JavaScript Notes on Decision Making Statements

Prepared by: Pramod Sir

🔹 What is Decision Making in Programming?

In real life, we make decisions like:

- "If I am hungry, I will eat."
- "If it's raining, I will carry an umbrella."

Similarly, **JavaScript allows us to make decisions using conditional statements** like `if`, `else if`, `else`, and `switch`.

These help us **control the flow** of code depending on certain conditions (true/false).

---

1. `if` Statement

📌 Syntax:

```
if (condition) {
  // block of code to execute if condition is true
}
```

✅ Example:

```
let age = 20;

if (age >= 18) {
  console.log("You are an adult.");
}
```

Explanation: The message prints only if `age` is 18 or more.

---

2. `if...else` Statement

When you want to do **one thing if a condition is true**, and **another if it's false**.

📌 Syntax:

```
if (condition) {
  // runs if condition is true
} else {
  // runs if condition is false
}
```

✅ Example:

```
let isRaining = false;

if (isRaining) {
  console.log("Take an umbrella.");
} else {
  console.log("Enjoy the sunshine!");
}
```

---

3. `if...else if...else` Ladder

Used when you have **multiple conditions to check** one after another.

📌 Syntax:

```
if (condition1) {
  // runs if condition1 is true
} else if (condition2) {
  // runs if condition2 is true
} else {
  // runs if none of the above are true
}
```

✅ Example:

```
let marks = 75;

if (marks >= 90) {
  console.log("Grade: A");
} else if (marks >= 75) {
  console.log("Grade: B");
} else if (marks >= 50) {
  console.log("Grade: C");
} else {
  console.log("Grade: F");
}
```

Explanation: It checks conditions from top to bottom and stops once one is `true`.

---

4. Nested `if` Statements

Used when you want to **check another condition inside a true condition**.

📌 Syntax:

```
if (condition1) {
  if (condition2) {
    // runs if both condition1 and condition2 are true
  }
}
```

✅ Example:

```
let username = "admin";
let password = "1234";

if (username === "admin") {
  if (password === "1234") {
    console.log("Login successful!");
  } else {
    console.log("Wrong password.");
  }
} else {
  console.log("User not found.");
}
```

---

5. `switch` Statement

Used when you want to check a **single variable** against **multiple values**. It's cleaner than writing many `else if`s.

📌 Syntax:

```
switch(expression) {
  case value1:
    // code block
    break;
  case value2:
    // code block
    break;
  default:
    // code block
}
```

✅ Example:

```
let day = 3;
```

```
switch (day) {
  case 1:
    console.log("Monday");
    break;
  case 2:
    console.log("Tuesday");
    break;
  case 3:
    console.log("Wednesday");
    break;
  default:
    console.log("Invalid day");
}
```

Explanation: `switch` matches the value of `day` and runs the matching `case`.

🔹 Block vs Single Line in `if`

If you have **one line**, curly braces `{}` are optional:

```
let age = 18;
if (age >= 18) console.log("You can vote");
```

But for **multiple lines**, always use `{}`:

```
if (age >= 18) {
  console.log("You can vote");
  console.log("Remember to carry your ID!");
}
```

Summary Table

Statemet	Use When...
`if`	You need to check **one condition**
`if...else`	You want **two possible outcomes**
`else if`	You want **many options**
`nested if`	You want **condition inside condition**
`switch`	You check **one variable, many values**

Practice for Students

Try predicting the output before running:

```
let num = 5;

if (num % 2 === 0) {
  console.log("Even");
} else if (num % 3 === 0) {
  console.log("Divisible by 3");
} else {
  console.log("Odd");
}
```

Why Use `else if` and `else`?

✅ Problem with only `if`:

If you use only `if` statements one after the other, **each one is checked**, **even after the correct one is already true**.

📌 **Example:**

```
let marks = 85;

if (marks >= 90) {
  console.log("Grade A");
}
if (marks >= 75) {
  console.log("Grade B");
}
if (marks >= 50) {
  console.log("Grade C");
}
```

✅ **Output:**

```
Grade B
Grade C
```

Why? → All conditions after the first match are still checked.

✅ Solution: Use `else if` and `else`

```
let marks = 85;

if (marks >= 90) {
  console.log("Grade A");
} else if (marks >= 75) {
  console.log("Grade B");
} else if (marks >= 50) {
  console.log("Grade C");
} else {
  console.log("Fail");
}
```

✅ **Output:**

```
Grade B
```

👉 **Efficient & Correct!** It stops checking once a match is found.

Practice Set: With Dialog Boxes

🟩 1. Age Group Checker

```
let age = prompt("Enter your age:");

if (age >= 0 && age <= 12) {
  alert("You are a child.");
} else if (age >= 13 && age <= 19) {
  alert("You are a teenager.");
} else if (age >= 20 && age <= 59) {
  alert("You are an adult.");
} else {
  alert("You are a senior citizen.");
}
```

🟨 2. Simple Calculator (using switch)

```
let a = Number(prompt("Enter first number:"));
let b = Number(prompt("Enter second number:"));
let op = prompt("Enter operation (+, -, *, /):");

switch (op) {
  case "+":
    alert("Result = " + (a + b));
    break;
  case "-":
    alert("Result = " + (a - b));
    break;
  case "*":
    alert("Result = " + (a * b));
    break;
  case "/":
    alert("Result = " + (a / b));
    break;
  default:
    alert("Invalid operator!");
}
```

🟦 3. Login Check

```
let user = prompt("Enter username:");
let pass = prompt("Enter password:");

if (user === "admin" && pass === "1234") {
  alert("Welcome, Admin!");
} else if (user === "admin") {
  alert("Incorrect password.");
} else {
  alert("User not found.");
}
```

🟧 4. Traffic Light Simulation

```
let signal = prompt("Enter traffic light color (red, yellow,
green):");

switch (signal.toLowerCase()) {
  case "red":
    alert("Stop!");
    break;
  case "yellow":
    alert("Get Ready!");
    break;
  case "green":
    alert("Go!");
    break;
  default:
    alert("Invalid color!");
}
```

🟪 5. Even or Odd Checker

```
let num = Number(prompt("Enter a number:"));

if (num % 2 === 0) {
  alert("Even number");
} else {
  alert("Odd number");
}
```

Conclusion

- ✅ Use `if...else if...else` for **clear, readable, and efficient** condition checking.
- ❌ Avoid chaining only `if` — it causes **unnecessary execution**.
- `switch` is great when checking **one variable against many values**.
- Dialog boxes help **interact with users** in a simple way for beginners.