

# LAB ASSIGNMENT – 4 ANSWERS

## 1. Explain Array methods in JavaScript. Specifically, demonstrate how `push()`, `pop()`, `shift()`, and `unshift()` modify an array.

JavaScript arrays provide various methods to insert, remove, and modify elements.

### `push()`

- Adds one or more elements at the *end* of the array.
- Returns the new length of the array.

### `pop()`

- Removes the *last* element from the array.
- Returns the removed element.

### `shift()`

- Removes the *first* element from the array.
- Shifts remaining elements to a lower index.

### `unshift()`

- Adds one or more elements at the *beginning* of the array.
- Returns the new length of the array.

These methods modify the array directly (they are *mutating* methods).

## 2. What are Promises in JavaScript, and how do `async/await` simplify working with asynchronous code?

A **Promise** is an object that represents the eventual completion or failure of an asynchronous operation.

It has three states:

- *pending*
- *fulfilled*
- *rejected*

Promises help avoid deeply nested callbacks (callback hell).

### `async/await`:

- **async** is used to declare an asynchronous function that implicitly returns a Promise.
- **await** pauses execution inside an `async` function until the Promise is resolved or rejected.

They make asynchronous code look like synchronous code, improving readability and error handling using `try...catch`.

## 3. Describe the concept of Event Delegation and explain the use of `addEventListener`.

**Event Delegation** is a technique in which a single event listener is added to a parent element instead of individual child elements.

Because events bubble up through the DOM tree, the parent can detect events that happen on its children. This improves performance and reduces memory usage.

**addEventListener** is used to attach an event to an element without overwriting existing events.

Syntax example conceptually:

```
element.addEventListener('eventName', callbackFunction)
```

It supports multiple events of the same type and allows options like capture and passive mode. Event delegation commonly uses addEventListener on a parent to manage events of dynamically created elements.