

### Experiment No: 03

**Write a PHP program to demonstrate the use of looping structures using**

- a) while statement
- b) Do-while else statement
- c) for statement
- d) for-each statement

#### Resources required:

Hardware	Software
Computer System	Any database tools such as XAMPP

#### Practical Significance:

A loop causes a section of a program to be repeated a certain number of times. The repetition continues while the condition set for it remains true. When the condition becomes false, the loop ends and the control is passed to the statement following the loop. Loop in PHP is used to execute a statement or a block of statements, multiple times until and unless a specific condition is met. This helps the user to save both time and effort of writing the same code multiple times.

#### Theoretical Background:

##### a) While Statement:

The while statement will execute a block of code if and as long as a test condition is true. The while is an entry controlled loop statement. i.e., it first checks the condition at the start of the loop and if its true then it enters the loop and executes the block of statements, and goes on executing it as long as the condition holds true.

##### Syntax:

```
while (if the condition is true)
{
    // code is executed
}
```

##### b) do-while Statement

This is an exit control loop which means that it first enters the loop, executes the statements, and then checks the condition. Therefore, a statement is executed at least once on using the do...while loop. After executing once, the program is executed as long as the condition holds true.

##### Syntax:

```
do
{
    //code is executed
} while (if condition is true);
```

##### c) for Statement:

The for statement is used when you know how many times you want to execute a statement or a block of statements. That is, the number of iterations is known beforehand. These type of loops are also known as entry-controlled loops. There are three main parameters to the code, namely the initialization, the test condition and the counter.

**Syntax:**

```
for (initialization expression; test condition; update expression)
{
    // code to be executed
}
```

**d) For-each statement:**

**foreach loop** is used for array and objects. For every counter of loop, an array element is assigned and the next counter is shifted to the next element.

**Syntax:**

```
foreach (array_element as value)
{
    //code to be executed
}
```

**Write a program in PHP to calculate Square Root of a number.**

**Program Code:**

```
<html>
<body>
<h2>Srinivas Godihall</h2>
<?php
// PHP program to calculate the square root of a number

$n = 25;

$r = NULL;
// n = number
// r = root

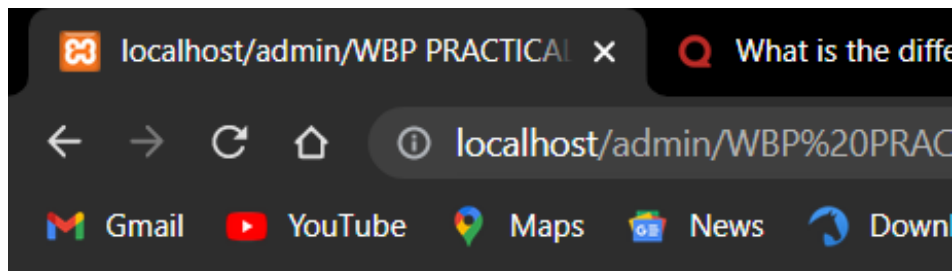
// Calculate square root of number
$r = (double)sqrt($n);

// Output
echo "Square root of " . $n . " = " . $r;

?>

</html>
</body>
```

**Output:**



**Srinivas Godihall**

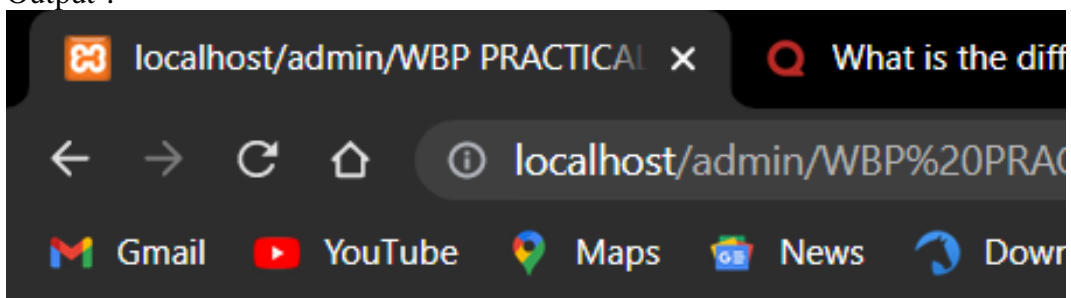
Square root of 25 = 5

### Practical related questions:

1. Write PHP program to print Fibonacci series.

```
2. <html>
3. <body>
4.
5. <?php
6. $num = 0;
7. $n1 = 0;
8. $n2 = 1;
9. echo "<h3>Fibonacci series for first 12 numbers: </h3>";
10. echo "\n";
11. echo $n1.' '.$n2.' ';
12. while ($num < 10 )
13. {
14.     $n3 = $n2 + $n1;
15.     echo $n3.' ';
16.     $n1 = $n2;
17.     $n2 = $n3;
18.     $num = $num + 1;
19. }
20. ?>
21. <h3>Srinivas Godihall</h3>
22. </html>
23. </body>
24.
```

