Lesson 4: Control Flow and Loops

Objective

By the end of this lesson, students will understand how to use control flow statements and loops in JavaScript to create dynamic and interactive programs.

1. Introduction to Control Flow

Control flow statements allow your code to make decisions and execute different actions depending on certain conditions.

- **`if`, `else if`, and `else` Statements**
  - **`if` Statement**: Executes a block of code if a specified condition is true.

```
let temperature = 30;

if (temperature > 25) {
    console.log("It's a hot day!");
}
```

  - **`else if` Statement**: Provides an additional condition if the first condition is false.

```
if (temperature > 25) {
    console.log("It's a hot day!");
} else if (temperature > 15) {
    console.log("It's a warm day.");
}
```

  - **`else` Statement**: Executes a block of code if all preceding conditions are false.

```
if (temperature > 25) {
    console.log("It's a hot day!");
} else if (temperature > 15) {
    console.log("It's a warm day.");
} else {
    console.log("It's a cool day.");
}
```

- **Comparison and Logical Operators**
  - **Comparison Operators**: Used to compare two values. Examples include `==`, `===`, `!=`, `!==`, `<`, `>`, `<=`, `>=`.
  - **Logical Operators**: Used to combine multiple conditions. Examples include:
    - `&&` **(AND)**: Both conditions must be true.
    - `||` **(OR)**: At least one condition must be true.
    - `!` **(NOT)**: Inverts the truth value of the condition.

```
let isSunny = true;
let temperature = 30;

if (isSunny && temperature > 25) {
    console.log("Let's go to the beach!");
}
```

2. Introduction to Loops

Loops allow you to execute a block of code multiple times, making them essential for repetitive tasks.

- **`for` Loop**
  - The `for` loop is used when you know how many times you want to run a block of code.

```
for (let i = 0; i < 5; i++) {
    console.log("Hello, World!");
}
```

- **`while` and `do...while` Loops**
  - **`while` Loop**: Executes a block of code as long as a specified condition is true.

```
let i = 0;

while (i < 5) {
    console.log("Hello, World!");
    i++;
}
```

  - **`do...while` Loop**: Executes a block of code once, and then repeats it as long as the condition is true.

```
let i = 0;

do {
    console.log("Hello, World!");
    i++;
} while (i < 5);
```

- **`for...in` and `for...of` Loops**
  - **`for...in` Loop**: Iterates over the properties of an object.

```
let person = {name: "John", age: 30, city: "New York"};

for (let key in person) {
    console.log(key + ": " + person[key]);
}
```

  - **`for...of` Loop**: Iterates over the values of an iterable object like an array.

```
let fruits = ["apple", "banana", "cherry"];

for (let fruit of fruits) {
    console.log(fruit);
}
```

3. Hands-On Practice

Exercise 1: Even or Odd

- Write a script that determines whether a given number is even or odd.

```
let number = 4;

if (number % 2 === 0) {
    console.log(number + " is even.");
} else {
    console.log(number + " is odd.");
}
```

Exercise 2: Counting with Loops

- Create a loop that prints the numbers from 1 to 10.

```
for (let i = 1; i <= 10; i++) {
    console.log(i);
}
```

Exercise 3: Sum of Even Numbers

- Write a function that takes an array of numbers and returns the sum of all even numbers.

```
function sumEvenNumbers(numbers) {
    let sum = 0;

    for (let number of numbers) {
        if (number % 2 === 0) {
            sum += number;
        }
    }

    return sum;
}

let numbersArray = [1, 2, 3, 4, 5, 6];
console.log(sumEvenNumbers(numbersArray));  // Output: 12
```

4. Homework/Assignment

Assignment 1: Highest Grade

- Write a script that loops through an array of student grades and determines the highest grade.

```
let grades = [88, 92, 79, 85, 90];
let highestGrade = 0;

for (let grade of grades) {
    if (grade > highestGrade) {
        highestGrade = grade;
    }
}

console.log("The highest grade is: " + highestGrade);
```

Assignment 2: Filter Odd Numbers

- Create a function that takes an array of numbers and returns a new array with only the odd numbers.

```
function filterOddNumbers(numbers) {
    let oddNumbers = [];

    for (let number of numbers) {
        if (number % 2 !== 0) {
            oddNumbers.push(number);
        }
    }

    return oddNumbers;
}

let numbersArray = [1, 2, 3, 4, 5, 6];
console.log(filterOddNumbers(numbersArray));  // Output: [1, 3, 5]
```

5. Recommended Resources

Documentation and Tutorials

- MDN Web Docs: Control Flow
  - A comprehensive guide on control flow and error handling in JavaScript.
- JavaScript.info: Loops
  - Detailed explanations and examples of different types of loops in JavaScript.
- MDN Web Docs: Loops and Iteration
  - An in-depth look at loops and iteration, including different types of loops and when to use them.

Video Tutorials

- Programming with Mosh: JavaScript Control Flow
  - A video tutorial explaining control flow concepts in JavaScript with practical examples.
- Traversy Media: JavaScript Loops Tutorial

- A comprehensive guide to understanding and using loops in JavaScript, with code examples.