**Intern Name:** Fatima Wajid

**Submitted to:** Ali Hyder

**Date:** 22nd July 2025

**Internship Role: Web Development**

## Objective of the Day

To understand how **JavaScript objects** work, how to create and manipulate them, and how to define and use **object methods** for encapsulating logic and functionality within objects.

## What Are JavaScript Objects?

Objects in JavaScript are **collections of key-value pairs**. They allow us to store multiple related values and functions (methods) together in a single unit. Objects are essential for organizing data and building real-world applications.

## Object Syntax:

```
const person = {
  name: "Alice",
  age: 25,
```

```
  isStudent: true
};
```

**Creating JavaScript Objects**

There are multiple ways to create objects:

**1. Using Object Literals (most common):**

```
const car = {
  brand: "Toyota",
  model: "Corolla",
  year: 2020
};
```

**2. Using the new Object() Constructor:**

```
const user = new Object();
user.name = "John";
user.email = "john@example.com";
```

**3. Using Constructor Functions:**

```
function Animal(type, sound) {
  this.type = type;
  this.sound = sound;
}
const dog = new Animal("Dog", "Bark");
```

**4. Using Classes (ES6):**

```
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
```

```
  }
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
}
const p1 = new Person("Emma", 30);
```

**Accessing and Modifying Object Properties**

**Access using dot notation:**

```
console.log(car.brand); // Toyota
```

**Access using bracket notation:**

```
console.log(car["model"]); // Corolla
```

**Modify property:**

```
car.year = 2021;
```

**Add new property:**

```
car.color = "Blue";
```

**Delete property:**

```
delete car.model;
```

**Object Methods**

Object methods are functions stored as object properties. They are used to define behaviors for objects.

**Example:**

```
const person = {
  name: "Ali",
  age: 22,
  greet: function () {
    return "Hello, I am " + this.name;
  }
};

console.log(person.greet()); // Hello, I am Ali
```

**Using ES6 Method Shorthand:**

```
const person = {
  name: "Ali",
  greet() {
    return `Hi! I'm ${this.name}`;
  }
};
```

**Built-In Object Methods**

Some common built-in methods available on objects include:

- Object.keys(obj) → returns an array of keys
- Object.values(obj) → returns an array of values
- Object.entries(obj) → returns an array of [key, value] pairs
- Object.assign(target, source) → copies properties to another object
- hasOwnProperty() → checks if the property exists in object

**Example:**

```
const student = {
  name: "Sara",
```

```
  grade: "A"
};

console.log(Object.keys(student)); // ["name", "grade"]
console.log(Object.values(student)); // ["Sara", "A"]
```

**Nested Objects**

Objects can contain other objects or arrays as properties.

**Example:**

```
const school = {
  name: "Green Valley School",
  address: {
    street: "Main Rd",
    city: "Lahore"
  },
  students: ["Ali", "Sara", "Ahmed"]
};
```

**What I Learned Today**

- How to create objects using different methods (literals, constructors, classes)
- How to access, modify, and delete object properties
- How to define and invoke object methods
- Usage of built-in object methods like Object.keys(), Object.values(), etc.
- How to manage and access nested objects

**Conclusion**

Understanding **objects and methods** is a key skill in JavaScript. They are the building blocks for structuring data and logic, especially in large applications. Today's session strengthened my foundation and prepared me for working with more complex data structures and OOP concepts in JavaScript.