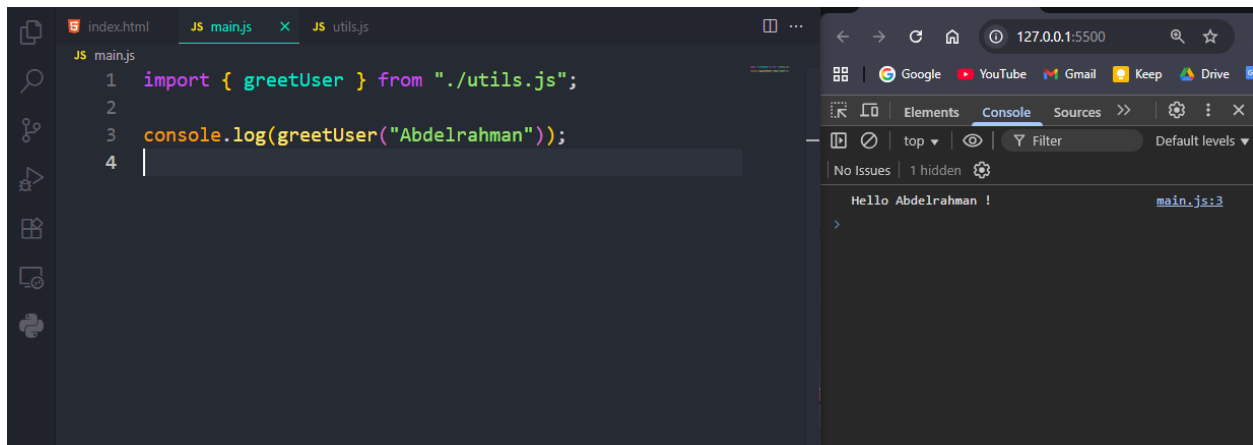


JavaScript Day-8 Assignments

1. Create a module file called 'utils.js' that exports a function named 'greetUser' which takes a name parameter and returns "Hello, [name]!". Then import and use this function in another file.



```
JS main.js
1 import { greetUser } from "./utils.js";
2
3 console.log(greetUser("Abdelrahman"));
4
```

127.0.0.1:5500

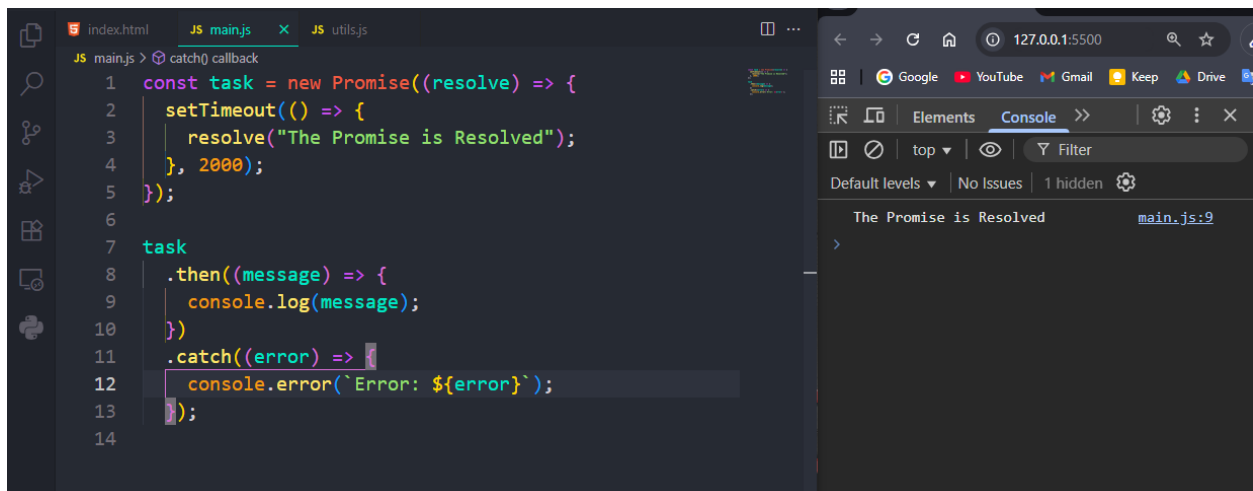
Google YouTube Gmail Keep Drive

Elements Console Sources >> Filter Default levels

No Issues 1 hidden

Hello Abdelrahman ! [main.js:3](#)

