Managing Data with JSON

Scenario: In this activity, you will fetch, parse, manipulate, and store JSON data using JavaScript. These tasks are essential for building interactive web applications that handle data dynamically and use localStorage for persistence.

Step 1: Create a New HTML File

1. Select File
2. Select New File...
3. Name it **index.html**
4. Press enter
5. Select OK

Step 2: Create the HTML Structure

1. Use <!DOCTYPE html> for HTML5.
2. Add <html>, <head>, and <body> tags to structure the content.
3. Inside the <head> section, include:
    a. A <title> tag with a title.
    b. A <meta> tag with charset="UTF-8".
4. Add the following elements inside the <body> tag:
    - A heading (<h1>) titled "Users List".
    - A element with the ID users-container.
    - A <script> element to link your JavaScript file.

Step 3: Create a New JavaScript File

1. Select File
2. Select New File...
3. Name it script.js
4. Press enter
5. Select OK

Step 4: Fetch and Display Data from an API

1. Write JavaScript to fetch data from the following API: https://jsonplaceholder.typicode.com/users
2. After fetching the data, display each user's name and email inside the with the ID users-container.
3. If the fetch request fails, log an error message to the console.

Step 5: Simulate and Parse JSON Data

1. Create a JSON string containing two users:
    - The first user: "Alice", age 25.
    - The second user: "Bob", age 30.
2. Parse the JSON string into a JavaScript object using JSON.parse().
3. Log Alice's name and Bob's age to the console.

Step 6: Convert JavaScript Objects to JSON

1. Create a JavaScript object representing a user with the following properties:
    - Name: "Charlie"
    - Age: 28
    - isActive: true
2. Convert the object to a JSON string using JSON.stringify().
3. Log the JSON string to the console.

Step 7: Store and Retrieve Data Using localStorage

1. Create a settings object with the following properties:
    - Theme: "dark"
    - Language: "en"
2. Store the object in localStorage as a JSON string.
3. Retrieve and parse the stored data, then log the theme and language to the console.

Step 8: Run your Code

- Click Go Live (in the lower right of the lab).
- A new tab should open up and display your webpage!
- If your code is not running as you expected, go to the next item to see the correct code.

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Users Page</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Users List</h1>

  <div class="section">
    <h2>Local Users (Alice & Bob)</h2>
    <div id="local-users"></div>
  </div>

  <div class="section">
    <h2>Charlie (from Object)</h2>
    <div id="charlie-data"></div>
  </div>

  <div class="section">
    <h2>Settings</h2>
    <div id="settings-data"></div>
  </div>

  <div class="section">
    <h2>Fetched Users</h2>
    <div id="users-container"></div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS:

```css
body {
  font-family: Arial, sans-serif;
  background-color: #1e1e2f;
  color: #f0f0f0;
  margin: 0;
  padding: 20px;
}

h1 {
  color: #61dafb;
  text-align: center;
  margin-bottom: 30px;
}

.section {
  background-color: #2a2a40;
  padding: 20px;
  border-radius: 10px;
  margin-bottom: 20px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
}

.section h2 {
  color: #ffd700;
  margin-top: 0;
}

p {
  margin: 6px 0;
}

#users-container p,
#local-users p,
#charlie-data p,
#settings-data p {
  padding-left: 10px;
}
```

JavaScript:

```
// DOM element selection

const usersContainer = document.getElementById('users-container');
const localUsersContainer = document.getElementById('local-users');
const charlieContainer = document.getElementById('charlie-data');
const settingsContainer = document.getElementById('settings-data');

// Local JSON string and parsing

const jsonString = `
[
  { "name": "Alice", "age": 25 },
  { "name": "Bob", "age": 30 }
]
`;

const localUsers = JSON.parse(jsonString);
console.log("Alice's name:", localUsers[0].name);
console.log("Bob's age:", localUsers[1].age);

// Display Alice and Bob
localUsers.forEach(user => {
  const p = document.createElement('p');
  p.textContent = `${user.name}, Age: ${user.age}`;
  localUsersContainer.appendChild(p);
});

// Object creation and conversion to JSON

const userCharlie = {
  name: "Charlie",
  age: 28,
  isActive: true
};

const charlieJSON = JSON.stringify(userCharlie);
console.log("Charlie as JSON:", charlieJSON);

// Display Charlie
const charlieData = JSON.parse(charlieJSON);
const charlieInfo = document.createElement('p');
charlieInfo.textContent = `${charlieData.name}, Age: ${charlieData.age},
Active: ${charlieData.isActive}`;
charlieContainer.appendChild(charlieInfo);

// Fetching and displaying external API data

fetch('https://jsonplaceholder.typicode.com/users')
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not OK');
    }
    return response.json();
  })
  .then(users => {
    users.forEach(user => {
      const userElement = document.createElement('p');
```

```
      userElement.textContent = `${user.name} (${user.email})`;
      usersContainer.appendChild(userElement);
    });
  })
  .catch(error => {
    console.error('Failed to fetch users:', error);
  });

// LocalStorage: storing and retrieving settings

const settings = {
  theme: "dark",
  language: "en"
};

localStorage.setItem("settings", JSON.stringify(settings));

const savedSettings = JSON.parse(localStorage.getItem("settings"));

console.log("Theme:", savedSettings.theme);
console.log("Language:", savedSettings.language);

// Display settings
const themePara = document.createElement('p');
themePara.textContent = `Theme: ${savedSettings.theme}`;
settingsContainer.appendChild(themePara);

const languagePara = document.createElement('p');
languagePara.textContent = `Language: ${savedSettings.language}`;
settingsContainer.appendChild(languagePara);
```