**JavaScript Control Statements**

Control statements in JavaScript are used to control the flow of execution in a program. They allow developers to make decisions, repeat execution, and perform various actions based on conditions. The main types of control statements in JavaScript include:

1. **Conditional Statements:**
   * **if Statement:** The **if** statement is used to execute a block of code if a specified condition is true.

if (condition) { // Code to execute if condition is true }

* + **if-else Statement:** The **if-else** statement allows the execution of one block of code if the condition is true and another block if the condition is false.

if (condition) { // Code to execute if condition is true } else { // Code to execute if condition is false }

* + **if-else if-else Statement:** The **if-else if-else** statement allows the evaluation of 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 }

1. **Looping Statements:**
   * **for Loop:** The **for** loop is used to execute a block of code a specified number of times.

for (initialization; condition; increment/decrement) { // Code to execute }

* + **while Loop:** The **while** loop is used to execute a block of code as long as a specified condition is true.

while (condition) { // Code to execute }

* + **do-while Loop:** The **do-while** loop is similar to the **while** loop but guarantees that the code block is executed at least once before the condition is tested.

do { // Code to execute } while (condition);

1. **Branching Statements:**
   * **break Statement:** The **break** statement is used to terminate a loop or switch statement.
   * **continue Statement:** The **continue** statement is used to skip the current iteration of a loop and continue with the next iteration.
   * **return Statement:** The **return** statement is used to exit a function and return a value to the caller.
2. **Switch Statement:**
   * The **switch** statement is used to execute one of several blocks of code, depending on the value of an expression.

switch (expression) { case value1: // Code to execute if expression matches value1 break; case value2: // Code to execute if expression matches value2 break; default: // Code to execute if expression doesn't match any case }

Control statements are fundamental in JavaScript programming as they enable developers to write dynamic and flexible code that responds to different conditions and requirements.

Sample Codes

1. Conditional Statement

// if Statement

let num = 10;

if (num > 0) {

console.log("Number is positive");

}

// Output: Number is positive

// if-else Statement

let temperature = 25;

if (temperature >= 30) {

console.log("It's hot!");

} else {

console.log("It's not too hot.");

}

// Output: It's not too hot.

// if-else if-else Statement

let score = 85;

if (score >= 90) {

console.log("A grade");

} else if (score >= 80) {

console.log("B grade");

} else {

console.log("C grade or below");

}

// Output: A grade

2.Looping Statements

// for Loop

for (let i = 1; i <= 5; i++) {

console.log(i);

}

// Output: 1 2 3 4 5

// while Loop

let count = 0;

while (count < 3) {

console.log("Count is: " + count);

count++;

}

// Output: Count is: 0

// Count is: 1

// Count is: 2

// do-while Loop

let x = 0;

do {

console.log("x is: " + x);

x++;

} while (x < 3);

// Output: x is: 0

// x is: 1

// x is: 2

// break Statement

for (let i = 1; i <= 5; i++) {

if (i === 3) {

break;

}

console.log(i);

}

// Output: 1 2

// continue Statement

for (let i = 1; i <= 5; i++) {

if (i === 3) {

continue;

}

console.log(i);

}

// Output: 1 2 4 5

// return Statement

function add(a, b) {

return a + b;

}

console.log(add(3, 4));

// Output: 7

1. Switch Statement

// Switch Statement

let day = 3;

switch (day) {

case 1:

console.log("Sunday");

break;

case 2:

console.log("Monday");

break;

case 3:

console.log("Tuesday");

break;

default:

console.log("Other day");

}

// Output: Tuesday