2. Write a Promise that resolves after 2 seconds with the message "Task completed!". Use .then() to log the result to the console.



```
JS main.js > catch() callback
1 const task = new Promise((resolve) => {
2   setTimeout(() => {
3     resolve("The Promise is Resolved");
4   }, 2000);
5 });
6
7 task
8   .then((message) => {
9     console.log(message);
10  })
11  .catch((error) => {
12    console.error(`Error: ${error}`);
13  });
14
```

127.0.0.1:5500

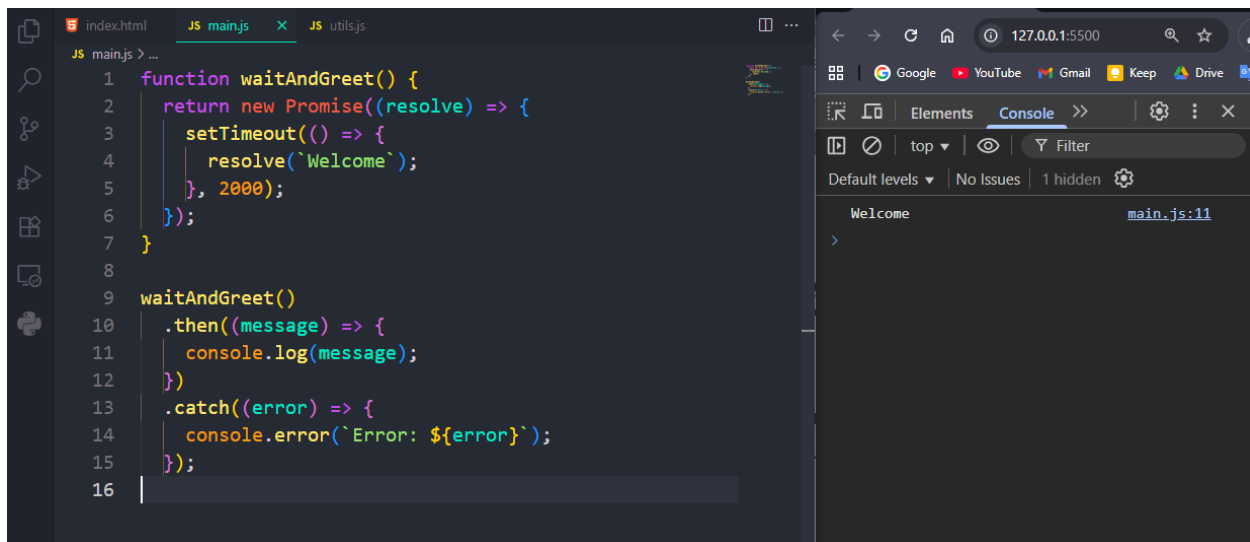
Google YouTube Gmail Keep Drive

Elements Console >> Filter Default levels

No Issues 1 hidden

The Promise is Resolved [main.js:9](#)

3. Create an async function called 'waitAndGreet' that uses setTimeout with a Promise to wait 1 second, then returns "Welcome!".

