Here are four examples of error handling in JavaScript using different techniques:

1. **Try-Catch Block:**

~~~
   try {
       // Code that may throw an error
       let result = 10 / 0;
       console.log(result);
   } catch (error) {
       // Handle the error
       console.error("An error occurred:", error.message);
   }
   ```
~~~

2. **Error Object:**
   ```javascript
   function divideNumbers(a, b) {
       if (b === 0) {
           throw new Error("Division by zero is not allowed");
       }
       return a / b;
   }

   try {
       let result = divideNumbers(10, 0);
       console.log(result);
   } catch (error) {
       console.error("An error occurred:", error.message);
   }
   ```

3. **Promise Rejection Handling:**
   ```javascript
   function fetchData() {
       return new Promise((resolve, reject) => {
           // Simulate an error condition
           reject(new Error("Failed to fetch data"));
       });
   }

   fetchData()
       .then(data => console.log(data))
       .catch(error => console.error("An error occurred:", error.message));
   ```

4. **Async/Await:**
   ```javascript
   async function fetchData() {
       // Simulate an asynchronous operation that may throw an error
       throw new Error("Failed to fetch data");
   }

   (async () => {
       try {
           let result = await fetchData();
           console.log(result);
       } catch (error) {
           console.error("An error occurred:", error.message);
       }
   })();
   ```

These examples cover various scenarios, including synchronous errors with try-catch, custom error handling using the `Error` object, promise rejection handling, and asynchronous error handling using async/await.