Output-:



**Fibonacci series for first 12 numbers:**

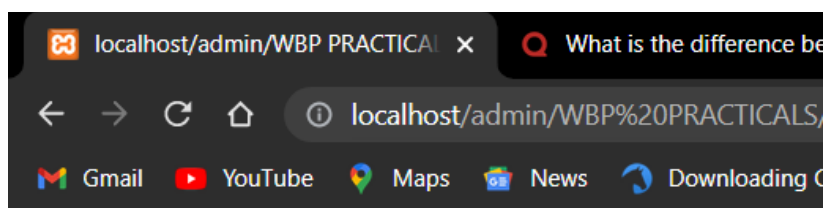
0 1 1 2 3 5 8 13 21 34 55 89

**Srinivas Godihall**

25. Write a PHP program to print prime number up to n.

```
<html>
<?php
$count = 0;
$num = 2;
while ( $count < 15 )
{
    $div_count=0;
    for ( $i=1; $i<=$num; $i++)
    {
        if (($num%$i)==0)
        {
            $div_count++;
        }
    }
    if ( $div_count<3)
    {
        echo $num." , ";
        $count=$count+1;
    }
    $num=$num+1;
}
?>
<h3>Srinivas Godihall</h3>
</html>
</body>
```

Output:-



2 , 3 , 5 , 7 , 11 , 13 , 17 , 19 , 23 , 29 , 31 , 37 , 41 , 43 , 47 ,

**Srinivas Godihall**

26. Difference between for and for-each loop.

<b>for loop</b>	<b>foreach loop</b>
The iteration is clearly visible.	The iteration is hidden.
Good performance.	Better performance.
The stop condition is specified easily.	The stop condition has to be explicitly specified.
Upon working with collections, it needs the usage of the <b>count()</b> function.	It can simply work without the usage of the <b>count()</b> method.

27. Difference between while and do-while loop.

<b>while Loop</b>	<b>do-while loop</b>
The while loop is also named as <b>entry control loop</b> .	The do-while loop is also named as <b>exit control loop</b> .
The body of the loop does not execute if the condition is false.	The body of the loop executes at least once, even if the condition is false.
Condition checks first, and then block of statements executes.	Block of statements executes first and then condition checks.
This loop does not use a semicolon to terminate the loop.	Do-while loop use semicolon to terminate the loop.

## Exercise:

1. Write the output for following script:

```
<?php
for ($x = 0; $x <= 10; print ++$x)
{
    print ++$x;
}
?>
```

Output:-

123456789101112

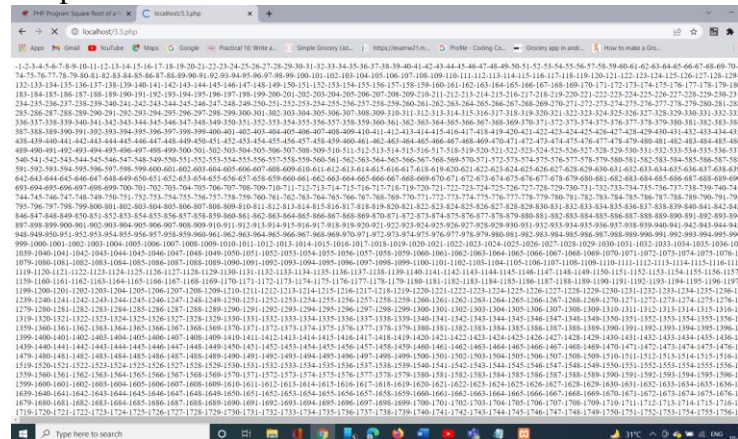
```
<?php
$i = 0;
for ($i)
{
    print $i;
}
?>
```

Output:-

**Parse error: syntax error, unexpected token ")", expecting ";;"**  
in C:\xampp\htdocs\admin\WBPRACTICALS\EXP\_2.4.php on line 3

```
<?php
for ($x = -1; $x < 10;--$x)
{
    print $x;
}
?>
```

Output:-



2. Create a script to construct the following pattern, using nested for loop.

```
<?php
for($i=0;$i<=5;$i++){
for($j=0;$j<=$i;$j++)
{
echo "*";
}
echo "<br/>";
}
?>
```

Output-:

```
*
* *
* * *
* * * *
* * * * *
```