Homework 1-1 (in class)
Create a Simple To-Do List

Project Description

Create a simple web application that allows users to create a simple to-do list. The project will consist of multiple steps, each building on the previous one.

Homework Submissions Requirement:

1. Create a fold named "**student id_HW1**". (ex: R12725088_HW1)
2. Each step should have its own folder.
3. A PDF document named README.pdf to explain what you have done in each step. (Attaching screenshots to help explain is recommended.)
4. Compressed the fold "student id_HW1" into a single zip file and uploaded it to COOL.

   The folder should look like this:

```
R12725088_HW1
├─ README.PDF
├─ folder 1
        ├─ index.html
        ├─ A.css
        ├─ B.js
├─ folder 2
        ├─ index.html
        ├─ A.css
        ├─ B.js
├─ folder 3
        ├─ .....
```

Steps

Step One: Build HTML Structure

1. Create a new HTML file and name it *index.html*.
2. Set up basic HTML structure including <head> and <body> tags.
3. Within the <head> tag, create the following information
    a. charset="UTF-8"
    b. RWD: name="viewport" content="width=device-width, initial-scale=1.0"
    c. title: to-do list
    d. link for css, called styles.css
4. Within the <body> tag, create a element with an ID, called todo-list.

Step Two: Design CSS Styles

1. Create a new file named styles.css.
2. Add basic CSS styles such as width = 300 px, padding = 10 px, and border = 1px solid #ccc, for the todo-list container.

Step Three: Add to-do input box and button

1. Add an <input> element and a <button> element in for entering to-do items and adding them to the list.

Step Four: Implement JavaScript Functionality

1. Create a new file named script.js.
2. Listen for one event, called button clicks, to add new to-do items to the list.
3. Retrieve input values from the user and dynamically create HTML elements to represent the to-do items.

MUST (30%)

Step One + Three (7%)

Step Two (8%)

Step Four (15%)

Homework 1-2 (at home)
Create a Simple Guess Number Game

Project Description

Create a simple web application that allows users to create a simple guess number game. The project will consist of multiple steps, each building on the previous one.

Homework Submissions Requirement:

1. Create a fold named "**student id_HW1**". (ex: R12725088_HW1)
2. Each step should have its own folder.
3. A PDF document named README.pdf to explain what you have done in each step. (Attaching screenshots to help explain is recommended.)
4. Compressed the fold "student id_HW1" into a single zip file and uploaded it to COOL.

   The folder should look like this:

```
R12725088_HW1
├─ README.PDF
├─ folder 1
        ├─ index.html
        ├─ A.css
        ├─ B.js
├─ folder 2
        ├─ index.html
        ├─ A.css
        ├─ B.js
├─ folder 3
        ├─ .....
```

Steps

Step One: Build HTML Structure

1. Create a new HTML file and name it *index.html*.
2. Set up basic HTML structure including <head> and <body> tags.
3. Within the <head> tag, create the following information
    a. charset="UTF-8"
    b. RWD: name="viewport" content="width=device-width, initial-scale=1.0"
    c. title: Guess Number
    d. link for css, called styles.css
4. Within the <body> tag, use <h1> tag for Guess Number.

Step Two: Add guess-input input box and button

1. Add an <input> element with type = number, id = guess-input, and placeholder = enter your guess, and a <button> element with id = submit-guess, for submitting the guessed number.

Step Three: Design CSS Styles

1. Create a new file named styles.css.
2. Add basic CSS styles such as h1 font size = 48 px, the width in guess-input = 500 px.

Step Four: Implement JavaScript Functionality

1. Create a new file named script.js.
2. Generate a random number between 1 and 100 as the answer to the game.
3. Listen for click events on the submit button, compare the guessed number with the answer, and provide feedback until the user guesses the right number.

MUST (70%)

Step One + Two (17%)

Step Three (18%)

Step Four (35%)