JS Lab 3

Due Thurs, Apr 26

Note: If you're having problems, make sure you review and understand the tutorials and the powerpoints (on the class web site)

Hints for problems encountered:

a. If nothing is working, there's probably a typo you'll have to find.
   i. Look for capital letters where there should be a small one, and vice versa
   ii. Look for missing " " (if you open it, you must close it)
   iii. Same with { and }
   iv. Same with (and) – for every one of the first, you must have one of the second.

b. Finally, make sure your html is valid (again, if you opened a tag, you should close it)

If you're still having problems, use comments to isolate exactly where the problem is. Any code you put between / and */ (e.g., see below) will be ignored by the browser.*

```
/*

  document.getElementByID('pic1).source = "cat.jpg"; <- this line is commented out,

*/
```

1. (6pts) Create a web page (or modify the existing web page) with an image on it of some picture of your choice (we'll call this picture1.jpg). Write a function that uses getElementById to change the picture in that image to something new (picture2.jpg). Write a second function that changes the picture in that image back to the first picture (picture1.jpg). Now modify the image on the page so that when you run your mouse over it, the first function is called, and when you run your mouse off of it, the second function is called.
2. (5 pts) For parts a and b, you're basically recreating what we did in class when we went through an array of pictures, then looped back to the beginning. We also had the ability to add pictures to the array, and the functions would automatically include those new pictures because we used .length
   a. Create a web page with an image on it, a paragraph on it, and a button. Now create a script (in your head section) that contains an array of pictures and a variable initialized to -1. As we saw in class, it should also contain a function that will first increase the variable by 1, then check to see if the variable is longer than the number of elements in the array, and, if so, resets the variable to 0. The function then displays the picture in the array at that variable number. So it will loop through the pictures in order, and when it gets to the end of the array, it should loop back to the beginning. Now make the button on your web page call that function. Make sure that if you add pictures (using the function you'll write in part b) to your array, this function will work regardless of how many pictures you add.

   (5 pts) b. Now add another button and another function. This function should allow you to add pictures to your array. The second button on your web page should call this second function. (again, this is pretty much what we went over in class).

(10 pts) c. Now, inside your script, but above your functions (either above or below the first array of pictures), create a second array. The array should hold text describing (in order) each of the pictures in the array of pictures you created. (Remember to put quotes around the text, so the array will look something like this:

textarray = new Array()

textarray[0] = “description of picture 0 goes here”

…

Modify the function you wrote in part a so that it also changes the paragraph on the web page’s text to what is in your array of text at the variable

(10pts) d. Finally, modify function part b so that when you add a picture to your picture array, you must also add text describing the picture to your text array (you’ll use two prompts for this: the first will get the new picture, and the second will get the new text).

3. (25 pts) Write a web page (or modify the existing web page) with an image and a corresponding paragraph. Create a button below the image and paragraph, and then a second paragraph below the button.
Now have the button call a function.
The function should contain 2 equal-length arrays: An array of pictures of faces displaying emotions, and a corresponding array of sentences describing the emotion. Then the function should generate 2 different random numbers. The first should cause the picture at that number in the array of pictures to be displayed in the image on the page. The second random number should cause the sentence in the array of descriptions to be displayed in the images corresponding paragraph on the page.
A confirm box should then ask the user, “Do these go together?”
If the user clicks “ok” and the two random numbers match, you should write to the third paragraph, “Good job! You are correct.” If the user clicks “ok” and the two random numbers don’t match, you should write out, “Sorry, that is not correct.”
If the user clicks “cancel” and the two random numbers match, you should write, “Sorry, you’re wrong. These two do match”, and if the user clicks “cancel” and the two random numbers don’t’ match, you should write to the paragraph, “Correct, these two don’t belong together.”
(Note: this could be a training exercise for children with autism. If you want to use this methodology to create a training tool, for, say, a foreign language (with pictures of something and a description in a different language), or anything else, you can. I don’t care what the array of pictures and the array of sentences contain, just that you create the training tool).

Note: *If, for some reason this doesn’t work in Chrome, try running it in Firefox. Chrome is funny about when it actually executes statements (there’s a way around this, but in the meantime, just run it in Firefox.*

4. (25 pts) Write a web page (or modify the existing web page) with an image on it. Position the image absolutely (you can use in-line css style if you like), with the left position being at 0 px. Add a button somewhere lower in the page that calls a javascript function using onclick.

Now add a javaScript. The javaScript should have a count variable, just like we had for going through an array sequentially. It should also have a function. Inside the function, you should first increase the count not by 1 (as we did for the arrays), but by 10. You should then use document.getElementById's style properties to change the position of the left to be count pixels over. (You'll have to use say:

```
document.getElementById("imgid").style.left=count+"px";
```

where "imgid" should be replaced by the id of the paragraph the image is in on the page (or the image itself, if you've styled the image instead of a paragraph surrounding the image).

Now when you click on the button, the image should move to the left by 10 pixels each time.

5. (15 pts) Start on your final project. Create a folder called FinalProject. Inside the folder create another folder called Images. This is where all the images for your final project will go.
   Inside the FinalProject folder, create a finalproject web page.
   For your final project, (see JS Lab 1) you will need to set up a background scene, with images placed strategically throughout the scene.
   a. Pick a theme for your project (e.g., Under Water, Outer Space, The Dessert, a Winter Scene, a Marsh, a Night Scene, whatever you like).
   b. Pick a background image. The image should fit with your theme.
   c. Create a web page with a div tag. Give the div the background image you chose.
   d. Now download appropriate objects to place within your background scene. Add the object inside the div in your web page. You should have at least 4 different objects (I think I ended up with closer to 8 or 9, and placed some of the pictures on the web page more than once. I used css to position the entire div relatively, and then used css to position each of the images inside the div using absolute positioning. Because they were inside the div, they were positioned absolutely based on the edge of the div.

Note that the web page itself should be inside the Final Project and all the images should be in the Images folder.

This is, in essence, a "rough draft" of what your final project will look like. But you should start thinking about it and start getting it set up.

Upload the FinalProject folder to the server, along with your JSLab3 folder.