```
/*

Async Await is another way to handle asynchronous operations in JavaScript. It uses the try and catch blocks to handle errors and ensure that your code does not break when an error occurs.

*/

// Define an asynchronous function named getJoke This function will fetch a random joke from an external API.

const getJoke = async () => {

    // Show a loading message while the joke is being fetched

    // This gives feedback to the user that something is happening

    document.getElementById('joke').innerText = 'Loading...';

    // Use try...catch to handle any errors that might occur during the fetch process

    try {

        // Use the fetch API to make a GET request to the joke API

        // 'await' pauses the function until the response is received

        const response = await fetch('https://official-joke-api.appspot.com/random_joke');

        // Convert the response to JSON format

        // Again, 'await' waits for this conversion to complete

        const jokesData = await response.json();

        // Display the joke on the webpage

        // The joke has two parts: setup and punchline

        document.getElementById('joke').innerText = `${jokesData.setup} - ${jokesData.punchline}`;
```

```
38
39      } catch (error) {
40
41          // If any error occurs (e.g., network issue, API down), this block will run
42
43          // Show a friendly error message to the user
44
45          document.getElementById('joke').innerText = 'Oops! Could not fetch a joke.';
46
47          // Log the error details to the console for debugging
48
49          console.error('Error fetching joke:', error.message);
50      }
51  }
```