

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

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

The `while` loop - Loops through a block of code as long as the specified condition is true.

---

# The PHP while Loop

The `while` loop executes a block of code as long as the specified condition is true.

## Syntax

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

```
<html>
<body>

<?php
$x = 1;

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

</body>
</html>
```

The number is: 1

The number is: 2

The number is: 3

The number is: 4

The number is: 5

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

---

# The PHP do...while Loop

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

## Syntax

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

```
<html>
<body>

<?php
$x = 1;

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

</body>
</html>
```

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

The `for` loop - Loops through a block of code a specified number of times.

---

# The PHP for Loop

The `for` loop is used when you know in advance how many times the script should run.

## Syntax

```
for (init counter; test counter; increment counter) {  
    code to be executed for each iteration;  
}
```

Parameters:

- *init counter*: Initialize the loop counter value
- *test counter*: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- *increment counter*: Increases the loop counter value

```
<!DOCTYPE html>
<html>
<body>

<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0  
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5  
The number is: 6  
The number is: 7  
The number is: 8  
The number is: 9  
The number is: 10

The `foreach` loop - Loops through a block of code for each element in an array.

---

# The PHP 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;  
}
```

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.



```
<html>
<body>

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

foreach ($colors as $value) {
    echo "$value <br>";
}
?>

</body>
</html>
```

red  
green  
blue  
yellow