**Q1. What are conditional statements? Explain conditional statements with syntax and examples.**

Ans:

Conditional statements in programming are used to execute different actions based on different conditions. They allow the program to make decisions and choose different paths of execution based on whether certain conditions are true or false.

Syntax of Conditional Statements:

1. if Statement:

The if statement executes a block of code if a specified condition is true.

if (condition) {

// Code to execute if the condition is true

}

Example:

let num = 10;

if (num > 0) {

console.log("Number is positive");

}

2. if...else Statement:

The if...else statement executes a block of code if the condition is true and another block if the condition is false.

javascript

Copy code

if (condition) {

// Code to execute if the condition is true

} else {

// Code to execute if the condition is false

}

Example:

javascript

Copy code

let num = -5;

if (num > 0) {

console.log("Number is positive");

} else {

console.log("Number is not positive");

}

3. if...else if...else Statement:

The if...else if...else statement allows checking multiple conditions.

if (condition1) {

// Code to execute if condition1 is true

} else if (condition2) {

// Code to execute if condition2 is true

} else {

// Code to execute if all conditions are false

}

Example:

let num = 0;

if (num > 0) {

console.log("Number is positive");

} else if (num < 0) {

console.log("Number is negative");

} else {

console.log("Number is zero");

}

Explanation:

Conditional statements help the program make decisions based on different conditions.

The condition inside the parentheses () of the if statement is evaluated. If it's true, the code inside the block {} following the if is executed.

In an if...else statement, if the condition is true, the code inside the first block executes; otherwise, the code inside the else block executes.

if...else if...else allows checking multiple conditions. The conditions are checked in order, and the first true condition's block of code executes.

Conditional statements are crucial for controlling the flow of a program based on different situations and conditions.

**Q3. What are loops, and what do we need them? Explain different types of loops with their syntax and examples.**

Loops in programming are structures that allow the execution of a set of instructions or code multiple times, enabling automation and repetition. They are essential for efficiently performing repetitive tasks and iterating over collections of data.

### Importance of Loops:

- \*\*Automation:\*\* Loops automate repetitive tasks, reducing manual effort and code duplication.

- \*\*Iterating over Data:\*\* They are used to iterate over arrays, lists, or any collection of items to perform operations on each element.

- \*\*Efficiency:\*\* Loops enhance code efficiency by executing a block of code repeatedly with minimal code.

### Types of Loops in JavaScript:

#### 1. \*\*for Loop:\*\*

The `for` loop repeats a block of code a specified number of times.

```javascript

for (initialization; condition; increment/decrement) {

// Code to be executed

}

```

Example:

```javascript

for (let i = 0; i < 5; i++) {

console.log(i); // Output: 0, 1, 2, 3, 4

}

```

#### 2. \*\*while Loop:\*\*

The `while` loop repeats a block of code as long as the specified condition is true.

```javascript

while (condition) {

// Code to be executed

}

```

Example:

```javascript

let i = 0;

while (i < 5) {

console.log(i); // Output: 0, 1, 2, 3, 4

i++;

}

```

#### 3. \*\*do...while Loop:\*\*

The `do...while` loop is similar to the `while` loop, but it ensures that the code inside the block executes at least once before checking the condition.

```javascript

do {

// Code to be executed

} while (condition);

```

Example:

```javascript

let i = 0;

do {

console.log(i); // Output: 0

i++;

} while (i < 0);

```

### Explanation:

- \*\*`for` loop:\*\* It includes an initialization step, a condition to be checked before each iteration, and an increment or decrement step.

- \*\*`while` loop:\*\* It continues to execute a block of code while the specified condition remains true. It's suitable when the number of iterations is not known beforehand.

- \*\*`do...while` loop:\*\* It executes the block of code once and then checks the condition. If the condition is true, it will continue to execute the code block.

Loops are fundamental in programming for repetitive tasks, iterating through data structures, and enabling efficient execution of code. Choosing the appropriate loop depends on the specific requirements of the task at hand.

**Q4. Generate numbers between any 2 given numbers.**

Ex:

Const num1 = 10

Const num2 = 25;

Output: 11, 12, 13, ...., 25

**JS code**

let num=1 ;

while (num < 25)

{

    num=num+1;

    num++;

    console.log(num);

}