Task 4: Using Template Literals and Destructuring

Objective:

Create a program that utilizes ES6 template literals and destructuring to generate a personalized greeting message. The program should accept an object containing user details (name, age, and country) and output a message that includes these details in a formatted string.

Pre-requisites:

- Basic JavaScript (variables, functions)
- Template literals
- Destructuring (arrays and objects)

Concepts Covered:

- Template literals
- Destructuring assignment

Setup:

Install Node.js:

- Ensure Node.js is installed on your machine. You can download it from nodejs.org.

Tasks:

1. **Generate Greeting Message:**
    - **Task:**
        - Define a function named `generateGreeting`.
        - The function should accept an object with the following properties: name, age, and country.
        - Use object destructuring to extract the properties from the input object.
        - Create a formatted string using template literals that includes the user's name, age, and country.
        - Return the formatted string.
    - **Outcome:**
        - Ensure the function generates the correct personalized greeting message using template literals and destructuring.

Example:

JavaScript File (`index.js`):

```
function generateGreeting(user) {
    const { name} = user;
    return `Hello, my name is ${name};
}

// Example Usage
console.log(generateGreeting({ name: "Alice"})); // Output: "Hello, my name is
Alice,"
```

Instructions:

- **Perform the following tasks:**
  - Write the required code in `index.js`.
  - Run the file using Node.js to ensure the code executes without errors and demonstrates the use of template literals and destructuring.

Example Input:

1. Input: `{ name: "Alice", age: 25, country: "Wonderland" }`

Expected Output:

1. Output: `"Hello, my name is Alice, I am 25 years old, and I come from Wonderland."`

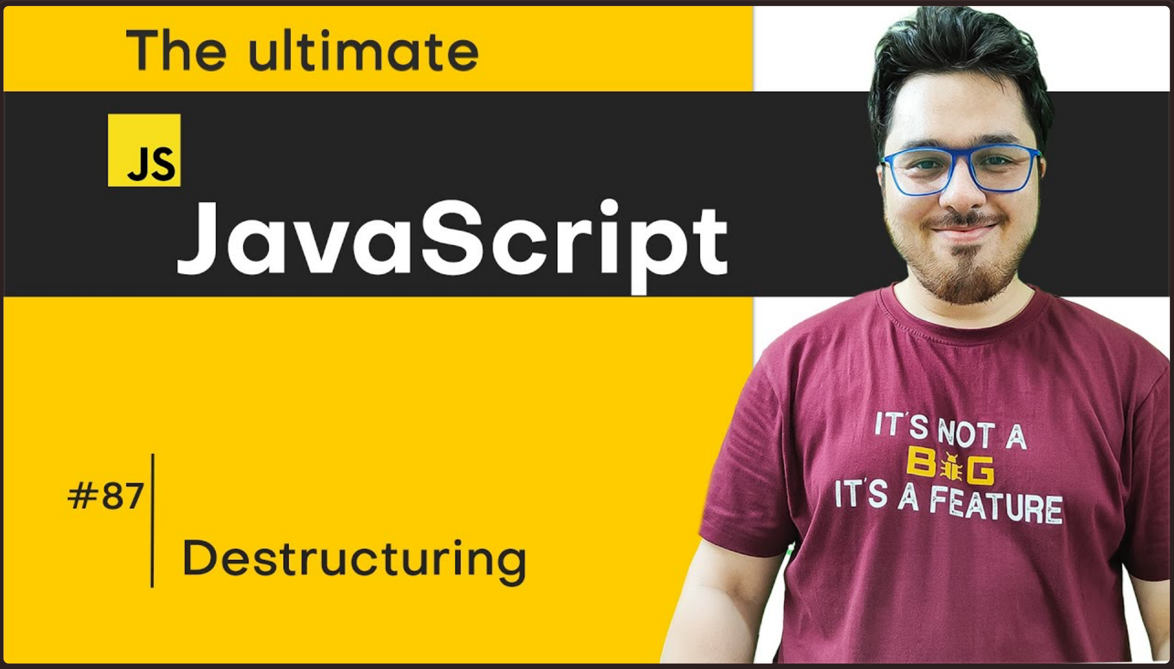
Resources:

- Template literals (Template strings) - JavaScript | MDN
  Template literals are literals delimited with backtick (`) characters, allowing for multi-line st...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals
- Destructuring assignment - JavaScript | MDN
  The destructuring assignment syntax is a JavaScript expression that makes it possible to u...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

videos:

-

  Telusko 5:06min 87,275 Views 1,920 Likes

- 

  CodeWithHarry 15:01min 98,808 Views 3,119 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, Visual Studio Code (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.js` file.
4. **Run and Test Your Code:**
    - In the VSCode terminal, navigate to the directory containing `index.js` .
    - Run your code to ensure it works correctly. Use the following command:

    ```
    node index.js
    ```

5. **Commit Your Changes:**
    - In the VSCode terminal, add your changes to git:

    ```
    git add index.js
    ```

    - Commit your changes with a meaningful message:

    ```
    git commit -m "Completed task 4"
    ```

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.js

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

# Run your code
node index.js

# Add, commit, and push your changes
git add index.js
git commit -m "Completed task 4"
git push origin main

# Create a pull request on GitHub
```