1. **Branching (Switch)**: Branching refers to choosing between different paths or options based on conditions. It's like choosing between different routes when driving based on traffic conditions or road signs.

```
Task Solution:
var dayNumber = 2; // Assume the user enters 2 for Monday

switch (dayNumber) {
    case 1:
        console.log("Sunday");
        break;
    case 2:
        console.log("Monday");
        break;
    case 3:
        console.log("Tuesday");
        break;
    case 4:
        console.log("Wednesday");
        break;
    case 5:
        console.log("Thursday");
        break;
    case 6:
        console.log("Friday");
        break;
    case 7:
        console.log("Saturday");
        break;
    default:
        console.log("Invalid day number");
}
```

While Loop:

- The while loop is a control flow statement that repeatedly executes a block of code as long as a specified condition is true.
- It consists of only the condition, which is evaluated before each iteration. If the condition is true, the loop continues; otherwise, it exits.

```
var sumWhile = 0;
const endLevel = 20;
var j = 1;

while (sumWhile < endLevel) {
    sumWhile += j;
    j++;
    console.log("While Loop:" + j);
}
console.log("Sum using while loop:", sumWhile);
```

Do-While Loop:

- The do-while loop is a control flow statement that executes a block of code at least once, and then repeats the loop as long as a specified condition is true.
- It consists of the code block followed by the **do** keyword and the condition, which is evaluated after each iteration. If the condition is true, the loop continues; otherwise, it exits.

```
var sumDoWhile = 0;
const endLevel = 20;
var k = 1;

console.log("Do-While Loop:");
do {
   sumDoWhile += k;
   k++;
} while (sumDoWhile < endLevel);

console.log("Sum using do-while loop:", sumDoWhile);
```

For Loop:

- The for loop is a control flow statement that allows you to execute a block of code repeatedly based on a specified number of iterations or a predefined range.
- It consists of three parts:
    - Initialization: Setting an initial value before starting the loop.
    - Condition: Checking the condition before each iteration. If the condition is true, the loop continues; otherwise, it exits.
    - Increment/Decrement: Modifying the loop control variable after each iteration.

```
var sumFor = 0;
const endLevel = 20;

for (var i = 1; i <= endLevel; i++) {
   sumFor += i;
   console.log("For Loop:" + i);

}

console.log("Sum using for loop:", sumFor);
```

Interview Questions:

Switch Statement

What is a switch statement in JavaScript?

Answer: A switch statement allows a variable to be tested for equality against a list of values, each with its own case, and executes the corresponding block of code when a match is found.

How does the break statement function within a switch statement?

- **Answer**: The break statement terminates the current case block and prevents the execution from falling through to subsequent cases.

What is a while loop in JavaScript?

- **Answer**: A while loop repeatedly executes a block of code as long as a specified condition evaluates to true.

What happens if the condition in a while loop always evaluates to true? Provide an example.

- **Answer**: The loop will become an infinite loop and will continue to execute indefinitely unless interrupted.

What is the difference between a while loop and a do...while loop?

- **Answer**: A while loop checks the condition before executing the loop body, whereas a do...while loop executes the loop body at least once before checking the condition.

When would you choose a do...while loop over a while loop?

- **Answer**: Use a do...while loop when you need to ensure that the loop body is executed at least once regardless of the condition.

What is a for loop in JavaScript?

- **Answer**: A for loop is used to repeat a block of code a certain number of times, with the syntax including initialization, condition, and increment/decrement expressions.

Can you explain the difference between continue and break statements in loops with examples?

- **Answer**: The break statement exits the loop entirely, while the continue statement skips the rest of the current loop iteration and proceeds to the next iteration.

```
for (let i = 0; i < 5; i++) {
    if (i === 2) {
        continue; // Skips the rest of the loop iteration when i is 2
    }
    if (i === 4) {
        break; // Exits the loop when i is 4
    }
    console.log(i);
}
// Output: 0 1 3
```