Assignment 2--Generate the grid for your project game

Due: 11:59pm February 2, 2015

Write JavaScript code to generate table HTML for a grid that is suitable for your project game. Instead of typing HTML for the grid into a file using an editor, you will write JavaScript code that generates the table HTML by creating a string that has all of the HTML tags in it (or by using the `createElement` function).

Writing code to generate HTML is a very important part of web applications and can be useful in other kinds of applications as well. Generating HTML is faster, less tedious, and less error-prone than typing in HTML directly. It's also more flexible, because a single function can be used to generate tables of any size. However, the most important reason for writing code to generate HTML is to allow a web application to respond to user inputs and make dynamic HTML documents. For example, in a word search puzzle or Minesweeper game, the user could choose the number of columns and rows in the grid, and then JavaScript code could generate a table of the specified size. In later assignments, you will be writing a lot of JavaScript code that generates HTML, so it's important to make sure you spend time now to learn how to do it well.

All of the table for the grid should be generated by JavaScript. You can generate HTML strings and then add them to your document using **`getElementById()`** and the **`innerHTML`** property, or you can use **`createElement()`** to make the necessary elements. The rest of this assignment page is focused on using `getElementById()` and `innerHTML`.

You can see an example of using `getElementById` and `innerHTML` here: genSample.html with detailed notes about the example here. The basic idea is that you put a div with an id attribute in your document and use `getElementById` to get a pointer to that div. You can read about `getElementById` in Chapter 2 of *Modern JavaScript*, starting on page 47, and in Chapter 9 on page 340.

Once you have a pointer to the div, you can set the `innerHTML` to a string, and the browser will add that HTML to the div just as it would if the string had been typed into the HTML file. The Lesson Notes for the lesson on scripting documents has another example of using `getElementById` and `innerHTML`: Document Object Model Notes

Your JavaScript code should use *nested loops* where only one table cell is generated each time the body of the inner loop is executed. It should also use variables (or parameters) for the number of rows and the number of columns in the grid so that the size of the grid can easily be changed. You should be able to change the size of the grid by changing only the values of those two variables (that specify the numbers of rows and the number of columns). Even if your game has a standard grid size, like an 8x8 chess board, you should still write your code so that it can easily be used to generate other grid sizes.

Notes:
--Even though all of the table HTML for the grid should be generated from JavaScript, other HTML on your game page (like the page title) can be typed directly into the HTML file.
--Be careful with quotes. Sometimes you will need to use JavaScript strings that have HTML quoted strings inside them. In that case, you can use double quotes for the JavaScript string and single quotes in the HTML (or vice versa), like this:

```
    var pTag = "<p style='font-weight: bold;'>";
```

Or, you can escape the HTML quotes with back slashes, like this:

```
var pTag = "<p style=\"font-weight: bold;\">";
```

Functions

Encapsulate your grid-generation code in a function, so that the table can be generated by calling a single function. Your function should have parameters to specify the size of the grid.

If you are using `innerHTML`, I recommend that your function returns an HTML string rather than modifying the document directly. That way the same function can be used to create grids in more than one place in the document. For example, in Battleship you could use the same function to make the grid for the player's ships and the grid for the computer's ships.

Make a pattern using CSS `class` attributes

After your JavaScript code is correctly generating the table HTML for your game's grid, modify it to also generate `class` attributes to make a checkerboard pattern in the grid. If there is another pattern that works better for your project, you can generate a different pattern. For example, in Sudoku you could make the lines between 3x3 boxes thicker than the lines between individual boxes in the grid. Just make sure that the grid squares don't all look the same and your code generates different `class` attribute values for different cells based on their coordinates in the grid. If you use `createElement()` to make your grid, then your JS code should assign different values to the `class` properties of the table cells to make the pattern. Make sure you use CSS class attribute values to make a pattern (like a checkerboard pattern) even if the grid for your game doesn't usually have a pattern.

Patterns made with images or text content do not satisfy this requirement. Also, it's possible to make a checkerboard pattern without generating different class attribute values for different cells (using the `nth-child` property) but that does not satisfy this requirement because it doesn't involve JS code assigning different class attribute values to different cells of the table.

Debugging

Debugging JavaScript is quite different from debugging programs written in some other programming languages, because of the way that JavaScript is used in a web browser. It's *very* important to learn how to debug JavaScript now, when you're writing a relatively simple program and it's easier to see what's going on.

Be sure you have read the Notes for Lesson 3, including reading the debugging tutorial and watching the Firebug video. Then, **be sure to install and use Firebug or a comparable debugging tool**, with your program for this assignment, including setting breakpoints and stepping through code. Firebug is not the only method of debugging JavaScript, but for some bugs it is the easiest and best way to find bugs, so learn how to use it now.
NOTE: If you prefer, you can use the Developer Tools in Chrome, rather than Firebug, or the tools in the Develop menu of Safari. The Developer Tools come with Chrome and don't have to be installed as as an add-on like Firebug does. In Chrome on Windows, click on the wrench icon to open a menu, then select Tools->Developer Tools to open the developer tools.

Don't use frameworks on the assignments

Frameworks, like jQuery or YUI, are important in web programming, but in the assignments for this class we are focusing on basic JavaScript. Please don't use jQuery or another JavaScript framework for assignments.

Turn in

Put your grid HTML file (including JavaScript) and your CSS file in a zip file. Turn in the zip file as specified on the Assignments page.

Points

30	JavaScript code uses nested loops to generate table HTML for the grid, generating one table cell each time through the inner loop.
20	Code to generate table HTML is encapsulated in a function with parameters for the size of the grid.
20	JavaScript code uses `getElementById` and `innerHTML` to add the table HTML to the document (or uses createElement)
30	JavaScript code generates CSS `class` attributes to make a checkerboard pattern (or other pattern) in the table
--	Practiced using Firebug (or Chrome Developer Tools) and other JavaScript debugging techniques. There are no assigned points for this part of the assignment, but it's very important.
100	TOTAL