JavaScript Lab: If Statements

Objective:

By the end of this lab, you will understand how to use `if` statements in JavaScript to control the flow of your program based on conditions.

Introduction:

An `if` statement is a conditional statement that runs a block of code if a specified condition evaluates to `true`. This is essential when you want to execute a certain part of your code only under specific circumstances.

Syntax of If Statement:

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

Example 1:

Check if a number is positive:

```
let number = 5;
if (number > 0) {
    console.log("The number is positive.");
}
```

In this example, the condition `number > 0` checks if the variable `number` is greater than 0. If true, it will print: "The number is positive."

Example 2:

Check if it's raining:

```
let isRaining = true;
if (isRaining) {
```

```
    console.log("Don't forget your umbrella!");
}
```

Here, the `if` statement checks if `isRaining` is true. If it is, the message "Don't forget your umbrella!" will be printed.

Example 3:

Check for eligibility to vote:

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

The condition `age >= 18` checks if a person is 18 years old or older, and if true, prints a message saying they can vote.

Part 1: If-Else Statement

Sometimes, you may want to run one block of code if the condition is true, and another block of code if the condition is false. This is where the `else` clause comes in.

Syntax of If-Else Statement:

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

Example 4:

Check if a number is even or odd:

```
let number = 7;
```

```
if (number % 2 === 0) {
    console.log("The number is even.");
} else {
    console.log("The number is odd.");
}
```

In this example, the program checks if the number is divisible by 2. If true, it prints "The number is even"; otherwise, it prints "The number is odd."

Part 2: Else-If Statement

When you have multiple conditions to check, you can use the `else if` statement. It allows you to check more than one condition.

Syntax of If-Else-If Statement:

```
if (condition1) {
    // Code if condition1 is true
} else if (condition2) {
    // Code if condition2 is true
} else {
    // Code if neither condition1 nor condition2 is true
}
```

Example 5:

Check a student's grade:

```
let score = 75;
if (score >= 80) {
    console.log("You got an A.");
} else if (score >= 60) {
    console.log("You got a B.");
} else {
    console.log("You need to improve.");
}
```

The program checks multiple conditions. If the score is 80 or more, the student gets an A; if it's between 60 and 79, the student gets a B; otherwise, they need to improve.

Exercises:

Exercise 1: Checking Temperature

Write a JavaScript program that checks the temperature and prints:

- "It's too hot!" if the temperature is above 30°C.
- "It's moderate." if the temperature is between 20°C and 30°C.
- "It's cold." if the temperature is below 20°C.

Hint: Use `if`, `else if`, and `else` to check multiple conditions.

```
let temperature = 25;
// Write your if-else statement here
```

Exercise 2: Admission Eligibility

Write a JavaScript program that checks if a person is eligible for admission to a school. The conditions are:

- If the person's score is greater than or equal to 70, print "Eligible for Admission."
- If the score is between 50 and 69, print "Admission on Probation."
- If the score is less than 50, print "Not Eligible for Admission."

```
let score = 55;
// Write your if-else statement here
```

Exercise 3: Access to Clean Water

Write a JavaScript program that checks if a village has enough clean water for its residents. The conditions are:

- If the water available is greater than 500 liters, print "Enough Water."
- If the water is between 200 and 500 liters, print "Water is limited."
- If the water is less than 200 liters, print "Water shortage."

```
let waterAvailable = 300;
```

```
// Write your if-else statement here
```

Part 3: Nested If Statements

You can also use `if` statements inside other `if` statements. This is called **nesting**.

Example 6:

Check if a number is positive, negative, or zero:

```
let number = -3;
if (number >= 0) {
    if (number === 0) {
        console.log("The number is zero.");
    } else {
        console.log("The number is positive.");
    }
} else {
    console.log("The number is negative.");
}
```

Here, if the number is non-negative, the nested `if` checks if it's zero.

Exercise 4: Nested If Example

Write a JavaScript program that checks the quality of a product. The conditions are:

- If the product is "expired", print "Do not use."
- If it's not expired, check the quality:
    - If the quality is greater than 8, print "Good Quality."
    - If the quality is between 5 and 8, print "Average Quality."
    - If the quality is below 5, print "Poor Quality."

```
let expired = false;
let quality = 7;
// Write your nested if statement here
```