

2- PHP Loop: For, ForEach, While, Do While

A Loop is an Iterative Control Structure that involves executing the same number of code a number of times until a certain condition is met.

PHP For Loop

For... loops execute the block of code a specified number of times. There are basically two types of for loops;

- for
- for... each.

Let's now look at them separately. **For loop** It has the following basic **syntax**

```
<?php
for (initialize; condition; increment)
{
//code to be executed
}
?>
```

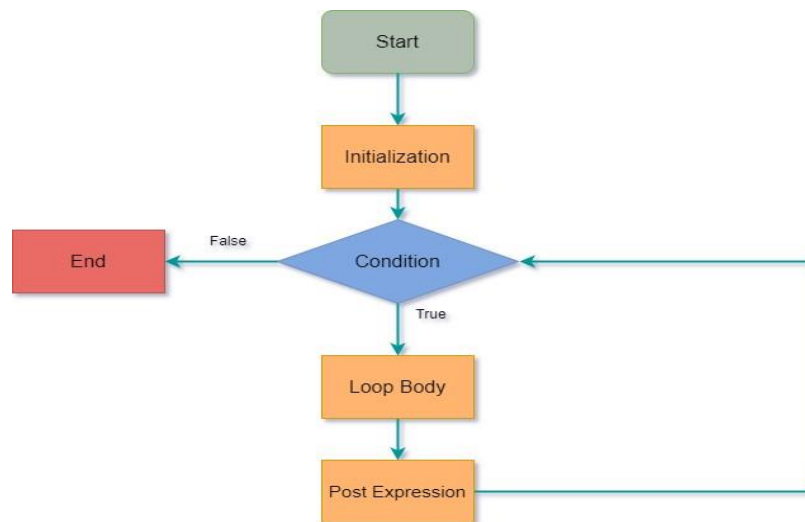
HERE,

- “**for...{...}**” is the loop block
- “**initialize**” usually an integer; it is used to set the counter's initial value.
- “**condition**” the condition that is evaluated for each php execution. If it evaluates to true, then execution of the for... loop continues. If it evaluates to false, the execution of the for... loop is terminated.
- “**increment**” is used to increment the initial value of counter integer.

How it works

The flowchart shown below illustrates how for loop in php works

Web Programming Lectures (PHP)



How to code

The code below uses the “for... loop” to print values of multiplying 10 by 0 through to 10

```
<?php
for ($i = 0; $i < 10; $i++)
{
$product = 10 * $i; echo "The product of 10 * $i
is $product <br/>";
}
?>
```

Output:

```
The product of 10 x 0 is 0
The product of 10 x 1 is 10
The product of 10 x 2 is 20
The product of 10 x 3 is 30
The product of 10 x 4 is 40
The product of 10 x 5 is 50
The product of 10 x 6 is 60
The product of 10 x 7 is 70
The product of 10 x 8 is 80
```

Web Programming Lectures (PHP)

The product of 10 x 9 is 90

PHP For Each loop

The php foreach loop is used to iterate through array values. It has the following basic syntax

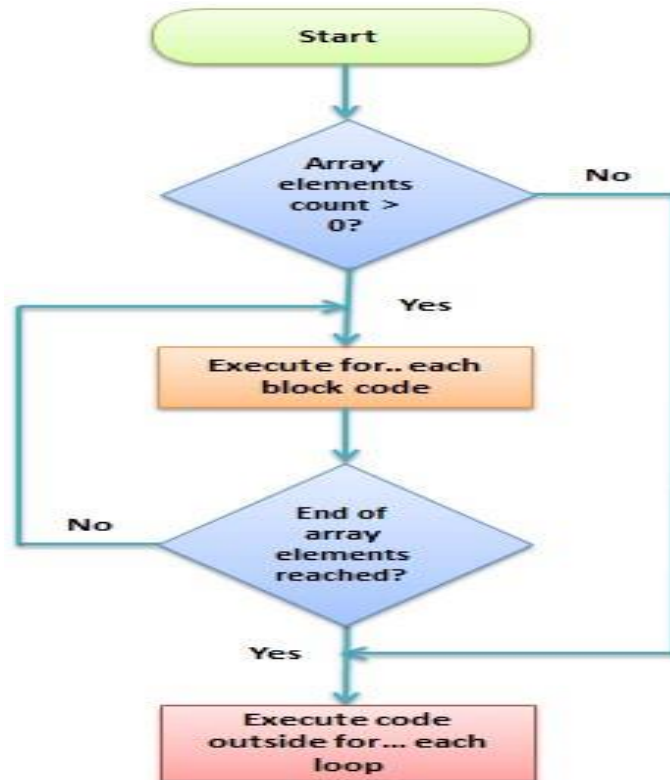
```
<?php foreach($array_variable as
$array_values)
{ block of code to be
executed
}
?>
```

HERE,

- “**foreach(...){...}**” is the foreach php loop block code
- “**\$array_data**” is the array variable to be looped through
- “**\$array_value** “ is the temporary variable that holds the current array item values.
- “**block of code...**” is the piece of code that operates on the array values

How it works The flowchart shown below illustrates how the for... each... loop works

Web Programming Lectures (PHP)



Practical examples

The code below uses for... each loop to read and print the elements of an array.

```
<?php
$animals_list = array("Lion","Wolf","Dog","Leopard","Tiger");
foreach($animals_list as $array_values) { echo $array_values .
"<br>";
}
?>
```

Output:

Lion

Wolf

Dog

Leopard

Tiger

Web Programming Lectures (PHP)

Let's look at another example that loops through an **associative array**. An associative array uses alphanumeric words for access keys.

```
<?php
$persons = array("Mary" => "Female", "John" => "Male", "Mirriam"
=> "Female"); foreach($persons as
$key => $value){ echo "$key is
$value". "<br>";
}
?>
```

The names have been used as array keys and gender as the values.

