Functions & Data Structures

By Naimul Islam Mehedi

Functions → Arrays → Objects

Functions

A function is like a *machine* in programming. You give it an **input** (parameters), it does some work (logic), and then it gives you an **output** (return value).

Functions help us **reuse code** instead of writing the same thing again and again.

```
function addNumbers(a, b) {
    return a + b;
}

let results = addNumbers(5, 3);
console.log(results);
```

👩‍🏫 **Question for you:**

Write a function called `multiply` that takes two numbers and returns their multiplication.

(You don’t need to run it, just write the code.)

Arrays

An **array** is like a *list* or *box with compartments* where we can store multiple values under one variable.

```
let numbers = [10, 20, 30, 40];
console.log(numbers[0]);
console.log(numbers[2]);

// Changing a value
numbers[1] = 25;
console.log(numbers);

// Adding a new value
numbers.push(50);
console.log(numbers);
```

👩‍🏫 **Question for you:**
Create an array called `colors` with at least 3 colors.
Then:

1. Print the **first color**.
2. Add a new color to the array.
3. Print the whole array.

Objects

An **object** is like a dictionary. It stores data in **key–value pairs**.

This is useful when we want to describe something with multiple properties.

👩‍💻 **Question for you:**
Create an object called `car` with these properties:

- `brand` (example: "Toyota")
- `model` (example: "Corolla")
- `year` (example: 2020)

Then:

1. Print the brand.
2. Update the year to `2023`.
3. Add a new property called `color`.
4. Print the whole object.

```
let person = {
    name: "Mehedi",
    age: 25,
    country: "Bangladesh"
};

console.log(person.name);
console.log(person["age"]);

person.age = 26;

person.job = "Engineer";

console.log(person)
```