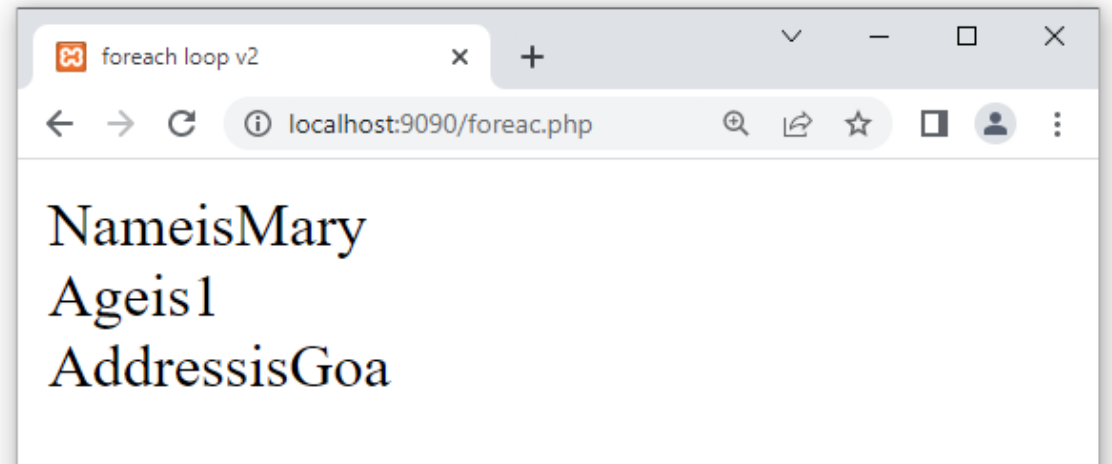
```
foreach (array as key->value)
{
statement;
}
```

Demo of foreach Loop syntax version 2

foreac - Notepad

File Edit Format View Help

```
<html>
<head><title>foreach loop v2</title></head>
<body>
<?php
$person = array('Name'=>'Mary', 'Age'=>1, 'Address'=>'Goa');
foreach($person as $key=>$value)
{
echo $key."is".$value."<br/>";
}
?>
</body>
</html>
```



Defining the Nested Looping Statement

- The placement of one loop inside the body of another loop is called nesting. When you nest two loops, the outer loop controls the number of iterations of the inner loop. You can nest all types of loops. In the demo ,two for loops are used, outer and inner. In the outer for loop, the \$num2 variable is initialized as zero and a condition is set to evaluate whether the value of the \$num2 variable is less than or equal to 2.If the specified condition is true , i.e, if the value of \$num2 variable is less than or equal to 2, the inner for loop is executed. In the inner for loop, the \$num1 variable is initialized as zero and a condition is set to check whether the value of \$num1 is less than or equal to one.

- If the condition of inner for loop is true, then the value of \$num1 is incremented by 1 and the echo statement of the inner for loop displays the result to the user.
- When the program is executed for the first time, the initial values of the \$num2 (0) and \$num1 (0) variables are displayed to the user. The inner for loop is executed until the value of the \$num1 variable does not exceed beyond 1. Once the value of \$num1 exceeds 1, the control of the program is transferred to the outer for loop and checks the condition. The process of checking the condition of the outer for loop and transferring control to the inner for loop continues as long as the value of the \$num2 variable remains less than or equal to 2.

Demo of Nested for Loop

```
Nestfor - Notepad
File Edit Format View Help

<html>
<head><title>nested loop</title></head>
<body>
<?php
for($num2=0;$num2<=2;$num2++)
{
    for($num1=0;$num1<=1;$num1++)
    {
        echo $num2."And".$num1."<br/>";
    }
}
?>
</body>
</html>
```

nested loop
localhost:9090/Nestfor.php

0And0
0And1
1And0
1And1
2And0
2And1

Break, Continue and Exit Statements

- Sometimes the loops start without any condition, and allow the statements inside the brackets to decide when to exit the loop.
- There are three special statements that can be used inside a loop , which are:
 1. Break
 2. Continue
 3. Exit

