[8] BREAK/CONTINUE AND LABELED STATEMENTS

In JavaScript, the break and continue statements are used within loops to control the flow of execution.

Break Statement

The break statement is used to exit a loop prematurely, before its normal termination condition is met. It is commonly used to exit a loop when a specific condition is satisfied.

```
for (let i = 1; i <= 5; i++) {
if (i === 3) {
break;
}
console.log(i);
}
```

Continue Statement

The continue statement is used to skip the current iteration of a loop and proceed to the next iteration. It is commonly used when you want to skip certain iterations based on a condition without exiting the loop entirely. Example:

```
for (let i = 1; i <= 5; i++) {
if (i === 3) {
continue;
}
console.log(i);
}
```

LABELED STATEMENTS

In JavaScript, a labeled statement is a way to mark a specific statement in your code so that you can refer to it later using a label. Labels are identifiers followed by a colon (:), and they are usually used with control flow statements like loops (for, while, etc.) and conditional statements (if, switch, etc.).

The primary use of labeled statements is to control the flow of your code, especially when you have nested loops or conditional statements. You can use the break and continue statements with labels to specify which loop or conditional statement you want to exit or continue.

Labeled statements can be helpful for managing complex control flow scenarios, but they should be used judiciously as excessive use of labels can make your code less readable and harder to maintain

```
---Syntax
labelName: statement

---Example[1]
outerLoop: for (let i = 0; i < 3; i++) {
innerLoop: for (let j = 0; j < 3; j++) {
if (i === 1 && j === 1) {
// This will break out of the inner loop when i is 1 and j is 1
break innerLoop;
}
console.log(`i: ${i}, j: ${j}`);
}
}
```

In this example, we have two nested loops, outerLoop and innerLoop, both labeled.

- When the condition i === 1 && j === 1 is met, the break innerLoop; statement is executed, which breaks out of the inner loop labeled innerLoop.
- Without labels, it would only break out of the innermost loop, but with labels, you can control which loop is affected.
- Similarly, you can use labels with the continue statement to skip to the next iteration of a specific loop.