

11 PHP Looping - While Loops

Loops execute a block of code a specified number of times, or while a specified condition is true.

11.1. PHP Loops

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 almost equal lines in a script we can use loops to perform a task like this.

In PHP, we have the following looping statements:

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

11.2. The while Loop

The while loop executes a block of code while a condition is true.

Syntax

```
while (condition)
{
    code to be executed;
}
```

Example 1

The example below first sets a variable *i* to 1 (`$i=1;`).

Then, the while loop will continue to run as long as *i* is less than, or equal to 5. *i* will increase by 1 each time the loop runs:

```
<html>
<body>

<?php
$i=1;
```

```
while ($i<=5)
{
    echo "The number is " . $i . "<br>";
    $i++;
}
?>

</body>
</html>
```

Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

11.3. The do...while Statement

The do...while statement will always execute the block of code once, it will then check the condition, and repeat the loop while the condition is true.

Syntax

```
do
{
    code to be executed;
}
while (condition);
```

Example 2

The example below first sets a variable *i* to 1 (\$i=1;).

Then, it starts the do...while loop. The loop will increment the variable *i* with 1, and then write some output. Then the condition is checked (is *i* less than, or equal to 5), and the loop will continue to run as long as *i* is less than, or equal to 5:

```
<html>
<body>
```

```
<?php
$i=1;
do
{
    $i++;
    echo "The number is " . $i . "<br>";
}
while ($i<=5);
?>

</body>
</html>
```

Output:

```
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
```

The for loop and the for-each loop will be explained in the next chapter.