# Report Day: 4.4

**Submitted By: Ayesha Abid**

**Submitted To: Mr. Ali Hyder Hidayat**

**Group: Frontend Development @ProSensia**

**Dated: 17/07/2025**

## Learning Outcomes:

## 1. Introduction

In JavaScript, loops are used to execute a block of code repeatedly based on a condition, and arrays are used to store multiple values in a single variable. Together, these concepts are essential for efficiently handling repetitive tasks and managing collections of data in web applications.

### 2. Loops in JavaScript

Loops allow a program to **execute a block of code multiple times** without writing it repeatedly.

#### 2.1 for Loop

The for loop is commonly used when the number of iterations is known in advance.

```
for (initialization; condition; update) {
  // code to execute
}
```

#### 2.2 while Loop

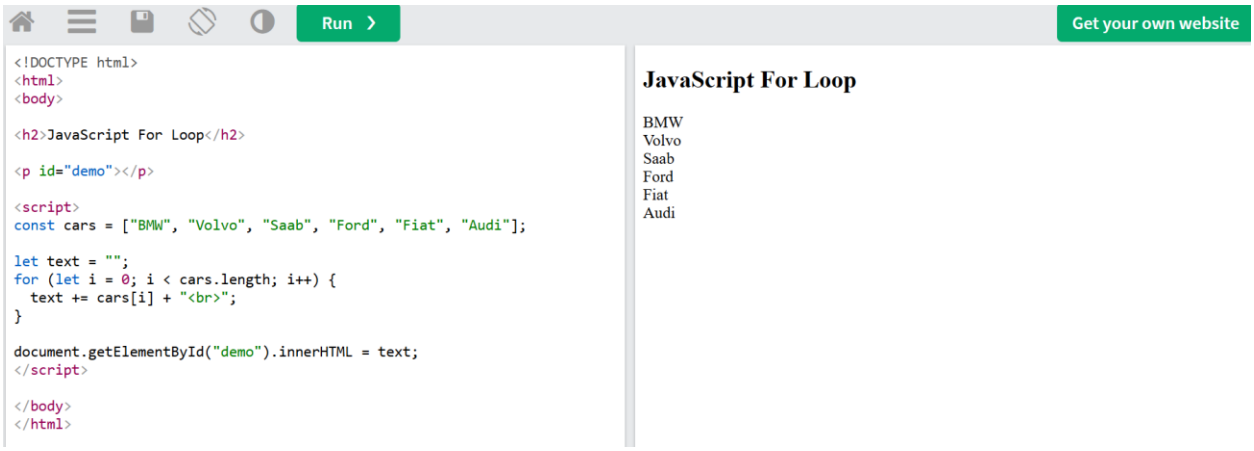
The while loop is used when the number of iterations is **not known beforehand**. It checks the condition before running the loop body

```
let i = 1;
while (i <= 3) {
  console.log("i is: " + i);
  i++;
}
```

### 2.3 do...while Loop

The do...while loop **executes the block at least once**, then continues if the condition is true.

```
let i = 1;
do {
  console.log("Counting: " + i);
  i++;
} while (i <= 2);
```




## 3. Arrays in JavaScript

An **array** is a data structure that stores **multiple values** in a single variable. Each value is called an **element**, and each has an **index** starting from 0.

**Accessing Array Elements:**

```
let fruits = ["Apple", "Banana", "Mango"];
console.log(fruits[0]); // Apple
console.log(fruits[2]); // Mango
```

**Modifying Array Elements:**

```
fruits[1] = "Orange";
console.log(fruits); // ["Apple", "Orange", "Mango"]
```

**Array Length**

Returns the total number of elements in the array.

```
console.log(fruits.length); // 3
```

**Looping Through an Array:**

```
for (let i = 0; i < fruits.length; i++) {
  console.log(fruits[i]);
}
```

### 3.6 Common Array Methods

| Method | Description | Example |
|---|---|---|
| push() | Adds element at end | fruits.push("Grapes") |
| pop() | Removes last element | fruits.pop() |
| shift() | Removes first element | fruits.shift() |
| unshift() | Adds element at start | fruits.unshift("Pineapple") |
| indexOf() | Returns index of element | fruits.indexOf("Mango") |
| includes() | Checks if element exists | fruits.includes("Apple") → true |
| join() | Joins array into string | fruits.join(", ") |

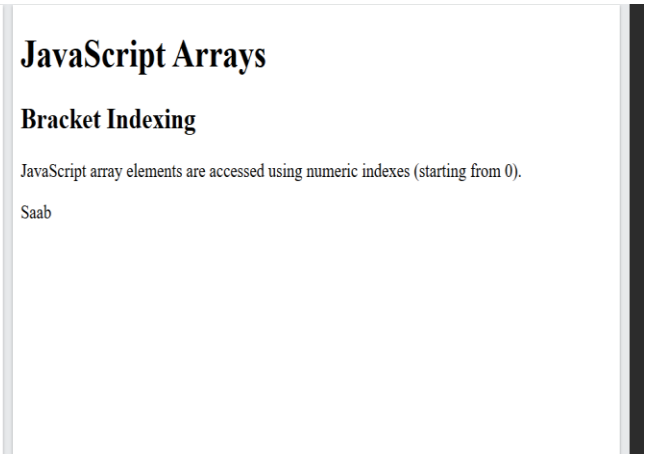
```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accessed using numeric indexes
(starting from 0).</p>

<p id="demo"></p>

<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>

</body>
</html>
```




## 5. Conclusion

Loops and arrays are core features in JavaScript for handling repetitive tasks and storing data collections. Mastery of for, while, and do...while loops alongside array operations enables developers to write clean, efficient, and scalable code. Together, they form the foundation for building dynamic applications and logic-driven program.