Task 2: Dynamic Content Creation for a Profile Page

Objective:

Define an array of objects for profiles, iterate over the array to generate HTML cards for each profile, and display these cards on the webpage. This task aims to enhance your skills in using JavaScript for dynamic content creation and manipulation of the DOM.

Pre-requisites:

- Basic understanding of HTML and JavaScript
- Familiarity with a code editor like Visual Studio Code

Concepts Covered:

- Defining an Array of Objects
- Iterating Over an Array and Generating HTML Content
- Displaying Dynamic Content on a Webpage

Concepts:

1. **Defining an Array of Objects:**

   Create a constant array containing objects. Each object should represent a profile with properties like `name`, `bio`, and `skills`.

```
const profiles = [
    {
        name: "John Doe",
        bio: "Web Developer with a passion for creating interactive and dynamic 
web applications.",
        skills: ["HTML", "CSS", "JavaScript", "React"]
    },
    {
        name: "Jane Smith",
        bio: "Graphic Designer who loves bringing creativity and innovation to 
web designs.",
        skills: ["Photoshop", "Illustrator", "CSS", "HTML"]
    }
];
```

2. **Iterating Over the Array and Generating HTML Content:**

   Use a `forEach` loop to iterate over the array. For each item, create a new `div` element representing a card, and populate it with the item's details.

```
profiles.forEach(profile => {
    const card = document.createElement('div');
    card.classList.add('card');

    const nameElement = document.createElement('h2');
    nameElement.innerText = profile.name;
    card.appendChild(nameElement);

    const bioElement = document.createElement('p');
    bioElement.innerText = profile.bio;
    card.appendChild(bioElement);

    const skillsElement = document.createElement('ul');
    profile.skills.forEach(skill => {
        const skillItem = document.createElement('li');
        skillItem.innerText = skill;
        skillsElement.appendChild(skillItem);
    });
    card.appendChild(skillsElement);

    document.querySelector('.container').appendChild(card);
});
```

3. **Displaying the Cards on the Webpage:**

   Create a container `div` in your HTML and append each card to this container using `appendChild`. Ensure the layout is visually appealing and readable.

```
<div class="container"></div>
```

Setup:

1. **Install Visual Studio Code (VS Code):**

   Download and install VS Code from Visual Studio Code.

2. **Web Browsers:**

   Use Google Chrome or Mozilla Firefox for viewing your webpage and utilizing their developer tools for debugging.

Tasks:

1. **Define an Array of Objects for Profiles (10 minutes):**
   - Create a constant array containing objects. Each object should represent a profile with properties like `name`, `bio`, and `skills`.
   - Example:

```
const profiles = [
    {
        name: "John Doe",
        bio: "Web Developer with a passion for creating interactive and
dynamic web applications.",
        skills: ["HTML", "CSS", "JavaScript", "React"]
    },
    {
        name: "Jane Smith",
        bio: "Graphic Designer who loves bringing creativity and innovation
to web designs.",
        skills: ["Photoshop", "Illustrator", "CSS", "HTML"]
    }
];
```

2. **Iterate Over the Array and Generate HTML Cards (10 minutes):**
    - Use a `forEach` loop to iterate over your array.
    - For each item, create a new `div` element representing a card, and populate it with the item's details.
    - Example:

```
profiles.forEach(profile => {
    const card = document.createElement('div');
    card.classList.add('card');

    const nameElement = document.createElement('h2');
    nameElement.innerText = profile.name;
    card.appendChild(nameElement);

    const bioElement = document.createElement('p');
    bioElement.innerText = profile.bio;
    card.appendChild(bioElement);

    const skillsElement = document.createElement('ul');
    profile.skills.forEach(skill => {
        const skillItem = document.createElement('li');
        skillItem.innerText = skill;
        skillsElement.appendChild(skillItem);
    });
    card.appendChild(skillsElement);

    document.querySelector('.container').appendChild(card);
});
```

3. **Display the Cards on the Webpage (10 minutes):**
    - Create a container `div` in your HTML.
    - Append each card to this container using `appendChild`.
    - Ensure the layout is visually appealing and readable.

