Mathematical Operators & Conditional Statements in JavaScript

Instructor: Pramod Kumar Jena

Overview

Today's session will focus on **Mathematical Operators**, **Relational Operators**, and **Conditional Statements** in JavaScript. We'll learn through code examples, real-life scenarios, and a hands-on task that solidifies these concepts.

Common Mathematical Operators in JavaScript

JavaScript supports several mathematical operators. Let's explore them using real-life scenarios.

Scenario 1: Calculating Total Bill in a Restaurant

Imagine a restaurant bill calculation where we add the cost of items and apply discounts based on conditions.

Example:

```
var foodPrice = 200;
var drinkPrice = 50;
var totalBill = foodPrice + drinkPrice; // Addition
var tip = 0.1 * totalBill; // 10% tip on total
var finalAmount = totalBill + tip; // Adding tip to the total bill
console.log("Total Bill: " + finalAmount);
```

Output:

```
Total Bill: 275
```

Real-Life Explanation:

In this example, we calculate the total bill by adding food and drink prices, and then calculate a 10% tip to get the final amount.

Scenario 2: Product Discounts in an E-Commerce Website

Imagine an e-commerce website where a discount is applied based on the total amount of the order.

Example:

```
var originalPrice = 1500;
var discount = 200;
var discountedPrice = originalPrice - discount; // Subtraction
console.log("Discounted Price: " + discountedPrice);
```

Output:

```
Discounted Price: 1300
```

Modulo (Remainder) Operator

The modulo operator (`%`) returns the remainder of a division operation. It is often used to check if a number is even or odd.

Real-Life Scenario: Checking Alternating Payments

Imagine you're building an app to check whether a customer should make a full payment or half payment based on an alternating rule (even for full, odd for half).

Example:

```
var customerID = 101;
if (customerID % 2 === 0) {
  console.log("Customer needs to make a full payment");
} else {
  console.log("Customer needs to make a half payment");
}
```

Output:

```
Customer needs to make a half payment
```

Real-Life Explanation:

If the `customerID` is even, the customer makes a full payment. If it's odd, the customer makes a half payment.

Exponentiation Operator

Scenario: Power Calculations in Energy Usage

Let's say you're developing a system to calculate energy usage, where the power (kW) of a device needs to be calculated.

Example:

```
var base = 10; // Base unit of energy
var hours = 2;
var energyUsage = base ** hours; // Exponentiation
console.log("Energy Usage: " + energyUsage + " kW");
```

Output:

```
Energy Usage: 100 kW
```

Real-Life Explanation:

This calculation can be applied to systems like power management, where you raise a base value to the number of hours to get total energy consumption.

Relational Operators with Real-Life Example

Scenario: Eligibility for Voting

Imagine creating a simple app that checks if a person is eligible to vote based on their age.

Example:

```
var age = 18;
if (age >= 18) {
  console.log("Eligible to vote");
} else {
  console.log("Not eligible to vote");
}
```

Output:

```
Eligible to vote
```

Real-Life Explanation:

In this real-life case, the system checks whether the user's age is 18 or above to determine if they can vote.

Comparison Operators with Real-Life Example

Scenario: Checking Delivery Eligibility

Consider an app that checks if an order qualifies for free delivery based on its total cost.

Example:

```
var orderAmount = 499;
if (orderAmount >= 500) {
  console.log("Eligible for free delivery");
} else {
  console.log("Not eligible for free delivery");
}
```

Output:

```
Not eligible for free delivery
```

Real-Life Explanation:

In this example, the code checks whether the order amount meets the threshold of ₹500 for free delivery.

Conditional Statements with Real-Life Example

Scenario: Grading System in Schools

Imagine creating a grading system that assigns grades based on the marks scored by a student.

Example:

```
var marks = 85;
if (marks >= 90) {
  console.log("Grade: A");
} else if (marks >= 75) {
  console.log("Grade: B");
} else if (marks >= 60) {
  console.log("Grade: C");
} else {
  console.log("Grade: D");
}
```

Output:

```
Grade: B
```

Real-Life Explanation:

The grading system checks the marks and assigns a grade accordingly.

Expanded Task: Building a Simple E-Commerce Bill Calculator

Task Objective:

In this task, the students will build a simple e-commerce bill calculator using JavaScript. They will use **mathematical operators** and **conditional statements** to calculate the total bill, apply discounts, and check if the user is eligible for free delivery.

Task: E-Commerce Bill Calculator

Problem Statement:
Build a simple bill calculator that:

1. Takes the price of three items as input.
2. Calculates the total amount.
3. Applies a discount if the total amount is greater than or equal to ₹1000.
4. Checks if the total qualifies for free delivery (orders of ₹500 or above get free delivery).
5. Adds a delivery charge of ₹50 if the total is below ₹500.
6. Outputs the final amount the user has to pay.

Steps:

1. **Input:**
    - Use `prompt()` to take input from the user for the prices of three items.
2. **Calculate the Total:**
    - Add the prices of the three items.
3. **Apply Discount:**
    - If the total is above ₹1000, apply a discount of 10%.
4. **Check for Free Delivery:**
    - If the total is greater than ₹500, free delivery is applied.
    - If less, add a ₹50 delivery charge.
5. **Output:**
    - Display the final amount the user needs to pay.

Example Solution:

```
// Get prices for three items
var item1 = parseInt(prompt("Enter price of item 1:"));
var item2 = parseInt(prompt("Enter price of item 2:"));
var item3 = parseInt(prompt("Enter price of item 3:"));

// Calculate the total
var totalAmount = item1 + item2 + item3;
```

```
// Apply discount if totalAmount >= 1000
if (totalAmount >= 1000) {
  totalAmount = totalAmount - (0.1 * totalAmount); // 10% discount
  console.log("A 10% discount has been applied");
}

// Check for free delivery
if (totalAmount >= 500) {
  console.log("Eligible for free delivery");
} else {
  totalAmount = totalAmount + 50; // Add delivery charge
  console.log("A delivery charge of ₹50 has been added");
}

// Final amount
console.log("Final Amount: ₹" + totalAmount);
```

Task Explanation:

- **Step 1:** Students will first take the prices of three items using `prompt()`.
- **Step 2:** Calculate the total price by adding the prices of all items.
- **Step 3:** If the total is ₹1000 or above, a 10% discount is applied to the total.
- **Step 4:** Check if the total price qualifies for free delivery. If the total is less than ₹500, a ₹50 delivery charge is added.
- **Step 5:** The final amount is displayed.