

Full Stack Development

Week:15

Aim:Write a Program on Looping constructs in PHP

15.a

PHP For Loop

PHP for loop can be used to traverse set of code for the specified number of times.

It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code.

It allows users to put all the loop related statements in one place. See in the syntax given below:

Syntax

1. **for**(initialization; condition; increment/decrement){
2. *//code to be executed*
3. }

Parameters

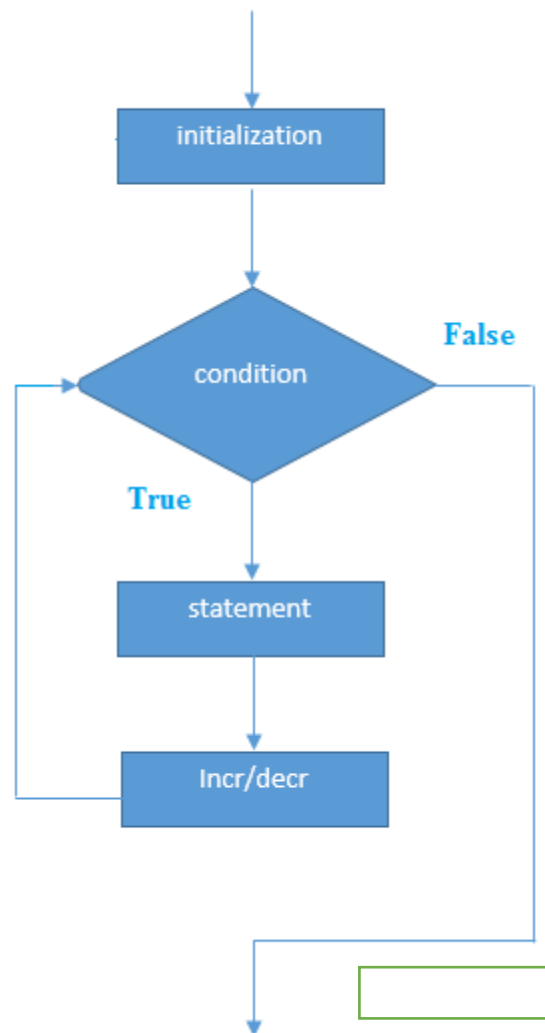
The php for loop is similar to the java/C/C++ for loop. The parameters of for loop have the following meanings:

initialization - Initialize the loop counter value. The initial value of the for loop is done only once. This parameter is optional.

condition - Evaluate each iteration value. The loop continuously executes until the condition is false. If TRUE, the loop execution continues, otherwise the execution of the loop ends.

Increment/decrement - It increments or decrements the value of the variable.

Flowchart



Code-1

```
1. <?php
2. for($n=1;$n<=10;$n++){
3. echo "$n<br/>";
4. }
5. ?>
```

Output:

```
2
3
4
5
6
7
8
9
10
```

Code-2

All three parameters are optional, but semicolon (;) is must to pass in for loop. If we don't pass parameters, it will execute infinite.

```
1. <?php
2.     $i = 1;
3.     //infinite loop
4.     for (;;) {
5.         echo $i++;
6.         echo "<br>";
7.     }
8. ?>
```

Output:

```
1
2
3
4
.
.
.
```

Code-3

PHP code to display left-aligned pyramid pattern using stars:

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

```
?>
```

Program Output:

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

15.b

PHP Nested For Loop

We can use for loop inside for loop in PHP, it is known as nested for loop. The inner for loop executes only when the outer for loop condition is found **true**.

In case of inner or nested for loop, nested for loop is executed fully for one outer for loop. If outer for loop is to be executed for 3 times and inner for loop for 3 times, inner for loop will be executed 9 times (3 times for 1st outer loop, 3 times for 2nd outer loop and 3 times for 3rd outer loop).

Code-1

1. <?php
2. **for**(\$i=1;\$i<=3;\$i++){
3. **for**(\$j=1;\$j<=3;\$j++){
4. echo "\$i \$j
";
5. }
6. }
7. ?>

Output:

```
1 1  
1 2  
1 3  
2 1  
2 2  
2 3  
3 1  
3 2
```

PHP code to display left-aligned pyramid pattern using numbers:

```
<?php
$n = 1; /*Initializing starting number*/
for ($i = 0; $i < 5; $i++)
{
    for ($j = 0; $j <= $i; $j++ )
    {
        echo $n." ";
    }
    $n = $n + 1;
    echo "<br/>";
}
?>
```

Program Output:

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

PHP code to display left-aligned pyramid pattern using letter:

```
<?php
$n = 65; /*Initializing ASCII value of (A) */
for ($i = 0; $i < 5; $i++)
{
    for ($j = 0; $j <= $i; $j++ )
    {
        echo chr($n)." ";
    }
    $n = $n + 1;
    echo "<br/>";
}
```

```
}  
?>
```

Program Output:

```
A  
B B  
C C C  
D D D D  
E E E E E
```

PHP For Each Loop

PHP for each loop is used to traverse array elements.