- Example:

  ```
  <div class="container"></div>
  ```

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile Cards</title>
    <style>
        .container {
            display: flex;
            flex-wrap: wrap;
            gap: 20px;
        }
        .card {
            background-color: #f0f0f0;
            border: 1px solid #ccc;
            border-radius: 10px;
            padding: 20px;
            width: 200px;
        }
        .card h2 {
            margin-top: 0;
            color: #ff5733;
        }
        .card ul {
            padding-left: 20px;
        }
        .card ul li {
            list-style-type: disc;
        }
    </style>
</head>
<body>
    <div class="container"></div>
    <script src="script.js"></script>
</body>
</html>
```

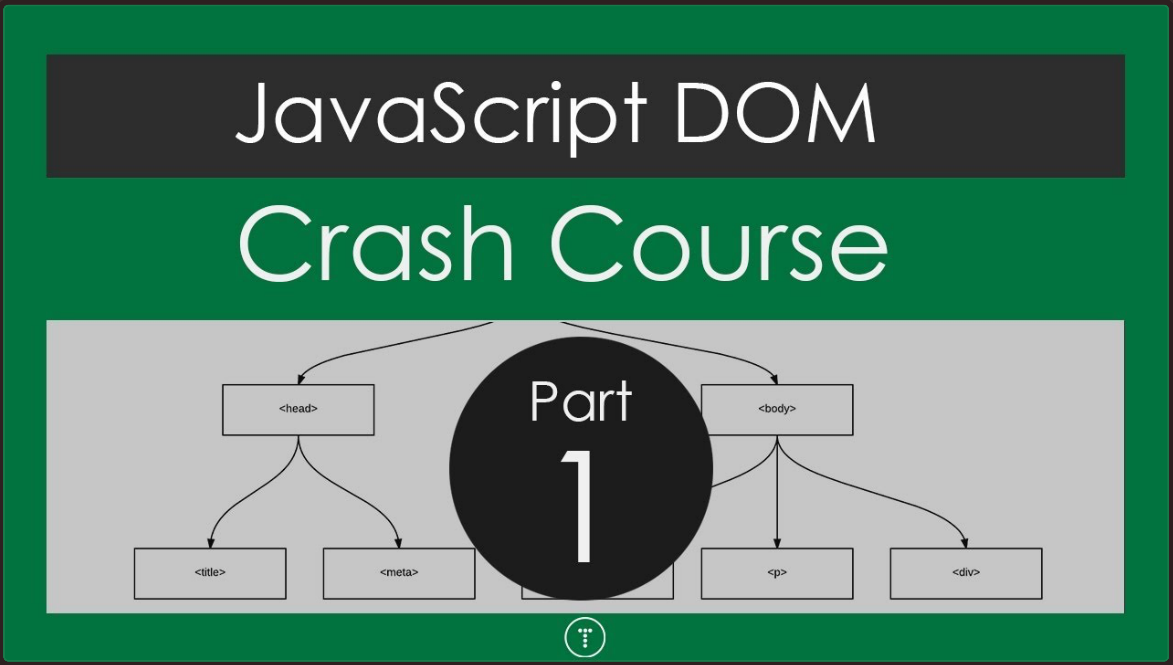
Instructions:

1. Write the required code in `index.html` and `script.js` .
2. Open the `index.html` file in your web browser to ensure the code displays correctly.
3. Use the browser's developer tools to debug and inspect the elements.

Resources:

- **Array - JavaScript | MDN**
  The Array object, as with arrays in other programming languages, enables storing a collecti...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
- **Introduction to the DOM - Web APIs | MDN**
  The Document Object Model (DOM) is the data representation of the objects that compri...
  https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- **Documentation for Visual Studio Code**
  Find out how to set-up and get the most from Visual Studio Code. Optimized for building and d...
  https://code.visualstudio.com/docs

Videos:

- 

  Traversy Media 39:01min 1,210,956 Views 22,421 Likes

GitHub Instructions:

1. **Open in Visual Studio Code:**
   After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, VSCode will open the repository directly. If prompted, select "Open" or "Allow" to open the repository in VSCode.
2. **Open the Terminal in VSCode:**
   In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.

3. **Complete the Task:**

   In VSCode, write your solution in the `index.html` and `script.js` files.

4. **Run and Test Your Code:**

   Open your `index.html` file in a web browser to ensure it works correctly. Use the following command:

   ```
   open index.html
   ```

5. **Commit Your Changes:**

   In the VSCode terminal, add your changes to git:

   ```
   git add index.html script.js
   ```

   Commit your changes with a meaningful message:

   ```
   git commit -m "Completed task 13"
   ```

6. **Push Your Changes to Your Repository:**

   Push your changes to your forked repository:

   ```
   git push origin main
   ```

7. **Create a Pull Request:**

   Go to your repository on GitHub.

   Click on the "Pull Requests" tab.

   Click the "New Pull Request" button.

   Ensure the base repository is the original template repository and the base branch is `main` .

   Ensure the head repository is your forked repository and the compare branch is `main` .

   Click "Create Pull Request".

   Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing index.html and script.js

# Navigate to the directory containing index.html
cd path/to/your/index.html

# Run your code
open index.html

# Add, commit, and push your changes
git add index.html script.js
git commit -m "Completed task 2"
git push origin main

# Create a pull request on GitHub
```