1.6. Loops in Javascript

In programming, loops are used to repeat a block of code. For example, if you want to show a message 100 times, then you can use a loop.

▼ Javascript for loop

The syntax of the `for` loop is:

```
for (initialExpression; condition; updateExpression) {
    // for loop body
}
```

Here,

1. The **initialExpression** initializes and/or declares variables and executes only once.
2. The **condition** is evaluated.
   - If the condition is `false`, the `for` loop is terminated.
   - If the condition is `true`, the block of code inside of the `for` loop is executed.
3. The **updateExpression** updates the value of **initialExpression** when the condition is `true`.
4. The **condition** is evaluated again. This process continues until the condition is `false`.

```
// program to display text 5 times
const n = 5;

// looping from i = 1 to 5
for (let i = 1; i <= n; i++) {
    console.log(`I love JavaScript.`);
}
```

▼ Javascript while loop

The syntax of the `while` loop is:

```
while (condition) {
    // body of loop
}
```

Here,

1. A `while` loop evaluates the **condition** inside the parenthesis `()`.
2. If the **condition** evaluates to `true`, the code inside the `while` loop is executed.
3. The **condition** is evaluated again.
4. This process continues until the **condition** is `false`.
5. When the **condition** evaluates to `false`, the loop stops.

▼ Javascript do-while loop

The syntax of `do...while` loop is:

```
do {
    // body of loop
} while(condition)
```

Here,

1. The body of the loop is executed at first. Then the **condition** is evaluated.
2. If the **condition** evaluates to `true`, the body of the loop inside the `do` statement is executed again.
3. The **condition** is evaluated once again.
4. If the **condition** evaluates to `true`, the body of the loop inside the `do` statement is executed again.
5. This process continues until the **condition** evaluates to `false`. Then the loop stops.

💡 **Note**: do...while loop is similar to the while loop. The only difference is that in do...while loop, the body of loop is executed at least once.

▼ break and continue

The `break` statement is used to terminate the loop immediately when it is encountered.

The `continue` statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration.

▼ Javascript for...in loop

The `for..in` loop in JavaScript allows you to iterate over all property keys of an object.

The syntax of the `for...in` loop is:

```
const student = {
    name: 'Monica',
    grade: 7,
    age: 12
}

// using for...in
for ( let key in student ) {

    // display the properties
    console.log(`${key} => ${student[key]}`);
}
```

In each iteration of the loop, a key is assigned to the key variable. The loop continues for all object properties.

▼ Javascript for...of loop

The `for..of` loop in JavaScript allows you to iterate over characters in string or values in an array

The syntax of the `for...of` loop is:

```
const str = "hello";
for (const c of str) {
    console.log(c);
}
// Output: 'h', 'e', 'l', 'l', 'o'

const count = ["one", "two", "three", "four"];
for (const num of count) {
    console.log(num);
}
// Output: "one", "two", "three", "four"]
```

Aspect	Java	JavaScript
For Loop	Traditional: `for(init; condition; increment)`. Enhanced: `for (type item : collection)`.	Traditional: `for(init; condition; increment)`. Array-specific: `for...of`. Object-specific: `for...in`.
While Loop	Supported: `while (condition)` and `do-while (condition)`.	Same syntax and behavior: `while` and `do...while`.
Break/Continue	`break` exits the loop; `continue` skips the current iteration.	Same functionality as Java.

Assignment

1. Write a JavaScript function that checks whether a passed string is palindrome or not.
2. Learn about various String methods in Javascript: https://www.programiz.com/javascript/library/string
3. Learn about various Math methods in Javascript: https://www.programiz.com/javascript/library/math