



Intern Name: Fatima Wajid

Submitted to: Ali Hyder

Internship Role: Web Development

1. Introduction:

Today's topic was **Asynchronous JavaScript**, which allows code to run **without blocking** other operations. This is important for tasks like API calls, timers, and animations.

2. setTimeout

Runs code **once** after a given delay (in milliseconds).

```
setTimeout(() => {  
  console.log("Hello after 2 seconds");  
}, 2000);
```

3. setInterval

Runs code **repeatedly** after a fixed time interval.

```
setInterval(() => {  
  console.log("Repeating every 1 second");  
}, 1000);
```

4. Promises

Promises handle asynchronous operations and represent a value that will be available in the future.

Example:

```
let promise = new Promise((resolve, reject) => {  
  let success = true;  
  if (success) {  
    resolve("Task completed!");  
  } else {  
    reject("Error occurred!");  
  }  
});
```

```
promise.then(result => console.log(result))  
  .catch(error => console.log(error));
```

5. Practical Example:

```
function fetchData() {  
  return new Promise((resolve) => {  
    setTimeout(() => {  
      resolve("Data received from server");  
    }, 3000);  
  });  
}
```

```
fetchData().then(data => console.log(data));
```

6. Conclusion:

Asynchronous JavaScript is crucial for creating smooth, responsive websites. **setTimeout** and **setInterval** control timing, while **Promises** handle tasks that take time, like data fetching.