Task 1: Setting Up the Website Structure

Objective:

Initialize and link a JavaScript file, create and append basic HTML elements using JavaScript, and apply styles to these elements. This task aims to provide a foundational understanding of manipulating the DOM (Document Object Model) with JavaScript to dynamically generate content and apply styles.

Pre-requisites:

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

Concepts Covered:

- Initializing and Linking a JavaScript File
- Creating and Appending HTML Elements using JavaScript
- Applying Styles to Elements using JavaScript

Concepts:

1. **Initializing and Linking a JavaScript File:**

   Create a new JavaScript file and link it to your HTML file to enable JavaScript functionality.

2. **Creating and Appending HTML Elements using JavaScript:**

   Use JavaScript to create new HTML elements and append them to the body of the document.

   ```
   const newDiv = document.createElement('div');
   const newH1 = document.createElement('h1');
   const newP = document.createElement('p');

   newH1.innerText = 'John Doe';
   newP.innerText = 'Welcome to my profile page.';

   newDiv.appendChild(newH1);
   newDiv.appendChild(newP);
   document.body.appendChild(newDiv);
   ```

3. **Applying Styles to Elements using JavaScript:**

   Set style properties for the created elements using JavaScript to enhance their appearance.

```
newDiv.style.backgroundColor = '#f0f0f0';
newDiv.style.padding = '20px';
newDiv.style.borderRadius = '10px';

newH1.style.color = '#ff5733';
newH1.style.fontSize = '2em';

newP.style.color = '#333';
newP.style.fontSize = '1em';
```

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. **Initialize and Link a JavaScript File (10 minutes):**
   - Start by creating a new JavaScript file named `script.js` .
   - In your HTML file, link this script at the bottom of the body tag using:

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

2. **Create and Append Basic HTML Elements using JavaScript (10 minutes):**
   - In `script.js` , use `document.createElement` to create new HTML elements like `div` , `h1` , and `p` .
   - Set their `innerText` property to add content.
   - Use `document.body.appendChild` to add these elements to the body of your HTML.
   - Example:

   ```
   const newDiv = document.createElement('div');
   const newH1 = document.createElement('h1');
   const newP = document.createElement('p');

   newH1.innerText = 'John Doe';
   newP.innerText = 'Welcome to my profile page.';

   newDiv.appendChild(newH1);
   newDiv.appendChild(newP);
   document.body.appendChild(newDiv);
   ```

3. **Apply Styles to Elements using JavaScript (10 minutes):**

  - For each created element, set its style properties like `element.style.color` and `element.style.fontSize`.
  - Experiment with different CSS properties to understand how styling works in JavaScript.
  - Example:

    ```
    newDiv.style.backgroundColor = '#f0f0f0';
    newDiv.style.padding = '20px';
    newDiv.style.borderRadius = '10px';

    newH1.style.color = '#ff5733';
    newH1.style.fontSize = '2em';

    newP.style.color = '#333';
    newP.style.fontSize = '1em';
    ```

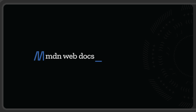
Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile Page Setup</title>
</head>
<body>
    <!-- Script link at the bottom of the body -->
    <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:

- 

  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

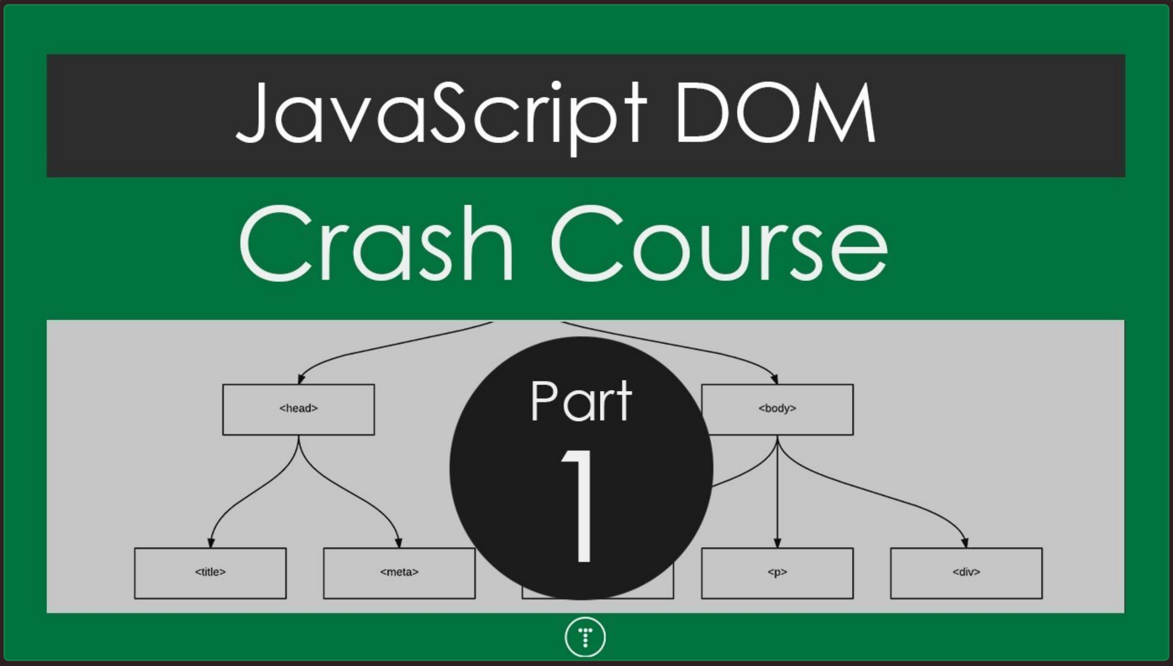
-

  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 1,198,799 Views 22,283 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 12"
```

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 1"
git push origin main

# Create a pull request on GitHub
```