

LOOPING STATEMENTS IN JAVASCRIPT

Sai vardhan T

LOOPING STATEMENTS IN JS

- WHILE
- DO_WHILE
- FOR
- FOR_IN
- FOR_OF

WHILE LOOP

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

Syntax:

javascript

```
while (condition) {  
    // code block to be executed  
}
```

Example:

javascript

```
var count = 0;  
while (count < 5) {  
    console.log("Count: " + count);  
    count++;  
}
```

EXAMPLE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>While Loop Example</title>
</head>
<body>
  <h1>While Loop Example</h1>
  <div id="output"></div>
  <script>
    var outputDiv = document.getElementById('output');
    var counter = 1;
    while (counter <= 7) {
      outputDiv.innerHTML += 'Number: ' + counter + '<br>';
      counter++;
    }
  </script>
</body>
</html>
```

DO - WHILE

- The do...while loop is similar to the while loop, but it always executes the block of code at least once before checking the condition.

Syntax:

javascript

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

Example:

javascript

```
var count = 0;  
do {  
    console.log("Count: " + count);  
    count++;  
} while (count < 5);
```

EXAMPLE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>While Loop Example</title>
</head>
<body>
  <h1>While Loop Example</h1>
  <div id="output"></div>
  <script>
    var outputDiv = document.getElementById('output');
    var counter = 1;
    do {
      outputDiv.innerHTML += 'Number: ' + counter + '<br>';
      counter++;
    }
    while (counter <= 7);
  </script>
</body>
</html>
```

FOR LOOP

- The for loop executes a block of code a specified number of times.

Syntax:

javascript

```
for (initialization; condition; increment/decrement) {  
    // code block to be executed  
}
```

Example:

javascript

```
for (var i = 0; i < 5; i++) {  
    console.log("Index: " + i);  
}
```

EXAMPLE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>For Loop Example</title>
</head>
<body>
  <h1>For Loop Example</h1>
  <ul id="list-container"></ul>

  <script>
    var listContainer = document.getElementById('list-container');
    for (var i = 1; i <= 5; i++) {
      var listItem = document.createElement('li');
      listItem.textContent = 'Item ' + i;
      listContainer.appendChild(listItem);
    }
  </script>
</body>
</html>
```


FOR IN LOOP

- The for...in loop iterates over the properties of an object.

Syntax:

javascript

```
for (variable in object) {  
    // code block to be executed  
}
```

Example:

javascript

```
var person = {  
    name: "John",  
    age: 30,  
    city: "New York"  
};  
  
for (var key in person) {  
    console.log(key + ": " + person[key]);  
}
```

EXAMPLE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>For...in Loop Example</title>
</head>
<body>
  <h1>For...in Loop Example</h1>
  <ul id="list-container"></ul>
  <script>
    var listContainer = document.getElementById('list-container');
    var person = {
      name: "John",
      age: 30,
      city: "New York"
    };
    for (var key in person) {
      var listItem = document.createElement('li');
      listItem.textContent = key + ': ' + person[key];
      listContainer.appendChild(listItem);
    }
  </script>
</body>
</html>
```

FOR OF LOOP

- The for...of loop iterates over iterable objects such as arrays, strings, maps, sets, etc.

Syntax:

javascript

```
for (variable of iterable) {  
    // code block to be executed  
}
```

Example:

javascript

```
var colors = ["red", "green", "blue"];  
  
for (var color of colors) {  
    console.log(color);  
}
```

EXAMPLE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>For...of Loop Example</title>
</head>
<body>
  <h1>For...of Loop Example</h1>
  <div id="output"></div>_
  <script>
    var outputDiv = document.getElementById('output');
    var colors = ["red", "green", "blue", "yellow", "orange"];
    for (var color of colors) {
      outputDiv.innerHTML += color + '<br>';
    }
  </script>
</body>
</html>
```

JUMP STATEMENTS

- The jump statements are used to control the loop
- If we want to control the loop once the loop is going on, then these jump statement comes into execution.
- **Type of Jump statements:**
 - Break
 - Continue

Break Statement

- The break statement, used alone, causes the innermost enclosing loop or switch statement to exit immediately.
- If the break statement is involved inside the loop, then the pointer will immediately come out of the Loop block
- Its syntax is simple:
- **Example:**

```
break;
```

```
for(var i = 0; i < a.length; i++) {  
    if (a[i] == target) break;  
}
```

Continue Statement

- The continue statement is similar to the break statement.
- Instead of exiting a loop, however, continue to restart a loop at the next iteration.
- Its Syntax is simple:
- **Example:**

```
continue;
```

```
for(i = 0; i < data.length; i++) {  
    if (!data[i]) continue; // Can't proceed with undefined data  
    total += data[i];  
}
```