Syntax

1. **foreach**(\$array **as** \$var){
2. //code to be executed
3. }
4. ?>

Code-2

1. <?php
2. \$season=**array**("summer","winter","spring","autumn");
3. **foreach**(\$season **as** \$arr){
4. echo "Season is: \$arr
";
5. }
6. ?>

Output:

```
Season is: summer  
Season is: winter  
Season is: spring  
Season is: autumn
```

15.b

PHP While Loop

PHP while loop can be used to traverse set of code like for loop. The while loop executes a block of code repeatedly until the condition is FALSE. Once the condition gets FALSE, it exits from the body of loop.

It should be used if the number of iterations is not known.

The while loop is also called an **Entry control loop** because the condition is checked before entering the loop body. This means that first the condition is checked. If the condition is true, the block of code will be executed.

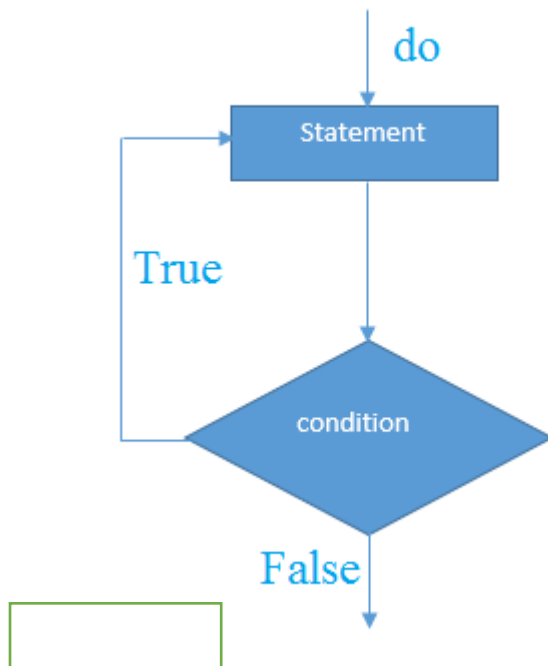
Syntax

1. **while**(condition){
2. *//code to be executed*
3. }

Alternative Syntax

1. **while**(condition):
2. *//code to be executed*
- 3.
4. **endwhile**;

PHP While Loop Flowchart



PHP While Loop Example

1. <?php
2. \$n=1;
3. **while**(\$n<=10){
4. echo "\$n
";
5. \$n++;
6. }
7. ?>

Output:

```
1
2
3
4
5
6
7
8
9
10
```


Alternative Example

1. <?php
2. \$n=1;
3. **while**(\$n<=10):
4. echo "\$n
";
5. \$n++;
6. **endwhile**;
7. ?>

Output:

```
1
2
3
4
5
6
7
8
9
10
```

Example

Below is the example of printing alphabets using while loop.

1. <?php
2. \$i = 'A';
3. **while** (\$i < 'H') {
4. echo \$i;
5. \$i++;
6. echo "
";
7. }
8. ?>

Output:

```
A
B
C
D
E
```

PHP Nested While Loop

We can use while loop inside another while loop in PHP, it is known as nested while loop.

In case of inner or nested while loop, nested while loop is executed fully for one outer while loop. If outer while loop is to be executed for 3 times and nested while loop for 3 times, nested while loop will be executed 9 times (3 times for 1st outer loop, 3 times for 2nd outer loop and 3 times for 3rd outer loop).

Example

```
1. <?php
2. $i=1;
3. while($i<=3){
4.   $j=1;
5.   while($j<=3){
6.     echo "$i $j<br/>";
7.     $j++;
8.   }
9.   $i++;
10. }
11. ?>
```

Output:

```
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
```

PHP Infinite While Loop

If we pass **TRUE** in while loop, it will be an infinite loop.

Syntax

1. **while**(true) {
2. //code to be executed
3. }

Example

1. <?php
2. **while** (true) {
3. echo "Hello Full stack developer !";
4. echo "
";
5. }
6. ?>

Output:

```
Hello Full stack developer !
Hello Full stack developer !
Hello Full stack developer !
Hello Full stack developer !
.
.
.
.
.
Hello Full stack developer !
Hello Full stack developer !
```