**1. What is Fetch API?**

The Fetch API provides a JavaScript interface for making network requests, similar to

XMLHttpRequest but with a more powerful and flexible feature set. It returns a

Promise that resolves to the Response to that request, allowing for cleaner and more manageable asynchronous code.

**2. How do promises work?**

A Promise is a JavaScript object that represents the eventual completion or failure of an asynchronous operation. It has three states:

* **Pending**: The initial state, neither fulfilled nor rejected.
* **Fulfilled**: The operation completed successfully.
* **Rejected**: The operation failed.

You can attach handlers to a promise using

.then() for a successful result and .catch() for a rejected result.

**3. Difference between synchronous and asynchronous code?**

* **Synchronous code** is executed in sequence, one line at a time. Each operation must complete before the next one can begin. This can cause the program to "block" or become unresponsive during long-running tasks.
* **Asynchronous code** allows a program to start a long-running task and continue executing other tasks without waiting for the first one to finish. The result of the asynchronous operation is handled later, typically with a callback function or a promise.

**4. How to handle errors in Fetch?**

Errors in a Fetch request can be handled using a

.catch() block. The

.catch() block is executed if the promise is rejected, which typically happens due to network errors. It's also important to check the

response.ok property within the .then() block, as a successful fetch response (e.g., a 404 Not Found status) will not automatically trigger the .catch() block.

**5. What is JSON?**

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is human-readable and easy for machines to parse and generate. It is commonly used for transmitting data between a server and a web application.

**6. What is CORS?**

CORS (Cross-Origin Resource Sharing) is a browser security feature that restricts web pages from making requests to a different domain than the one that served the web page. It is a mechanism that allows a server to specify who can access its resources.

**7. How to parse JSON?**

You can parse a JSON response from a Fetch request by calling the

.json() method on the Response object. This method returns a promise that resolves with the JavaScript object parsed from the JSON data.

**8. Explain async/await.**

async and await are modern JavaScript keywords that provide a more concise and readable way to work with promises.

* The async keyword is used to declare an asynchronous function, which will always return a promise.
* The await keyword can only be used inside an async function. It pauses the execution of the function until a promise is settled (either resolved or rejected) and returns the resolved value.

**9. What are HTTP status codes?**

HTTP status codes are three-digit numbers returned by a web server to indicate the status of a requested resource. They are grouped into five classes:

* **1xx (Informational)**: The request was received and is continuing.
* **2xx (Success)**: The request was successfully received, understood, and accepted. (e.g., 200 OK)
* **3xx (Redirection)**: Further action needs to be taken to complete the request.
* **4xx (Client Error)**: The request contains bad syntax or cannot be fulfilled. (e.g., 404 Not Found)
* **5xx (Server Error)**: The server failed to fulfill an apparently valid request.

**10. What is REST API?**

REST (Representational State Transfer) is an architectural style for designing networked applications. A REST API is an application programming interface that adheres to the principles of REST. It uses standard HTTP methods (like

GET, POST, PUT, DELETE) to perform operations on resources, which are typically represented by URLs