Training Day 16 Report

Date: 20 June 2024
Location: Science & Technology Entrepreneurs' Park
Project Title: *JavaScript Conditional Statements and Loops*

On Day 16, we continued exploring JavaScript and focused on two important topics: **conditional statements** and **loops**. These concepts are the foundation of making web pages dynamic and responsive to different situations. In simple terms, they help us write code that can **make decisions** and **repeat tasks**, just like we humans do in our daily lives.

I was excited to learn how to make a webpage “think” by using conditions. It was also fun to make things repeat using loops. These are powerful tools in programming, and understanding them gave me a lot more confidence in writing logical code.

Project Objectives

- Learn how JavaScript makes decisions using conditional statements.
- Understand `if`, `else if`, `else`, and `switch` statements.
- Learn how to compare values using comparison and logical operators.
- Understand loops and how they repeat tasks.
- Practice with `for`, `while`, and `do...while` loops.
- Understand basic flow of logic in programs.

Conditional Statements in JavaScript

Conditional statements allow us to execute different blocks of code depending on whether a condition is **true** or **false**. This is helpful when we want our program to behave differently under certain circumstances.

`if` Statement

This checks a condition and runs a block of code **only if** the condition is true.
Example: If a user enters their age, the website can check if they are above 18 and show them a different message.

`else` Statement

If the condition is not true, the `else` block runs instead.
This is like saying: *If something happens, do this. Otherwise, do something else.*

`else if`

Sometimes we want to check multiple conditions.
Example: If a score is above 90, we show “Excellent”. If it’s above 70, we show “Good”. Otherwise, we show “Needs Improvement”.

This kind of step-by-step decision making felt very similar to how we think in real life, so it made sense quickly.

Comparison and Logical Operators

To use conditions, we also need **comparison operators**. These compare two values and return `true` or `false`. Some examples we learned:

- `==` : checks if two values are equal
- `===` : checks both value and type
- `!=` : not equal
- `<`, `>`, `<=`, `>=` : number comparisons

We also practiced **logical operators**:

- `&&` : “and” – both conditions must be true
- `||` : “or” – at least one condition must be true
- `!` : “not” – reverses the condition

These helped us build more advanced conditions.
Example: Check if someone is between 18 and 25 years old using `&&`.

The `switch` Statement

We also learned about the `switch` statement, which is a cleaner way to check multiple exact matches.
It works well when checking values like days of the week or menu options.

Example: If someone selects “Option A” or “Option B”, we can use `case` labels inside a `switch`. This made the code shorter and easier to read than using many `else if` statements.

Loops in JavaScript

After learning how to make decisions, we moved on to **loops**, which let us repeat code multiple times.
Loops are super helpful for performing the same task several times, like printing numbers from 1 to 10.

`for` Loop

Runs a block of code a specific number of times.
It has three parts: a starting point, a condition to check, and a step to move to the next round.

Example: Use a `for` loop to print all even numbers from 2 to 10.

`while` Loop

Checks a condition before each round and keeps going as long as the condition is true.
It’s more flexible than a `for` loop and can run an unknown number of times.

 Important: Always **update the condition** inside the loop to avoid **infinite loops**.

`do...while` Loop

Similar to the `while` loop, but **always runs at least once**. The condition is checked **after** the first run.

Useful when we want to ask a user to enter a value at least once, and then check if it’s valid.

Real-World Examples We Discussed

To make learning more practical, our instructor gave real-life use cases:

- Show a welcome message only if a user is logged in.
- Calculate the total price of multiple items in a shopping cart.
- Repeat a quiz question until the correct answer is given.
- Show different content depending on the time of day.

These examples helped me connect the logic to actual websites I use every day.

Summary

Day 16 was a big leap forward in my JavaScript learning.
Conditional statements helped me understand how websites make decisions based on user input.
Loops showed me how we can automate repeated tasks.

The combination of both gave me the power to create more logical, efficient, and interactive web pages.
I practiced with small examples like checking marks and printing numbers. It was really satisfying to see how a few lines of code could control so much.

Although I’m still learning to think like a programmer, today gave me a solid foundation to build on.
Tomorrow, I’m excited to learn about **functions**, which are the next building block in writing organized and reusable code.

BY: *Ekamjot Kaur*
URN: 2302867
CRN: 2315264