LOOPS

Presenter:
Agubata Odinaka

What is a loop?

In computer programming a loop is a sequence of instructions that is continually repeated until a certain condition is reached.

Types of loops in php

PHP supports following four loop types.

❖**For loop -** loops through a block of code a specified number of times.

❖**while loop** - loops through a block of code if and as long as a specified condition is true.

❖**Do loop...**while - loops through a block of code once, and then repeats the loop as long as a special condition is true.

❖**Foreach loop -** loops through a block of code for each element in an array.

While loop

This is probably the most common loop, therefore we'll discuss it first.

You will give the while loop a condition to validate. As long as that condition is true, the code within the curly braces will be executed.

```
while (condition)
{
code to execute here;
}
```

While loop

```
<?PHP
$start = 1;
$times = 2;
while ($start < 11) {
    $answer = $start * $times;
    echo $start . " times " . $times . " = " . $answer . "<br>";
    $start++;
}
?>
```

While loop

Example1:

```
<?php
$a = 0;
while ($a<=10)
{
    echo "$a"."<br>";
    $a++;
}
?>
```

Alternate While loop

```
while (expr):
    statement
    ...
endwhile;
```

While loop

Example2:

```
<?php
$i = 1;
while ($i <= 10):
    echo “$i <br>”;
    $i++;
endwhile;
?>
```

do...while loop

The do...while loop is nearly identical to the while loop discussed above. The only difference is that the condition is tested after the code in question has been run once.

do…while loop

Example:

```
<?php
$i = 10;
do {
   echo $i.”<br>”;
$i--;
} while ($i > 0);
?>
```

For loop

for loops are the most complex loops in PHP. It takes three expressions.

The syntax of a for loop is:

```
for (expr1; expr2; expr3){
   statement
};
```

For loop

- ❖ The first expression (expr1) is evaluated (executed) once unconditionally at the beginning of the loop.

- ❖ In the beginning of each iteration, expr2 is evaluated. If it evaluates to TRUE, the loop continues and the nested statement(s) are executed. If it evaluates to FALSE, the execution of the loop ends.

- ❖ At the end of each iteration, expr3 is evaluated (executed).

For loop

Example:

```
<?php
for ($i = 1; $i <= 31; $i++) {
    echo ”$i<br>”;
}
?>
```

Foreach loop

The foreach loop works only on arrays, and is used to loop through each key/value pair in an array

Syntax

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

Foreach loop

For every 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.

Foreach loop

Example

```
<?php

$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {

   echo "$value <br>";

}

?>
```

Continue and break

Within loops you may need to either break out of the loop entirely or skip to the next item to be addressed in the loop. For these situations, you can use continue and break, respectively.

Continue

continue

continue is used within looping structures to skip the rest of the current loop iteration and continue execution at the condition evaluation and then the beginning of the next iteration.

Continue

```
<?php
for($i=0;$i<=10;$i++){
If($i==2)continue;
Echo “$i<br>”;
}
?>
```

Break

break ends execution of the current for, foreach, while, do-while or switch structure.

Example:1

```
<?php
for($i=0;$i<=5;$i++){
    if($i==2)break;
    echo $i.'<br>';
}
?>
```

Questions?