Assignment 4

Recommended readings:

- Lecture slides as starting literature
- MDN Links within slides for details, especially:
  - setInterval ○ clearInterval ○ replace

Note: All exercises must be solved with plain JavaScript and CSS not using additional libraries or frameworks.

Exercise 1 – Making an HTML Page Editable

Given the HTML document *editable.html* in folder *editable*, with sections and one h1 heading element as a child of each section. Write a JavaScript *editable.js* fulfilling the following requirements when referenced from the HTML document:

- Your code adds a form containing a select box, a text input, a textarea, and an initially disabled submit button after the last section.
- The select box allows the user to select any section of the document based on the heading of the section. After selection, the text input element shows the heading of the selected section, the textarea shows the innerHTML of the selected section, and the button is activated.
- When the button is clicked, the content of the selected section is updated with the data of the input element and the textarea. If the heading is changed in the textarea and in the input field, only the value of the input field applies. Be sure to correctly handle cases where the user removes the heading element in the HTML source. Also, be sure to update the content of the select box after a heading has been changed.

Do not change the HTML document except by adding the reference to your JavaScript file!

Exercise 2 – Image Gallery Part 2

Given the picture gallery from Assignment 3.5. Extend the gallery such that when a picture is clicked and shown in full window mode the user has controls for:

- "*play/pause*" implementing an automatic slideshow showing the pictures in an endless loop until *pause* or *exit* is clicked. The duration for showing each picture should be equal to the number of seconds specified in an input field next to the buttons (default value: 3 seconds).
- "*exit*" for stopping an eventually running slideshow and closing the full window mode.

Important: Your script must be working with any HTML content following the format of the HTML document of the lecture.

Exercise 3 – Puzzle Game in JavaScript

The folder *puzzle* contains the HTML document *puzzle.html*. The page shows a grid with 4 fields and a list of 4 images. The game's goal is to move each image of the list to the correct grid cell for solving the puzzle.

Moving images should be realized like in the lecture slides. If the user puts an image onto a field in the grid and clicks (the mouse position is the relevant position), then the image is removed and set as the background of the specified grid cell. If there is already another picture in the grid cell, then the previous picture is moved back to the list of pictures. When all pictures are in the correct position of the grid, the user wins, and a message is shown; otherwise, a message of wrong placement is shown.

Note: The "grid" is actually realized by the position of some divs. The solution is encoded in the data attribute *"data-result"* of each div. Your solution should not be restricted to the grid. It should be sufficient that each picture is moved to the div with the matching data-result attribute.

While you should apply the general principle of the lecture for moving the images to the grid, you can change the code such that the mouse is not positioned above the currently moved image during moving. This makes it easier to interact with the grid.