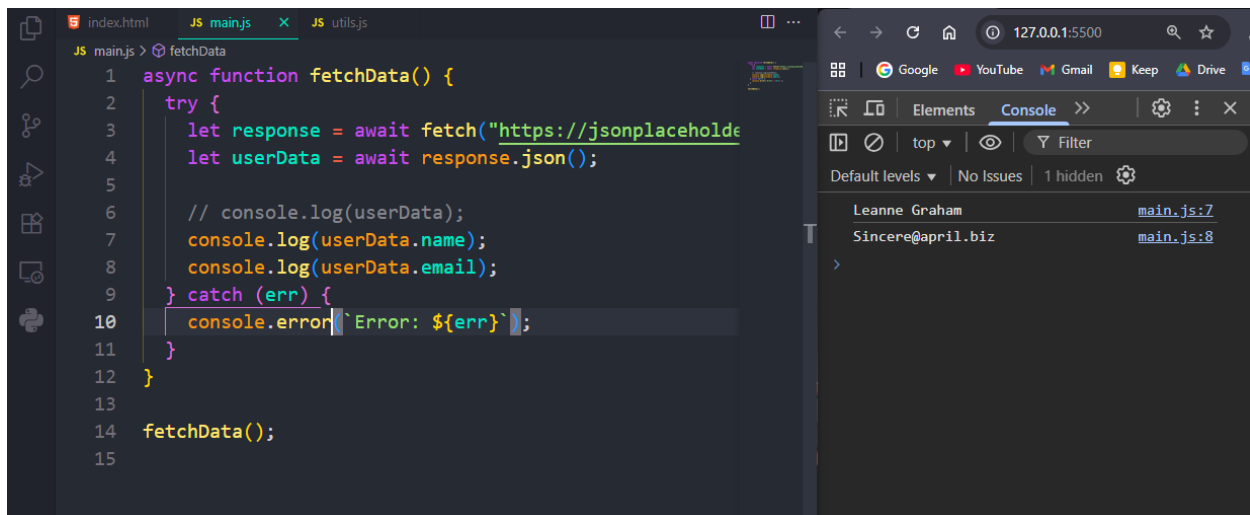


The screenshot shows a code editor with a file named 'main.js' containing the following JavaScript code:

```
1 function waitAndGreet() {  
2   return new Promise((resolve) => {  
3     setTimeout(() => {  
4       resolve('Welcome');  
5     }, 2000);  
6   });  
7 }  
8  
9 waitAndGreet()  
10 .then((message) => {  
11   console.log(message);  
12 })  
13 .catch((error) => {  
14   console.error(`Error: ${error}`);  
15 })  
16
```

The browser console on the right shows the output 'Welcome' from 'main.js:11'.

4. Write an async function that fetches user data from 'https://jsonplaceholder.typicode.com/users/1' and logs the user's name and email to the console.

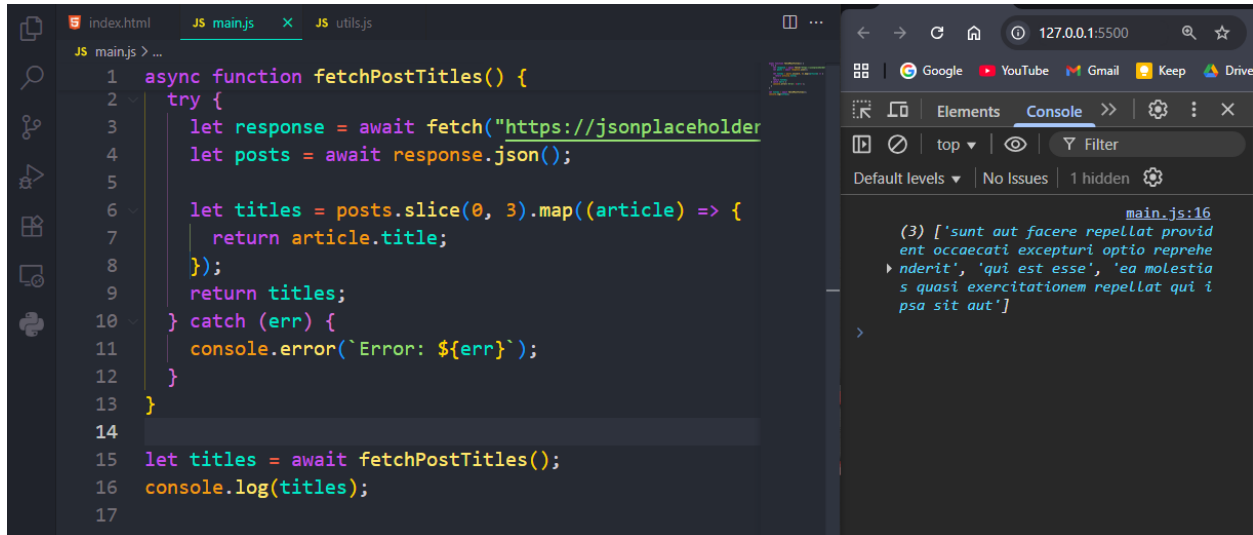


The screenshot shows a code editor with a file named 'main.js' containing the following JavaScript code:

```
1 async function fetchData() {  
2   try {  
3     let response = await fetch("https://jsonplaceholder  
4     let userData = await response.json();  
5  
6     // console.log(userData);  
7     console.log(userData.name);  
8     console.log(userData.email);  
9   } catch (err) {  
10    console.error(`Error: ${err}`);  
11  }  
12 }  
13  
14 fetchData();  
15
```

The browser console on the right shows the output 'Leanne Graham' from 'main.js:7' and 'Sincere@april.biz' from 'main.js:8'.

5. Create a function that fetches the first 3 posts from 'https://jsonplaceholder.typicode.com/posts' and returns only their titles as an array.



