Decision-Making and Control Statements in JavaScript

An Overview of Conditional and Looping Constructs

1. Decision-Making Statements

- These statements execute specific blocks of code based on conditions.

1.1 if Statement

- Executes a block of code if a specified condition is true.

Example:

```
let age = 18;
if (age >= 18) {
   console.log('You are eligible to vote.');
}
```

1.2 if...else Statement

- Executes one block if the condition is true, another if false.

Example:

```
let age = 16;
if (age >= 18) {
   console.log('You are eligible to vote.');
} else {
   console.log('You are not eligible to vote.');
}
```

1.3 if...else if...else Statement

- Allows multiple conditions to be checked.

Example:

```
let score = 85;
if (score >= 90) {
    console.log('Grade: A');
} else if (score >= 80) {
    console.log('Grade: B');
} else {
    console.log('Grade: F');
}
```

1.4 Nested if Statement

- An if statement inside another if.

Example:

```
let num = 15;
if (num > 0) {
   if (num % 2 === 0) {
      console.log('The number is positive and even.');
   } else {
      console.log('The number is positive and odd.');
   }
} else {
   console.log('The number is not positive.');
}
```

1.5 switch Statement

- Checks multiple possible values.

Example:

```
let day = 3;
switch (day) {
   case 1:
      console.log('Monday');
      break;
   case 2:
      console.log('Tuesday');
      break;
   case 3:
      console.log('Wednesday');
      break;
   default:
      console.log('Invalid day');
}
```

2. Control Flow Statements

- These statements are used to repeat or skip code blocks.

2.1 for Loop

- Executes a block of code a specified number of times.

Example:

```
for (let i = 1; i <= 5; i++) {
    console.log('Iteration:', i);
}
```

2.2 while Loop

- Executes as long as the condition is true.

Example:

```
let count = 1;
while (count <= 5) {
    console.log('Count:', count);
    count++;
}
```

2.3 do...while Loop

- Executes at least once before checking the condition.

Example:

```
let count = 1;
do {
    console.log('Count:', count);
    count++;
} while (count <= 5);
```

2.4 for...in Loop

- Iterates over properties of an object.

Example:

```
let student = { name: 'John', age: 20, grade: 'A' };
for (let key in student) {
    console.log(key + ':', student[key]);
}
```

2.5 for...of Loop

- Iterates over values of iterable objects.

Example:

```
let numbers = [10, 20, 30, 40];
for (let num of numbers) {
    console.log(num);
}
```

3. Break and Continue

- break: Exits the loop entirely.
- continue: Skips the current iteration.

Break Example:

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

4. Ternary Operator

- Shorthand for if...else.

Syntax:
condition ? expression1 : expression2;

Example:

```
let age = 20;
let status = (age >= 18) ? 'Adult' : 'Minor';
console.log(status); // Output: Adult
```

Statement	Purpose
if	Executes a block of code if a condition is true.
if...else	Executes different blocks of code for true and false conditions.
if...else if	Checks multiple conditions sequentially.
Nested if	Allows if inside another if.
switch	Executes code for a specific value of a variable.
for loop	Repeats a block of code for a specific number of times.
while loop	Repeats a block as long as the condition is true.
do...while	Executes the block at least once and repeats while the condition is true.
for...in	Iterates over the properties of an object.
for...of	Iterates over the values of iterable objects.
break	Exits a loop entirely.
continue	Skips the current iteration and moves to the next.