Break Statement

- The Break Statement ,when in a loop skips the remaining statements in the loop body and breaks the loop. Optionally, you can put a number after the break keyword indicating how many times you want to break the loop.
- In the demo ,the 0 value is assigned to the \$i variable. The loop iterates as long as the value of the \$i variable remains less than or equal to 5.After each iteration the value of the \$i variable is incremented by 1.The loop consists of the if condition, which contains a break statement. The execution of the loop terminates and the message is displayed as the output, if the condition is true

The syntax for break statement

```
for(condition)
{
    if (condition)
    }
    break;
    Statement;
    Statement;
    Statement;
}
```

Demo of break statement

br - Notepad

File Edit Format View Help

```
<html>
<head><title>break statemen</title></head>
<body>
<?php
for($i=0;$i<=10;$i++)
{
    if($i==3)
    {
        break;
    }
    echo "The number is ".$i;
    echo "<br/>";
}
echo "This is break statement";
?>
</body>
</html>
```

break statemen

localhost:9090/br.php

The number is 0
The number is 1
The number is 2
This is break statement

Continue Statement

- Continue Statement also skips all remaining statements in the loop for the current iteration, but returns to the top of the loop and allows it to continue running. In the demo 0 value is assigned to the \$i variable. The loop iterates as long as the value of the \$i variable remains less than or equal to 5. After each iteration, the value of the \$i variable is incremented by 1. In the demo, the for loop consists of the if condition, which contains a continue statement. The current iteration of the loop is terminated, if the condition is true. The control returns to the top of the loop, restarting the cycle once again.

The syntax for Continue Statement

```
for(condition)
{
    if (condition)
    {
        continue;
    }
    Statement;
    Statement;
    Statement;
}
```


continue - Notepad
File Edit Format View Help

```
<html>
<head><title>Continue Statement</title></head>
<body>
<?php
for($i=0;$i<5;++$i)
{
    if($i==2)
    {
        continue;
    }
    echo "The number is ".$i;
    echo "<br/>";
}
?>
</body>
</html>
```

Continue Statement

localhost/continue.php

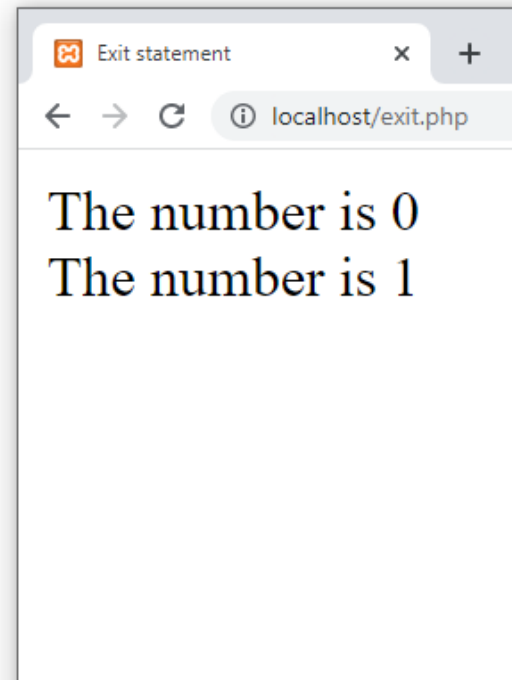
The number is 0
The number is 1
The number is 3
The number is 4

Exit Statements

- Exit Statement is used when you want to stop a program from running. It can block infinite looping statements in the program. In the demo the 0 value is assigned to the \$i variable. The loop iterates as long as the value of the \$i variable remains less than or equal to 5. After each iteration the value of the \$i variable is incremented by 1. In the demo the for loop consists of the if condition, which contains an exit statement. The execution of the script terminates and the message is displayed as the output, if the condition is true.

exit - Notepad
File Edit Format View Help

```
<html>
<head><title>Exit statement</title></head>
<body>
<?php
for($i=0;$i<5;++$i)
{
    if($i==2)
    {
        exit;
    }
    echo "The number is ".$i;
    echo "<br/>";
}
echo "This is exit statement";
?>
</body>
</html>
```



Assignment 3

- Q3) Combining Conditional Statements write a PHP program to find if 2022 is a leap year (5 marks)
- *Assignment Announced to students : AA :01st June 2022*
- *Assignment to be Submitted by students : AS: 06th June 2022*