**Lesson 4: Conditional Statements in JavaScript**

By the end of this lesson, students will be able to:

* Demonstrate the Use of if, else if, and else Statements:
  + Students can write JavaScript programs that utilise if, else if, and else statements to execute different code blocks based on specific conditions.
* Implement switch Statements to Handle Multiple Cases:
  + Students will be able to use switch statements to manage scenarios where a variable needs to be compared against multiple values and execute corresponding code blocks.
* Analyse and Debug Conditional Logic in JavaScript:
  + Students can identify and correct logical errors in programs that use conditional statements, ensuring that the code behaves as expected under different conditions.
* Apply Nested Conditional Statements to Solve Complex Problems:
  + Students can construct nested if-else statements to handle more complex decision-making processes in JavaScript programs.

**Conditional Statement**

Conditional statements are a fundamental part of programming that allows you to make decisions in your code. These decisions enable your program to execute certain code blocks based on specified conditions. The primary conditional statements in JavaScript are if, else if, else, and switch. This lesson will guide you through understanding and using these statements effectively.

* The *if* Statement:
  + The if statement executes a block of code only if a specified condition is evaluated as true.

Syntax:

if (condition) {

// Code to be executed if the condition is true

}

Example:

let age = 18;

if (age >= 18) {

console.log("You are eligible to vote.");

}

In this example, the if statement checks if the age value is greater than or equal to 18. If the condition is true, the message "You are eligible to vote." is printed on the console.

* The *else* Statement:
  + The else statement is used in conjunction with if. It specifies a code block to be executed if the condition in the if statement is false.

Syntax:

if (condition) {

// Code to be executed if the condition is true

} else {

// Code to be executed if the condition is false

}

Example:

let age = 16;

if (age >= 18) {

console.log("You are eligible to vote.");

} else {

console.log("You are not eligible to vote.");

}

In this case, since the age is 16, the condition is false, and the else block executes, printing "You are not eligible to vote."

* The *else if* Statement:
  + The else if statement allows you to test multiple conditions. If the first if condition is false, the program checks the next else if condition.

Syntax:

if (condition1) {

// Code to be executed if condition1 is true

} else if (condition2) {

// Code to be executed if condition2 is true

} else {

// Code to be executed if both condition1 and condition2 are false

}

Example:

let score = 85;

if (score >= 90) {

console.log("Grade: A");

} else if (score >= 80) {

console.log("Grade: B");

} else if (score >= 70) {

console.log("Grade: C");

} else {

console.log("Grade: F");

}

The program checks the score against multiple conditions to assign a grade. Since the score is 85, the else if (score >= 80) condition is true, and "Grade: B" is printed.

* The switch Statement:
  + The switch statement is another way to handle multiple conditions. It compares a variable's value against multiple possible values (cases) and executes the corresponding block of code.

Syntax:

switch (expression) {

case value1:

// Code to be executed if expression === value1

break;

case value2:

// Code to be executed if expression === value2

break;

// Add more cases as needed

default:

// Code to be executed if expression does not match any case

}

Example:

let day = 3;

let dayName;

switch (day) {

case 1:

dayName = "Sunday";

break;

case 2:

dayName = "Monday";

break;

case 3:

dayName = "Tuesday";

break;

case 4:

dayName = "Wednesday";

break;

case 5:

dayName = "Thursday";

break;

case 6:

dayName = "Friday";

break;

case 7:

dayName = "Saturday";

break;

default:

dayName = "Invalid day";

}

console.log("Day: " + dayName);

In this example, the switch statement checks the value of the day variable. Since the day is 3, the program sets dayName to "Tuesday" and prints "Day: Tuesday".

* Nested Conditional Statements:
  + You can nest if, else if, and else statements inside one another to create more complex decision structures.

Syntax:

let num = 10;

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.");

}

In this example, the program first checks if num is greater than 0. If true, it then checks if num is even or odd and prints the corresponding message.

* Avoiding Common Mistakes:
  + Misplaced else: Ensure that else follows an if or else if statement directly.
  + No break-in switch: Forgetting to include break statements in switch cases can lead to unintended fall-through behaviour.
  + Using === vs. ==: === checks for value and type equality, while == only checks for value equality.