■ JavaScript Activity: Guess the Number – Learning if Statements

Objective: Learn how to use if, else if, and else statements to control the flow of a JavaScript program.

Step 0: Create Your HTML File

Before running your JavaScript, create a simple HTML page that connects to your script. Save this file as index.html in the same folder as your JavaScript file.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Guess the Number</title>
</head>
<body>
  <h1>Guess the Number Game</h1>
  <p>Open the console or follow the on-screen prompts to play!</p>

  <script src="script.js"></script>
</body>
</html>
```

■ Save your JavaScript code in a separate file called script.js. When you open index.html in your browser, your prompts and alerts will appear.

Step 1: Run the Code

Copy and paste this code into your **script.js** file or an online editor (like JSFiddle, CodePen, or VS Code Live Server):

```
// Generate a random number between 1 and 10
let randomNumber = Math.floor(Math.random() * 10) + 1;

// Prompt the user for their guess
let guess = prompt("Guess a number between 1 and 10:");

// Convert the guess to a number
guess = parseInt(guess);

// Check if the guess is correct
if (guess === randomNumber) {
  alert("You guessed it!");
} else {
  alert(`Wrong guess. Try again.
```

```
    The number was ${randomNumber}`);
}
```

Run the program several times and observe what happens when your guess is correct or incorrect.

Step 2: Understand How if Works

Look at this part of the code and note how it checks for a condition:

```
if (guess === randomNumber) {
  alert("You guessed it!");
}
```

The **if** checks if the condition is true. If it is, the code inside the braces runs. Otherwise, JavaScript skips to the next condition.

Step 3: Add More Conditions

Add these lines between the if and the else to give more feedback to the user:

```
else if (guess > randomNumber) {
  alert("Too high! Try a smaller number.");
} else if (guess < randomNumber) {
  alert("Too low! Try a bigger number.");
}
```

Test your program again — it now tells you if your guess was too high or too low.

Step 4: Reflection

At the bottom of your code, add a comment answering these questions:

```
// 1. What does the if statement check for?
// 2. When does the else if part run?
// 3. Why do we use parseInt() before comparing the numbers?
```

■ Bonus Challenge (No Loops!)

Let's give the user a second chance to guess the number — but without using loops. After the first else block, add this code:

```
// Give the user a second chance
if (guess !== randomNumber) {
  let secondGuess = prompt("Try again! Guess one more time:");
  secondGuess = parseInt(secondGuess);
  if (secondGuess === randomNumber) {
```

```
    alert("Nice! You got it on the second try!");
  } else {
    alert(`Sorry, still wrong. The number was ${randomNumber}.`);
  }
}
```

Run your code again and check that:

- The user only gets a second prompt if the first guess was wrong.
- The correct message appears for both tries.