Lecture-06: Arrays in JavaScript

- **Teacher:** Sir Arsalan Shah
- **Typist:** Yousuf Naveed

Introduction to Arrays:

In JavaScript, an array is a special type of variable that can store multiple values in a single variable. Each value in an array is called an element, and each element has an index, starting from 0.

Creating and Accessing Array Elements:

```
let names = ["khuzaima", "huzaifa", "sana", "misbah", "shoaib",
"yousuf", "afzal", "muneeb", "kashaf", "kinza", "abdul rafay"];
```

In this example, we created an array called "names" that contains several names as elements.

Accessing Elements:

```
        console.log(names[4]); // Output: "shoaib"
        document.write(names[4]); // Output: "shoaib"
```

- We can access individual elements in an array using their index. In this case, index 4 corresponds to the element "shoaib" in the array.

Array Properties and Methods:

.length:

```
        console.log(names.length); // Output: 11
```

- The ".length" property returns the number of elements in the array. In this case, the array "names" has 11 elements.

.push:

```
        names.push("zainab");
        console.log(names); // Output: ["khuzaima", "huzaifa", ...,
"abdul rafay", "zainab"]
```

- The ".push" method adds a new element to the end of the array. In this example, we added the name "zainab" to the "names" array.

.unshift:

```
        names.unshift("ahmed");
        console.log(names); // Output: ["ahmed", "khuzaima", ...,
"abdul rafay", "zainab"]
```

- The ".unshift" method adds a new element to the beginning of the array. In this example, we added the name "ahmed" to the start of the "names" array.

.shift:

```
        names.shift();
        console.log(names); // Output: ["khuzaima", ..., "abdul rafay",
"zainab"]
```

- The ".shift" method removes the first element from the array. In this example, we removed the element "ahmed" from the "names" array.

.pop:

```
        names.pop();
        console.log(names); // Output: ["khuzaima", ..., "abdul rafay"]
```

- The ".pop" method removes the last element from the array. In this example, we removed the element "zainab" from the "names" array.

.sort:

```
        names.sort();
        console.log(names); // Output: ["abdul rafay", "afzal", ...,
"yousuf"]
```

- The ".sort" method arranges the elements of the array in alphabetical order. In this example, the "names" array is sorted alphabetically.

Modifying Array Elements:

```
        names[3] = "khuzaima";
        console.log(names); // Output: ["abdul rafay", "afzal",
"khuzaima", ..., "yousuf"]
```

- We can modify array elements by assigning new values to specific indexes. In this case, we changed the element at index 3 to "khuzaima."

Finding Index of an Element:

```
        console.log(names.indexOf("huzaifa")); // Output: 1
```

- The "indexOf" method returns the index of the first occurrence of the specified element in the array. In this example, "huzaifa" is found at index 1.

Concatenating Arrays:

```
        let additionalNames = ["khuzaima", “huzaifa"];
        let allNames = names.concat(additionalNames);
        console.log(allNames); // Output: ["abdul rafay", "afzal", ...,
"yousuf", "khuzaima", "huzaifa"]
```

- The "concat" method combines two or more arrays into a new array. In this example, we concatenated the "names" array with the "additionalNames" array to create "allNames."

Arrays are versatile data structures that allow us to store and manipulate multiple values efficiently. They play a crucial role in JavaScript programming.