

### 3. JavaScript Array Handling and Array Methods:

JavaScript is a versatile programming language that provides built-in support for arrays. Arrays are a fundamental data structure in JavaScript, allowing you to store and manipulate collections of values or objects. In this explanation, I will cover array handling in JavaScript and several commonly used array methods with suitable examples.

#### Array Basics:

```
let fruits = ['apple', 'banana', 'cherry'];
```

#### Accessing Elements:

```
console.log(fruits[0]); // Outputs 'apple'
```

#### Array Methods:

JavaScript provides a variety of built-in methods for working with arrays such as:

##### a. push and pop

These methods are used to add and remove elements from the end of an array.

```
fruits.push('date'); // Adds 'date' to the end  
fruits.pop(); // Removes the last element ('date')
```

##### b. unshift and shift

These methods add and remove elements from the beginning of an array.

```
fruits.unshift('grape'); // Adds 'grape' to the beginning  
fruits.shift(); // Removes the first element ('grape')
```

##### c. splice:

splice is used to add, remove, or replace elements at a specified position within an array.

```
fruits.splice(1, 1, 'kiwi'); // Replaces the element at index 1 with 'kiwi'
```

##### d. slice:

slice creates a new array by extracting a portion of the original array.

```
let citrus = fruits.slice(1, 3); // Returns a new array ['banana', 'cherry']
```

##### e. concat:

concat combines two or more arrays to create a new array.

```
let moreFruits = ['grapefruit', 'blueberry'];  
let allFruits = fruits.concat(moreFruits);
```

##### f. indexOf and lastIndexOf:

These methods find the index of the first/last occurrence of an element in an array.

```
let index = fruits.indexOf('banana'); // Returns 1
```

### **g. filter:**

filter creates a new array containing elements that meet a specific condition.

```
let longFruits = fruits.filter(fruit => fruit.length > 5); // Returns ['banana', 'cherry']
```

### **h. map:**

map creates a new array by applying a function to each element of the original array.

```
let upperCaseFruits = fruits.map(fruit => fruit.toUpperCase()); // ['APPLE', 'BANANA', 'CHERRY']
```

## **3. illustration of the features and necessity of web servers and web clients when designing a web page,**

- **Web Server:**

- Hosts the web site files (HTML, CSS, JavaScript, images etc.)
- Processes requests and serves responses using HTTP protocol.
- Key features: speed, reliability, scalability, security
- Example: Apache, Nginx, IIS

The web server is necessary to host the files that make up a website and deliver them to clients. Without a web server, the website would not be accessible to clients.

- **Web Client:**

- Browser used to access the web site (Chrome, Firefox, Safari etc.)
- Sends HTTP requests and renders responses
- Key features: rendering engine, developer tools, extensions
- Runs on user's device (desktop, mobile etc.)

The web client is necessary for users to access the website. It sends requests to the web server and renders the responses into the interactive web page. Without a client, the user has no way to view or interact with the website.

- **Example website:**

- Web server hosts HTML, CSS, images, other files
- Client sends request for index.html
- Server returns index.html + linked CSS and images
- Browser renders page with styling and images