1 - Basic HTML/CSS and the DOM API

## Learning Outcomes

On completion of this lab you will have:

* Reviewed basic HTML and CSS language features and implemented code fragments for creating and styling common web page elements in the browser
* Created DOM elements using the just the native JS API

## Organisation

Please complete the exercises individually.

## Grading

This worksheet is worth up to 10% of your overall module grade.

You may work on this worksheet during lab 1 and lab 2 with instructor assistance. You may also be requested to demonstrate your submission to the lab instructor in order to receive credit.

## Submission

The deadline for submission is Sunday Oct 6, 2019 @23:59 through Brightspace. Details to follow.

The work and submission workflow is as follows:

* Create a sub-folder for each problem below. problem-1, problem-2, problem-3 etc.
* Put your solution for each problem in their respective folders.
* When you are finished developing your worksheet solution, compress and zip all problems into one zip file. Name this ***<student-id>***-lab-1.zip
* ***<student-id>*** is something like C12345678
* Upload to Labs / Lab 1 / Lab 1 – Upload.

## Resources

You are free to research whatever you need to solve the problems in this lab. Some recommended resources include:

* <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML>
* <https://developer.mozilla.org/en-US/docs/Web/CSS>
* <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
* <https://www.codecademy.com/learn/javascript>
* <https://www.wolframalpha.com/input/?i=rule+60>
* <https://www.youtube.com/watch?v=FQsBmnZvBdc>

## Problem Sets

Credit will be given for well-organised and maintainable code, so keep it [DRY](http://deviq.com/don-t-repeat-yourself/)

|  |  |  |
| --- | --- | --- |
| **1** | You are required to create a visual model that demonstrates list insertion. The model must consist of three elements:    List display section: The section on the page where the list will be displayed.    Text field: A field to provide input for inserting a new element into the list.    Button (named "Insert"): Clicking this button inserts a new element at the end of the existing list, the new element being the value provided in the text field.      Requirements:    No element should be inserted if the text field is empty when the user clicks the button.  Every third element in the list must be displayed in red and the remaining elements in black.  Each of the list elements should follow the format <li style="">text\_to\_be\_inserted</li>. | 10 marks |
| 2 | Create a flexible grid with 1 row and 4 columns. The width of each column is 25% of the window size. This percentage width must be maintained even if the page is resized. Each cell of the grid can contain another 1x4 flexible grid. The border of the grid must be 1px black.    Perform the following operations based on the value of window size:    If the window size is less than 720px, then the 1x4 flexible grid becomes a 2x2 grid. That is, the 3rd and 4th columns slide down onto the 2nd row.  If the window size is less than 360px, then the 1x4 flexible grid becomes 4x1 grid. Each column slides under the one before it. The 2nd column slides under the 1st, the 3rd slides under the 2nd, and the 4th slides under the 3rd.  **NO BOOTSTRAP!** | 35 marks |
| 3 | Create a simple note-taking app. A user should be able to:    Add a note  Edit a note  Delete a note  Also:    Each note should be in a colored rectangular box.  Box colors can be selected from a fixed list of colors. | 55 marks |

**Please work on this yourself and do not be tempted to plagiarise a solution, even in part, from a colleague or from the Internet.**