Lab 1

In this lab we will continue practicing Javascript in the browser. You will download an HTML file, then add Javascript code in its <script> tag. When you open the HTML file in the browser, you will be able to see how you've done.

The file to use will be this[1]. Download it and save it somewhere on your computer, then open it in both the web browser and in SublimeText.

You will find in that file a <script> section. You will add your Javascript code there to answer the following questions. Every time you save the file and reload the page on the browser, you can watch your progress.

The questions you need to answer follow. When you are satisfied with your answers, save the HTML and email it to me.

- This assignment builds the logic for a tic-tac-toe game. You will implement a board and various methods related to it.
- The board should consist of a single array of length 9, whose contents represent each square, counting horizontally first, and from left to right then top to bottom.
- While there are other ways to implement the board, you MUST do it this way.
- A value of null in an entry indicates that nothing has been placed there yet.
- Otherwise a string indicates a placement of a mark in that square. The system will be calling your functions and giving them "o" and "x" as the relevant strings, but your code should NOT use that fact.
- Players are identified by the same strings that are used for their markers (i.e. "o" and "x" but your code should NOT use that fact).

1. Implement the checkLine function. This function takes as input three values, which may be strings or null, and must return one of two things. If all three values are equal, then it must return that common value. Otherwise it must return null. The idea is that you can use this function on a triple of squares to see if they make one of the players a winner. The function will end up returning that winner's string if there is one, and null if that triple does not produce a winner.
2. Implement the initBoard function. This function is meant to initialize the board variable, which as mentioned earlier must be an array of 9 values, all initialized to null.3. Implement the set function. This takes as arguments the row and column and a value, and must set the entry in the board that corresponds to the square specified by the row and column to this value. Rows and columns are indexed from 0. For example the square on row 2 (the third row) and column 1 (the second column) should end up setting the next to last element of the array. Your function should throw an error if there is already a value set for that square.
3. Implement the get function. It given a row and column, and return the value of the board for the corresponding square.

[1] https://github.com/skiadas/WebAppsCourse/blob/gh-pages/labs/lab1.html

4. Implement the winner function. This is the bulk of the work you have to do for this assignment. This function is supposed to examine the board and determine if there is a winner given the current values of the board. If there is no winner, it must return null. If there is a winner, then it must return an object with two properties: A "player" property which is the string corresponding to the winner (the same string as returned from your checkLine function), and an "entries" property which must be the array of indices corresponding to the winning triple, in increasing order. For instance if the winning triple is the first column, the value of entries must be [0, 3, 6].