Lesson: Summary

Module Name: Async JS, Browser API & Window

Introduction to Async JS

- Async JS (Asynchronous JavaScript) enables concurrent task execution without blocking the main thread, resulting in more efficient and responsive web applications.
- Asynchronous operations start and run concurrently with other code execution.

Introduction to AJAX

- AJAX (Asynchronous JavaScript and XML) allows web pages to update parts asynchronously without reloading the entire page, leading to faster and more responsive interfaces.
- The XMLHttpRequest object is fundamental for making HTTP requests and retrieving data asynchronously.

XMLHttpRequest Object

- Methods include abort(), getAllResponseHeaders(), getResponseHeader(), send(), and setRequestHeader().
- Properties include onreadystatechange, readyState, responseText, responseXML, status, and statusText.

Promises

- Promises represent values that will be available in the future, allowing asynchronous code execution without blocking.
- Lifecycle states: Pending, Resolved, Rejected, Settled.
- Methods for consuming promises: .then(), .catch(), .finally().

Promise Methods

- Promise.all(): Waits for all promises to resolve.
- Promise.any(): Resolves with the first fulfilled promise.
- Promise.resolve(): Creates a resolved promise.
- Promise.reject(): Creates a rejected promise.

Async/Await

- async keyword marks a function as asynchronous, returning a promise.
- await keyword pauses function execution until the promise resolves or rejects.
- Benefits include improved readability, better error handling, and avoiding "callback hell."

JSON Parse and Stringify

- JSON.parse(): Converts JSON string to JavaScript object.
- JSON.stringify(): Converts JavaScript object to JSON string.
- Best practices include error handling, validation, performance considerations, and avoiding circular references
    -