Task 1: Basic Syntax and DOM Manipulation

Level: Beginner

Description:

Create a simple JavaScript application that processes user input for their name, age, and whether they are a student. The application should display a personalised message based on the input. Additionally, implement a function to toggle a boolean value for a separate task.

Objective:

Implement basic JavaScript concepts including variables, data types, conditional statements, loops, and functions.

Pre-requisites:

- Understanding of basic JavaScript (variables, data types, functions)
- Familiarity with conditional statements
- Knowledge of loops

Concepts Covered:

- Variables and data types
- Conditional statements
- Loops

Instructions:

1. Define variables to store the user's name, age, and student status.
2. Create a function to generate a personalised message based on the input values.
3. Use conditional statements to check if the user is a student and create a personalised message.
4. Use a loop to repeat the personalised message based on the user's age (if age is less than or equal to 10, repeat the message that many times).
5. Implement a function to toggle a boolean value.

Example Input:

Set 1:

- Name: `"Alice"`
- Age: `5`
- Is Student: `true`

Set 2:

- Name: `"Bob"`
- Age: `15`

- Is Student: `false`

Set 3:

- Name: `"Charlie"`
- Age: `8`
- Is Student: `true`

Expected Output:

Set 1:

```
"Hello, my name is Alice and I am a student."

"Hello, my name is Alice and I am a student."

"Hello, my name is Alice and I am a student."

"Hello, my name is Alice and I am a student."

"Hello, my name is Alice and I am a student."
```

Set 2:

```
"Hello, my name is Bob."
```

Set 3:

```
"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."

"Hello, my name is Charlie and I am a student."
```

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. **Create Your Directory**:

In the VSCode terminal, navigate to the tasks/student directory:

```
Unset
cd tasks/student
```

Create a new directory with your GitHub username (replace your-username with your actual GitHub username):

```
Unset
mkdir your-username cd your-username
```

4. **Create a Directory for the Task**:

Create a new directory for the task, e.g., task1:

```
Unset
mkdir task1 cd task1
```

5. **Complete the Task**:
    - In VSCode, create a new file named index.js in your task1 directory and Write your solution in the index.js file.

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

In the VSCode terminal, navigate back to the root directory of the repository:

```
Unset
cd ../../../../
```

Run your code to ensure it works correctly. Use the following commands:

```
Unset
node tasks/student/your-username/task1/index.js "Alice" 5 true

node tasks/student/your-username/task1/index.js "Bob" 15 false

node tasks/student/your-username/task1/index.js "Charlie" 8 true

node tasks/student/your-username/task1/index.js toggle
```

7. **Commit Your Changes**:

In the VSCode terminal, add your changes to git:

```
Unset
git add tasks/student/your-username/task1/index.js
```

Commit your changes with a meaningful message:

```
Unset
git commit -m "Completed task 1"
```

Push your changes to your repository:

```
Unset
git push origin main
```

8. **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

```
Unset
# Open in Visual Studio Code

# Open terminal in VSCode

# Navigate to the tasks/student directory
cd tasks/student

# Create your own directory and navigate to it
mkdir your-username
cd your-username

# Create a directory for the task and navigate to it
mkdir task1
cd task1

# Complete the task by editing index.js

# Navigate back to the root directory
cd ../../../../

# Run your code
node tasks/student/your-username/task1/index.js "Alice" 5 true
node tasks/student/your-username/task1/index.js "Bob" 15 false
node tasks/student/your-username/task1/index.js "Charlie" 8 true
node tasks/student/your-username/task1/index.js toggle

# Add, commit, and push your changes
git add tasks/student/your-username/task1/index.js
git commit -m "Completed task 1"
git push origin main

# Create a pull request on GitHub
```

Template Repository Structure

```
.
├── README.md
├── setup.sh
├── tasks/
|   └── student/
|       └── your-username/
|           ├── task1/
|           |   └── index.js
|           ├── task2/
|           |   └── index.js
|           └── ...
└── package.json
```

Example Solution:

JavaScript

```javascript
// Define variables
let name = "Alice";
let age = 5;
let isStudent = true;

// Function to generate a personalised message
function generateMessage(name, age, isStudent) {
    let message = `Hello, my name is ${name}`;
    if (isStudent) {
        message += ` and I am a student.`;
    }
```

```
    let output = '';
    for (let i = 0; i < Math.min(age, 10); i++) {
        output += `${message}\n`;
    }
    return output;
}

// Toggle function
let isActive = false;
function toggleActive() {
    isActive = !isActive;
    return isActive;
}

// Example usage
console.log(generateMessage(name, age, isStudent));
console.log(toggleActive()); // true
console.log(toggleActive()); // false
```

Different Example Input Sets:

Set 1:

- Name: `"Alice"`
- Age: `5`
- Is Student: `true`

Expected Output:

```
"Hello, my name is Alice and I am a student."
"Hello, my name is Alice and I am a student."
"Hello, my name is Alice and I am a student."
"Hello, my name is Alice and I am a student."
"Hello, my name is Alice and I am a student."
```

Set 2:

- Name: `"Bob"`
- Age: `15`
- Is Student: `false`

Expected Output:
`"Hello, my name is Bob."`

Set 3:

- Name: `"Charlie"`
- Age: `8`
- Is Student: `true`

Expected Output:
```
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
"Hello, my name is Charlie and I am a student."
```