

Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several code-lines in a script, we can use loops to perform a task like that.

By using loops statements we can decrease the number of code lines.

In PHP, there are following looping statements:

while – loops through a block of code as long as the specified condition is true

do...while – In this loop a statement is 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

while Loop- The while loop executes a block of code untill the specified condition is true.

Syntax:

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

Example:

```
$a = 5;  
while($x <= 10) { echo "The number is: $a "; $a++; } ?>
```

OUTPUT:

The number is:5

The number is: 6

The number is: 7

The number is: 8

The number is: 9

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

Syntax:

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

Example:

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

OUTPUT:

The number is: 5

The number is: 6

The number is: 7

The number is: 8

The number is: 9

for Loop: for loop is used when the user knows in advance that how many times the code should be executed.

Syntax:

```
for (initialization; condition; increment) {  
code ;  
}
```

Example:

```
for ($a= 0; $a <= 5; $a++) { echo "The number is: $a "; } ?>
```

OUTPUT:

```
0  
1  
2  
3  
4  
5
```

foreach Loop: It is used to in arrays and is used to loop through each key/value pair in an array.

Syntax:

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

In each 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.

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

OUTPUT:

red
green
blue
yellow

Related Posts:

1. How to get MySQL
2. INSTALLATION OF MYSQL ON WINDOWS
3. PHP Variables
4. PHP Data types
5. PHP Constant
6. PHP Switch Statement
7. Use of echo statement
8. PHP echo2
9. Factorial using for loop
10. PHP if statement
11. PHP if2

12. PHP if else2
13. PHP if else2
14. PHP use of function
15. PHP use of array
16. SQL in MySQL
17. LIKE IN SQL mySQL
18. Select statement in SQL
19. PHP Project
20. Write brief introduction of PHP with its origin
21. Why PHP is better than its alternatives? Explain
22. Explain interfaces to external system in PHP ?
23. What are the hardware and software requirement of PHP
24. Why is PHP known as scripting language?
25. What does a PHP Script look like? Explain
26. Describe the basics of web designing
27. What is WYSIWYG?
28. How PHP helps in designing the webpage? Give relevant example
29. How can we receive user input in PHP? Give examples
30. Explain the procedure to repeat code in PHP.
31. Explain the working of PHP script.
32. What is the basic syntax of PHP? Explain with example
33. Explain various data types in PHP ?
34. What is Google caffeine? How it works? What are its benefits
35. Explain various types of operators available in PHP.
36. How can we display data type information in PHP? Give example
37. How can we change data type? Explain
38. Explain variable manipulation in PHP

- 39. What are dynamic variables in PHP? Explain
- 40. PHP Previous Years Solved
- 41. PHP Short Notes for DAVV MBA