JS Lab 2:

Due Thurs, May 4

Part 0: Study for the Celebration of Knowledge. Be aware that you do not memorize programming. It is important to be able to understand programming. It's like reading – yes you can memorize the words and the syntax, but the words and syntax can be put together in so many ways that you must understand the text in order to follow anything that is written.

Part 1:

1. (20 pts) Create a web page with a paragraph on it. Then create a javascript in the head section.
   a. In the head section, create a function that changes the style of the paragraph's font. It should change the font color, the font weight, the font size, and any other style element you want to change.
   b. Write a second function that changes the font style back to plain black, normal weight, white background, normal size, etc.
   c. Using setTimeout, make the first function call the second function after 2000 msec.
   d. Using setTimeout make the second function call the first function after 2000 msec.
   e. Within the paragraph itself, add a function call to the first function when the paragraph is clicked on.
2. (30) Create a web page with an image, a paragraph and a button on it. Now write a javascript in the head section. In the javascript, add a function. The function should generate 2 different random numbers between 0 and 800. In the function set the positioning of the image to be absolute, and then set the left position of the image to be the first random number, and the top position of the image to be the second random number.
   a. Inside the function, use setTimeout to call the function again relatively quickly (maybe every second or faster).
   b. Have the button on your web page call this function so that it starts moving the image around randomly. It should call the function when you click on the button.
   c. Outside of the function, but within the javascript in the head section, create a variable that will be used for counting. Initialize it to 0
   d. Create a second function (below the counting variable in the javaScript). This second function will first increase the counting variable by 1. It will then change the paragraph to be the new counting variable. (Note that this function should only have 2 lines inside it – it doesn't use setTimeout.
   e. Inside the image, add a call to the second function every time you click on it.

   Now play. What should happen when you click the button is that the image should start bopping all over the screen. Every time you are able to click on the moving image, the count should go up by 1 inside the paragraph on your web page.

3. Your final project: (optional for this week's lab, but must be done for the final project):
    a. Continue to refine your html and css until you have the look you want.
    b. Start your javaScript.
    For this project you will need a lot of arrays:
        1. Create an array of the ids of your bottom monsters (the monster images at the bottom of your web page, each with its own id.
        2. Create an array of random numbers. This array should be exactly the same length as your first array. The random numbers should be initialized to something between 150 and 350 (although you may wish to play with this range later on) These will be the maximum top position your monster will get to.
        3. Create a third array for the Monster's current position from the top (so a third array of integers the same length as your first array, initializing each value in the array to something like 800, although again you may wish to play with that number later.
        4. Create yet another array the same length as the monster array. This will control the direction the monster is moving in. Each value should be initialized to -10 (because we want these monsters to move towards the top).
        5. Final bottom monster array: create one last array the exact same length as the monster array. Each value in the array should be set to the corresponding monster's position from the left on the screen (so look at your css and see how far over you positioned it from the left.)

        6. Now repeat the entire thing with a second set of 5 arrays for your monster's at the top. Difference? The random numbers should be closer to between 50 and 250, the top position array should start at maybe 5 (closer to the top), and the direction should be initialized to 10 (so they're going in the opposite direction from your bottom monsters).

        7. Create a function. Inside your function create a variable x. Set the variable x to be a number between 0 and the length of your bottom monster array. X will be used as the index into all of the arrays associated with the bottom monsters. So, for instance, if you want to access the monster's id, you'll use the monsteridarray[x] (or whatever you happened to call that array). If you want to access the monster's current position from the top, you'll use the monstercurrtoparr[x] (or whatever you happened to call that array).

            You are going to change the position of your monster using the arrays you created above:

            a. First, add the monster direction array at index x's number to the monster top position array (so it will actually go down by 10).
            b. Next use document.getElementById and the monster id array to set monster x's position to be absolute

c. Then use document.getElementById and the monster id array at x to change the position of your bottom monster to its new top position in the monster position array at index x.
d. At the bottom of the function, use setTimeOut to call this function every 100 msec (you can play with this number until you get the speed you want later).
e. For testing purposes, make the start button on your web page call this function and test to make sure that one of your monster x moves upward. (we'll change what the start button calls later).
f. Add a condition: if the monster x's top position gets to be less than the monster's max top position, change the monster x's direction from -10 to 10.
g. Test again, and make sure your monster x goes up and then down.
h. Now add another if condition: if the monster x's top position gets to be greater than a particular bottom number (maybe 800, although you can play with this), change the monster x's direction from 10 to -10. In addition, since you want the monster to move up to a different maximum top position each time, change monster x's top position to a new random number between 150 and 350.
i. Test again. Now your monster should be moving up and down, each time moving up to a different height. Try changing the value of x and seeing if different monster's move up to different heights.

That's enough for now. If you get this working, and you want to repeat the process (in reverse, because the top monster's are moving downwards to a random number), feel free to write a second function for your top monsters.