Web Dev 101

Week 1

Guidelines
- Course material repository https://github.com/olivierpas13/webdev101_material
- Course Psets repository https://github.com/olivierpas13/webdev101-PSets
- Each student should upload their classwork, knowledge inquiries and Psets to the corresponding week folder in a separate folder with their tag name.
- Tag names consist in the first letter of the first name concatenated with their lastname i.e. Opaspuel
- Psets deadline is always friday before 7:00 am.
- Course consists of 5 weeks, please consider that the difficulty of the course follows a exponential growth, i.e. it is expected that the difference in difficulty of week 3 and week 1 would be quite massive, don't get overwhelmed by it.
- Mondays consist of material revision and lectures about concepts, usually at the end of the class a small problem will be presented as *classwork* for students to complete as quick as possible.
- Wednesdays will be used as QA days helping students to complete the Psets, keep in mind this does not mean writing code for you or giving you the answer, you are welcome to present progress in your Pset though.
- Fridays will be used as grading and feedback days, comments and grades will be posted as commit messages inside your folder and added to the course spreadsheet, if you want to talk about your grades or comments, there will be a class link allowed for it.

Classwork

- Create an interactive webpage that allows users to input their name and select their favorite IoT device from a dropdown menu. When the form is submitted, display a personalized message (e.g., "Hello [Name], your favorite IoT device is [Device].") using JavaScript.

Key requirements:

- Use an HTML form with a text input for the name and a dropdown select for IoT devices.
- Use JavaScript to handle the form submission and dynamically display the result on the webpage **without reloading it.**
- Style the form using basic responsive design techniques (e.g., flexbox or grid) to ensure it looks good on mobile and desktop screens.

Knowledge Inquiries

1. Select one concept and gather more information about it, write a brief explanation of the information (we'll check it next class)

Weekly Problem Set 0

Problem 1

Create a web-based application that calculates the minimum number of coins (quarters, dimes, nickels, and pennies) required to make a given amount of change. The user will input an amount of money in dollars and cents, and the program should output the number of each type of coin needed, as well as the total number of coins.

Requirements:

1. **User Interface:**
    - A simple input field for the user to enter the amount of change (in dollars).
    - A button labeled "Calculate" to trigger the calculation.
    - A section to display the results, showing how many quarters, dimes, nickels, and pennies are needed, and the total number of coins.
2. **JavaScript Logic:**
    - Take the user's input (amount in dollars and cents) and convert it into cents (an integer).
    - Use a greedy algorithm to calculate the minimum number of coins needed:
        - **Quarters (25¢)**
        - **Dimes (10¢)**
        - **Nickels (5¢)**
        - **Pennies (1¢)**
    - Display the result back to the user, showing the breakdown of coins and the total number of coins.

Problem 2

Interactive To-Do List with Local Storage: Build a simple to-do list where users can add and remove tasks. Use JavaScript to store the tasks in the browser’s local storage so that the tasks persist even when the page is refreshed.

Requirements:

- Use a responsive layout that adjusts for both mobile and desktop views.
- Add input validation to ensure tasks are not empty.
- Use local storage to save tasks.

Things to consider:

- Use of CSS libraries or frameworks is acceptable
- No JS frameworks allowed (this includes Reactjs)