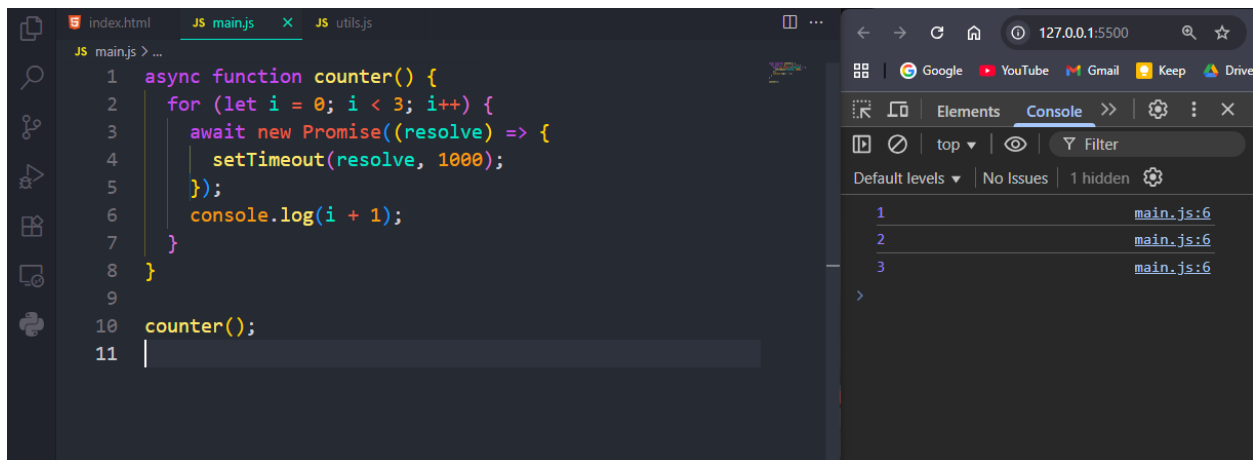
The screenshot shows a code editor with a file named `main.js`. The code defines an asynchronous function `fetchPostTitles` that uses `fetch` to retrieve data from `https://jsonplaceholder.typicode.com/posts`. It then slices the first three items and maps them to their titles. The function is called, and the titles are logged to the console. The browser's developer console shows the output: an array of three Latin titles.

```
1 async function fetchPostTitles() {
2   try {
3     let response = await fetch("https://jsonplaceholder.typicode.com/posts");
4     let posts = await response.json();
5
6     let titles = posts.slice(0, 3).map((article) => {
7       return article.title;
8     });
9     return titles;
10  } catch (err) {
11    console.error(`Error: ${err}`);
12  }
13 }
14
15 let titles = await fetchPostTitles();
16 console.log(titles);
17
```

Console output:

```
(3) ['sunt aut facere repellat provident occaecati excepturi optio reprehenderit', 'qui est esse', 'ea molestia sint quis qui nam repellat voluptatis illam']
```

6. Create a simple timer function using Promise that counts from 1 to 3, logging each number after 1 second intervals.



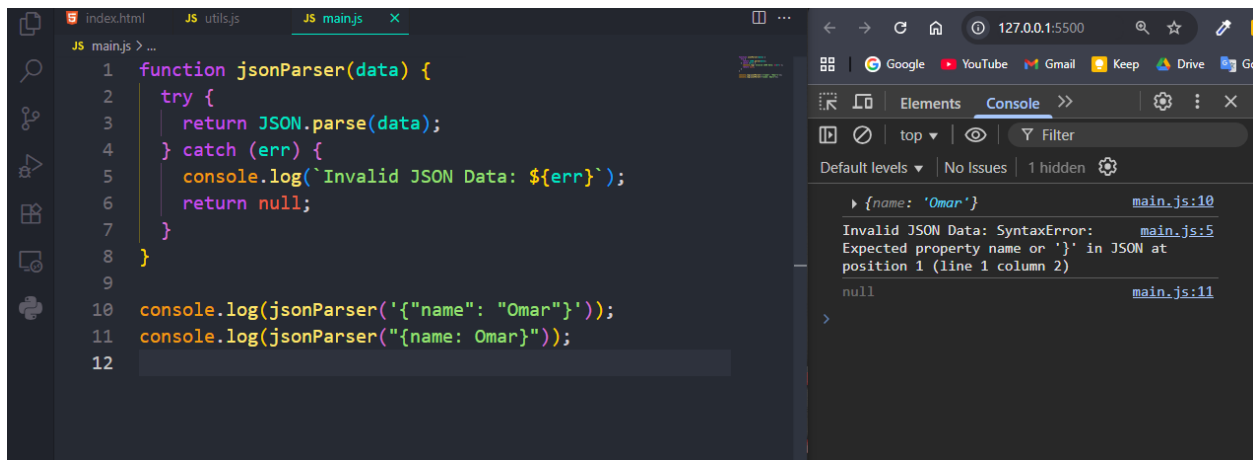
The screenshot shows a code editor with a file named `main.js`. The code defines an asynchronous function `counter` that uses a `for` loop and `Promise.resolve` with `setTimeout` to log the numbers 1, 2, and 3 at one-second intervals. The function is called, and the console shows the numbers being logged sequentially.

```
1 async function counter() {
2   for (let i = 0; i < 3; i++) {
3     await new Promise(resolve => {
4       setTimeout(resolve, 1000);
5     });
6     console.log(i + 1);
7   }
8 }
9
10 counter();
11
```

