# Assignment – Random Color Grid (JavaScript loops, functions, conditional statements)

This assignment is broken into two parts. Start by solving the first problem.

**THE FIRST PROBLEM:** Write a program that generates a 10x10 HTML table filled with randomly selected numbers between 1 and 100 (inclusive). If the random number is greater than 50, the number should be made bold using the HTML <b> tag.

Use parameters so that this program can be easily modified to generate any Row x Column grid.

For example: 3x2, or 7x9, or 5x5.

This start file already contains a function called getRandomInt that will return a random integer between 1 and x. For example: getRandomInt(6) will return a random integer between 1 and 6. You must use this function that has been provided to you.

Here’s some sample output:

|  |  |
| --- | --- |
|  |  **This is a 10 x 10 table. Your output should be similar to this!** |
|  | **This is a 1 x 5 table (1 row, 5 columns)** |
|  | **This is a 3 x 1 table (3 rows, 1 column)** |

**BE CAREFUL:** Don’t get “rows” and “columns” confused!

**REQUIREMENTS:** Your solution to the first problem should use three functions: buildGrid(), buildCell(), and getRandomInt(). The getRandomInt() function has already been provided for you. You must use it.

## Getting started (Together as a class)

1. Retrieve assignment13\_randomgrid.zip provided by your instructor.
2. Extract the code into your mis2402workspace and open the index.html file in Visual Studio Code.
3. Use the web developer tools in Chrome to test your getRandomInt() function. Sample screenshot follows:



1. Complete the buildCell() function. Use a conditional statement inside the buildCell() function to test if the input parameter x is greater than 50. If it is, wrap x in an HTML <b> tag. Otherwise, just return x.



1. Begin the buildGrid() function. Notice the parameters provided numrows and numcols. Some suggestions follow:
	* This function will return a result that starts with the “<table>” tag. That would be a good initial value for your output variable.
	* You’ll need two loops to solve this problem. One loop will generate all the rows. Write a for loop that generates rows 1 to numrows. In each iteration of the loop add to your result string (using concatenation) the start “<tr>” tag and the end “</tr>” tag.
	* Now, ***inside of*** the first loop, write your second loop. This second loop should generate 1 to numcells. In each iteration of the loop add to your result string (using concatenation) the start “<td>” tag, the loop counter variable, and the end “</td>” tag.
	* Be sure to test your code by both looking at the output on the screen, as well as using the Chrome web console to test your function.
	* Make sure your output ends with a “</table>” tag.
	* See the next page for some sample output.



On your own

1. Complete the buildGrid() function. Use both the getRandomInt() function and the buildCell() functions to complete the task.

You should be calling those functions from the innermost loop in buildGrid().

Some sample output follows.

1. Oh no! It looks like there's some sort of bug in the getRandomInt function. You better fix that.
2. Test your work. Be sure to test both the output that is seen in the browser, as well as checking each individual function in the Chrome web console.
3. When everything is working ok, move on to part 2.

**THE SECOND PROBLEM:** We want to add some random colors to our solution.

Here’s some sample output:



1. Complete the function called getRandomColor. This function should randomly generate a number between 1 and 5, depending on what number is generated, return one of the following file strings: ‘red’,’blue’,’green’,’purple’, ‘yellow’.
2. Use the Chrome Debug console to ensure that getRandomColor works as expected.



1. Now revisit your code in the buildGrid function. Add to / edit your code so that the random color is used as part of a style attribute in each <td> tag.

For example:
<td style="background-color:purple">
2. Test your work again. The random numbers and the random colors of the table should change every time you press the Go button.
3. Publish your work. Be sure that you can find your work on the class server by typing in its URL in the browser. Test your work \*again\* on the class server.

For example:
http://misdemo.temple.edu/tux99999/assignment13\_randomgrid

How will this assignment be graded?

This assignment will be evaluated by an automated process.

* If your work is not found at the expected location on misdemo, you will get a score of **zero**.
* If your work generates **all output** correctly, you will get a score of 100%.
* If your work generates **almost all output** correctly (**only one** bad output), you will get a score of 80%
* If your work generates **some output** correctly (some right output, some wrong output), you will get a score of 60%
* If your work generates **only one output** correctly, you will get a score of 40%
* If your work does not generate any correct output, you will get a score of **zero**.