Here are 20 unique **JavaScript code challenges** covering the topics related to **JSON**, **AJAX**, **Asynchronous Programming**, and **APIs**:

1. Parse JSON String into Object

- Write a JavaScript code that uses `JSON.parse()` to convert a JSON string into a JavaScript object.

2. Stringify JavaScript Object into JSON

- Write a function that takes a JavaScript object and converts it into a JSON string using `JSON.stringify()`.

3. Display User Data Using JSON

- Create a JSON string representing a user with properties like `name`, `age`, and `email`. Parse the JSON string and display the user's information in the console.

4. Parse JSON and Modify Data

- Parse a JSON string of a product, update its price, and then display the updated product information.

5. Convert JavaScript Array to JSON

- Write a function that takes an array of objects and converts it into a JSON string using `JSON.stringify()`.

6. JSON Validation

- Write a function that checks if a string is valid JSON before attempting to parse it. Use a `try...catch` block.

7. Handle API Response with JSON

- Fetch data from a JSON placeholder API (`https://jsonplaceholder.typicode.com/users`), parse the JSON response, and log the first user's name.

8. Using XMLHttpRequest for JSON API

- Write a function that uses `XMLHttpRequest` to fetch data from a public API that returns a JSON response. Parse and log the result.

9. Create JSON Object from Form Data

- Create a JSON object from the values in an HTML form (e.g., name, email, and message fields) and log the JSON string.

10. Fetch JSON Data and Display in HTML

- Use the `fetch()` API to retrieve JSON data from a remote server and display the fetched information in an HTML list.

11. Display Multiple User Data from API

- Use `fetch()` to get an array of users from an API and display their names and emails in an HTML table.

12. AJAX Request to Send JSON Data

- Create an AJAX request using `XMLHttpRequest` that sends a POST request with JSON data (such as a new user's information) to a server.

13. Handling Asynchronous Data with Promises

- Write a function that uses a `Promise` to simulate an asynchronous operation that fetches JSON data, and then resolve it by logging the data to the console.

14. Async-Await with Fetch API

- Use the `fetch()` API with `async/await` to retrieve data from a public API (like JSONPlaceholder) and log the response.

15. Convert Nested Object to JSON

- Write a function that converts a nested JavaScript object (such as a user with address and contact details) into a JSON string using `JSON.stringify()`.

16. Send JSON Data via Fetch POST Request

- Use the `fetch()` API to send a POST request with JSON data to an API endpoint. Log the response or success message.

17. Use Fetch API with Query Parameters

- Use `fetch()` to retrieve data from an API with query parameters (e.g., `?id=123`). Log the returned JSON response.

18. Use XMLHttpRequest to Get JSON Data

- Write a function using `XMLHttpRequest` to fetch JSON data from an API and log the data to the console.

19. Asynchronous Data Fetching with Callback

- Write a function that fetches data asynchronously using `XMLHttpRequest` with a callback function to handle the response and display the data.

20. Error Handling with Fetch and JSON

- Use the `fetch()` API to retrieve JSON data. If the response is not successful (e.g., 404), handle the error by displaying an error message.