

For Loop

Used when you know in advance how many times you want to execute a statement or a block of statements.

```
for($x = 0; $x <= 5; $x++) {  
    echo "The number is: $x <br>";  
}
```

While Loop

It executes a block of code as long as the specified condition is true.

```
$x = 1;  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}
```

Do...While Loop

Similar to the while loop, but the condition is tested after the execution of the loop's statements.

```
$x = 1;  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);
```

Foreach Loop

Used for looping through arrays. For each loop iteration, the value of the current array element is assigned to the value variable and the array pointer is moved by one, until it reaches the last array element.

```
$colors = array("red", "green", "blue", "yellow");  
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
  
$age = array("Peter" => "35", "Ben" => "37", "Joe" => "43");  
foreach($age as $key => $value) {  
    echo "$key is $value years old. <br>";  
}
```

Break

It is used to exit a loop prematurely. Once break is encountered, the loop terminates and the program resumes execution after the loop.

```
for ($i = 0; $i < 10; $i++) {  
    if ($i == 3) {  
        break;  
    }  
    echo $i . "<br>";  
}
```

Continue

The continue statement is used to skip the rest of the current loop iteration and proceed to the evaluation of the loop condition for the next iteration.

```
for ($i = 0; $i < 10; $i++) {  
    if ($i == 3) {  
        continue;  
    }  
    echo $i . "<br>";  
}
```

#php