Console output:

```
1
2
3
```

7. Write a function that safely parses JSON data with try/catch. Test it with both valid JSON string '{"name": "Omar"}' and invalid JSON '{name: Omar}'.



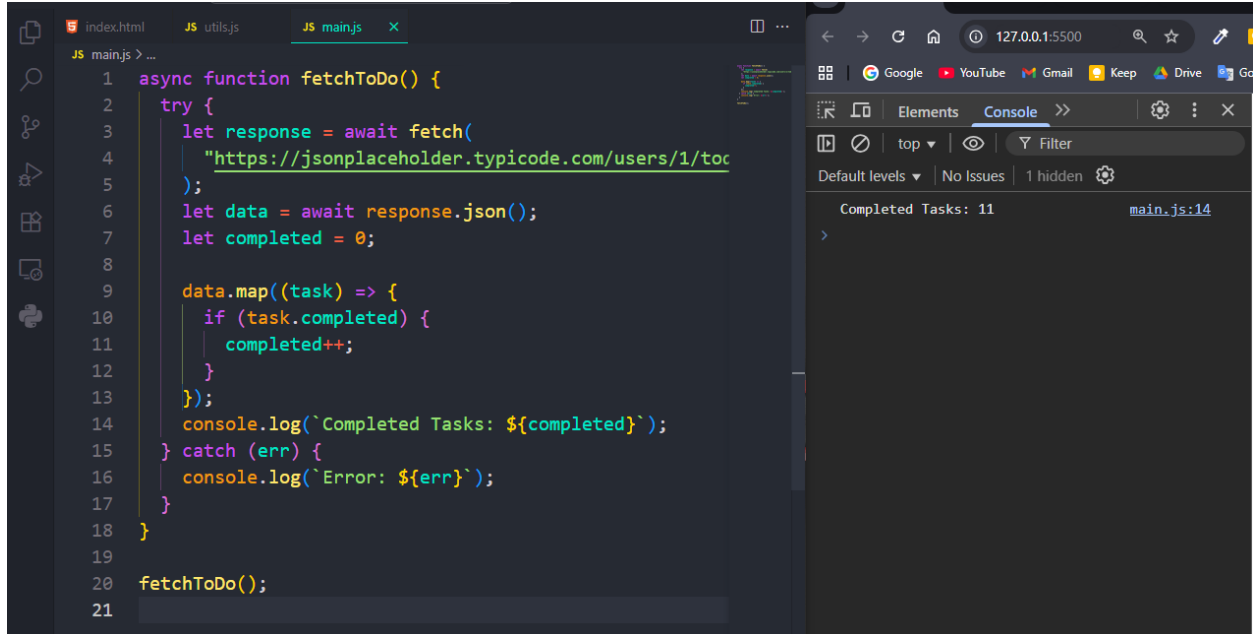
The screenshot shows a web browser with a JavaScript file named `main.js` open. The function `jsonParser` is defined with a try/catch block. It attempts to parse a JSON string using `JSON.parse`. If it fails, it logs an error message and returns `null`. The function is tested with two inputs: a valid JSON string and an invalid one. The console shows the results of these tests.

```
1 function jsonParser(data) {
2   try {
3     return JSON.parse(data);
4   } catch (err) {
5     console.log(`Invalid JSON Data: ${err}`);
6     return null;
7   }
8 }
9
10 console.log(jsonParser('{"name": "Omar"}'));
11 console.log(jsonParser('{name: Omar}'));
12
```

Console output:

```
{name: 'Omar'}
Invalid JSON Data: SyntaxError:
Expected property name or '}' in JSON at
position 1 (line 1 column 2)
null
```

8. Create an async function that fetches data from 'https://jsonplaceholder.typicode.com/users/1/todos', converts it to JSON, and returns the count of completed todos.