Output:

Mary is Female

John is Male

Mirriam is Female

While Loop

They are used to execute a block of code a repeatedly until the set condition gets satisfied

When to use while loops

- While loops are used to execute a block of code until a certain condition becomes true.
- You can use a while loop to read records returned from a database query.

Types of while loops

- **Do... while** – executes the block of code at least once before evaluating the condition
- **While...** – checks the condition first. If it evaluates to true, the block of code is executed as long as the condition is true. If it evaluates to false, the execution of the while loop is terminated.

Web Programming Lectures (PHP)

While loop

It has the following syntax <?php

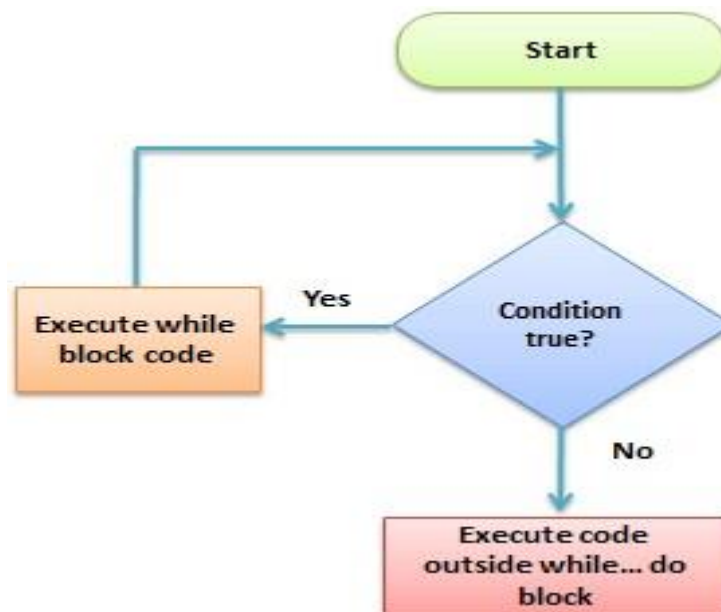
```
while (condition)
{ block of code to be
executed;
}
?>
```

HERE,

- “**while(...){...}**” is the while loop block code
- “**condition**” is the condition to be evaluated by the while loop
- “**block of code...**” is the code to be executed if the condition gets satisfied

How it works

The flow chart shown below illustrates how the while... loop works



Practical example

The code below uses the while... loop to print numbers 1 to 5.

```
<?php
```

Web Programming Lectures (PHP)

```
$i = 0; while ($i <
5){ echo $i + 1 .
"<br>";
$i++;
}
?>
```

Output:

```
1
2
3
4
5
```

PHP Do While

The difference between While... loop and Do... while loop is do... while is executed atleast once before the condition is evaluated.

Let's now look at the basic syntax of a do... while loop

```
<?php do { block of code to
be executed
} while(condition);
?>
```

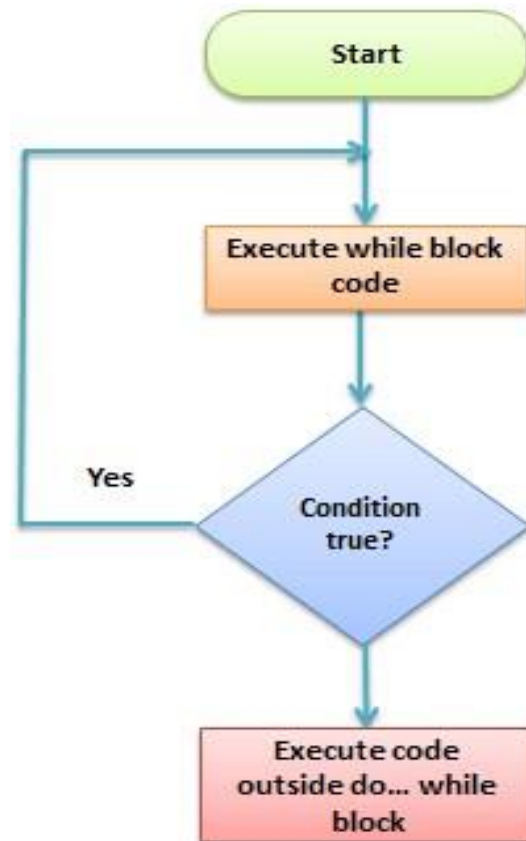
HERE,

- “**do{...} while(...)**” is the do... while loop block code
- “**condition**” is the condition to be evaluated by the while loop
- “**block of code...**” is the code that is executed at least once by the do... while loop

How it works

The flow chart shown below illustrates how the while... loop works

Web Programming Lectures (PHP)



Practical example

We are now going to modify the while... loop example and implement it using the do... while loop and set the counter initial value to 9.

The code below implements the above modified example

```
<?php
$i = 9; do { echo "$i
is"." <br>";
} while($i <
9);
?>
```

The above code outputs:

9

Note the above example outputs 9 only.

This is because the do... while loop is executed at least once even if the set condition evaluates to false.

Web Programming Lectures (PHP)

Summary

- The for... loop is used to execute a block of a specified number of times
- The foreach... loop is used to loop through arrays
- While... loop is used to execute a block of code as long as the set condition is made to be false
- The do... while loop is used to execute the block of code at least once then the rest of the execution is dependent on the evaluation of the set condition