JS Fundamentals

By Naimul Islam Mehedi

Variables, Data Types, Control Flow (loops, if/else)

Variables in JavaScript

A **variable** is like a container that stores data.

In JavaScript, we usually declare variables using **`let`**, **`const`**, or **`var`** (though `var` is old and less recommended).

- `let` → block-scoped, can be reassigned.
- `const` → block-scoped, cannot be reassigned.
- `var` → function-scoped, can cause unexpected issues (not recommended).

```
let age = 25; // variable that can be changed
const pi = 3.1416; // canstant value (cannot be
changed)
var name = "Mehedi"; // old of declaring (avoid in
modern JS)

age = 30;
pi = 3.15

console.log(age);
console.log(pi);
console.log(name)
```

Data Types in JavaScript (Primitive)

In JavaScript, there are **two main categories of data types**:

1. Primitive Types (basic building blocks)

- `String` → text (e.g., `"Hello"`, `'World'`)
- `Number` → numbers (e.g., `42`, `3.14`)
- `Boolean` → true/false
- `Undefined` → declared but not assigned
- `Null` → intentional empty value
- `Symbol` → unique value (advanced, less used in basics)
- `BigInt` → very large numbers

2. Non-Primitive Types

- `Object` → collection of key-value pairs
- `Array` → list of values
- `Function` → block of reusable code

```
let name = "Mehedi" // String
let age = 25; // Number
let isStudent = true; // Boolean
let address; // undefined (not assigned yet)
let car = null // Null
let numbers = [1, 2, 3]; // Array
let person = {name: "Ali", age: 30}; // Object
```

Control Flow (if / else)

- **Control flow** means deciding which code should run depending on conditions.
- The most basic way is using **if / else statements**.

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

```
let age = 18;

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

Control Flow (if / else)

- **Control flow** means deciding which code should run depending on conditions.
- The most basic way is using **if / else statements**.

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

```
let age = 18;

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

```
// Example 2 (with else if):

let marks = 75;

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

https://github.com/naimulislammehedi/Front-End-Web-Development

Loops in JavaScript

A **loop** lets us repeat a block of code multiple times, instead of writing it again and again.

1. for loop: Runs a block of code a fixed number of times.

2. while loop: Runs as long as the condition is true.

3. do...while loop: Runs the code at least once, **then** checks the condition.

```
// for Loop
for (let i = 1; i <= 5; i++) {
    console.log("Number: " + 2);
}

// while Loop
let count = 1;
while (count <= 3) {
    console.log("Count is: " + count);
    count++;
}

// do while Loop
let num = 1;
do {
    console.log("Num is: " + num);
    num++;
} while (num <= 3);
```

https://github.com/naimulislammehedi/Front-End-Web-Development