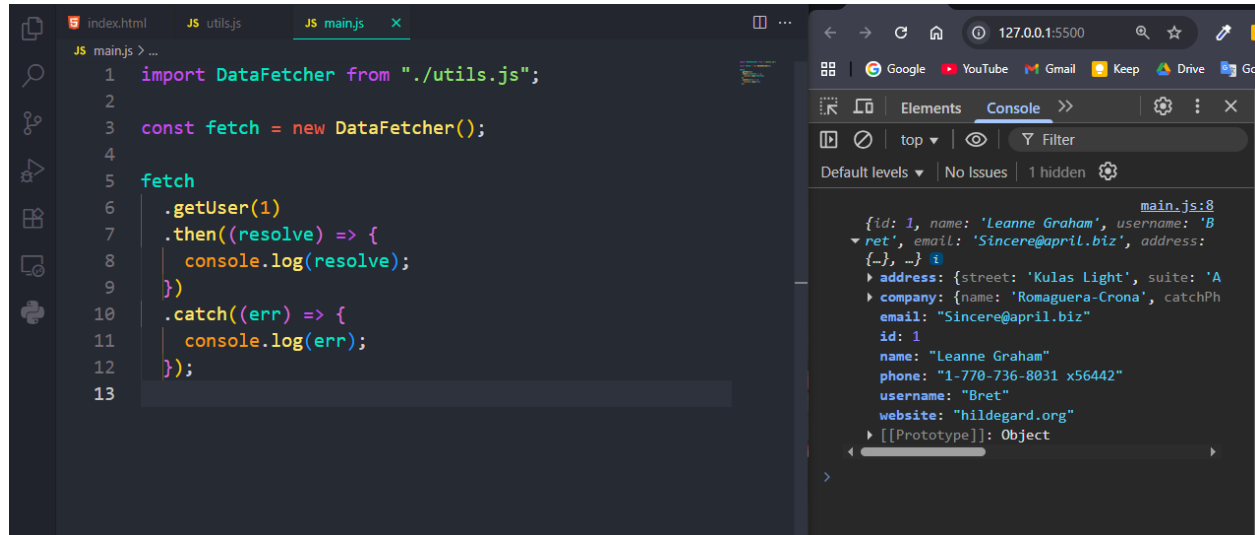
The screenshot shows a web browser with a JavaScript file named `main.js` open. The function `fetchToDo` is defined as an async function. It uses `fetch` to get data from the specified URL, converts it to JSON, and then counts the number of completed tasks. The function is called at the bottom of the file. The console shows the count of completed tasks.

```
1 async function fetchToDo() {
2   try {
3     let response = await fetch(
4       "https://jsonplaceholder.typicode.com/users/1/todos"
5     );
6     let data = await response.json();
7     let completed = 0;
8
9     data.map((task) => {
10       if (task.completed) {
11         completed++;
12       }
13     });
14     console.log(`Completed Tasks: ${completed}`);
15   } catch (err) {
16     console.log(`Error: ${err}`);
17   }
18 }
19
20 fetchToDo();
21
```

Console output:

```
Completed Tasks: 11
```

9. Build a simple module that exports a default class called 'DataFetcher' with a method 'getUser(id)' that fetches and returns user data from the JSONPlaceholder API.



The screenshot shows a web browser with a dark theme. On the left, a code editor displays the following JavaScript code in a file named `main.js`:

```
1 import DataFetcher from "./utils.js";
2
3 const fetch = new DataFetcher();
4
5 fetch
6   .getUser(1)
7   .then((resolve) => {
8     console.log(resolve);
9   })
10  .catch((err) => {
11    console.log(err);
12  });
13
```

On the right, the browser's developer console is open, showing the output of the code. The output is a JSON object representing a user from the JSONPlaceholder API:

```
main.js:8
{id: 1, name: 'Leanne Graham', username: 'Bret', email: 'Sincere@april.biz', address: {street: 'Kulas Light', suite: 'A', city: 'New York', zipcode: '10014-2045', geo: {lat: '42.152165', lng: '-71.985356'}}, phone: '1-770-736-8031 x56442', website: 'hildegard.org', company: {name: 'Romaguera-Crona', catchPhrase: 'Facilisi quia', address: '2311 New Circle Drive', city: 'Vancouver', state: 'British Columbia', postalCode: 'V6C 5R9', geo: {lat: '49.282745', lng: '-123.120265'}}}, ->
```