JS Lab 1:

(Due Thurs, April 27)

For this lab, you may work with a partner, or you may work alone. If you choose a partner, this will be your partner for the final project. If you choose to do it with a partner, make sure both names are on the lab. Only one person needs to turn in the lab, and the other partner should say in sakai who they worked with. Regardless of your partner's participation, you are responsible for completing the lab and turning it in on time.

Make sure you review and understand the material covered in the powerpoints in class.

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)

Problems (due Thursday at Midnight):

Make a folder named JSLab1, and place all of the web pages for the problems in the folder. Upload the folder with the web pages to the server, and submit the URL of the folder on Sakai. Only one partner should submit, but if one partner fails to submit, both partners are responsible.

Part 1: (15 pts, 3 pts each)

1. Write a web page. Inside the web page, make a head tag, with the content, "JS Lab 1 Question Answers.
   Under that, create an ordered list.
   Inside the first list item,explain the difference between document.write and document.getElementById.
2. In the next list item, explain the difference between a variable and an array
3. In the next list item, explain why we need a variable before a prompt command
4. In the next list item, write in JavaScript how we'd generate a random number between 50 and 350 (not including 350).
5. If you generate 4 different random numbers, at the very least, how many variables should you have in your javaScript?

Part 2:

(Note: On your web page you can have something like,

```
<img src = "cat.jpg" id = "img1" width = "300" height = "400" alt = "cute cat">
```

When I refer to the image on your web page, I'm referring to the <img…> tag, whereas when I refer to the picture, I'm talking about what the src is set to, or "cat.jpg")

1. (5 pts)Create a web page with a paragraph with an id of “p1” that says, “Secret Message”. The paragraph should have a call to a function using onMouseOver. The function (placed in a script in the head section of your web page) should change the paragraph p1’s innerHTML to “Computer Science is awesome!” (You may choose another phrase). When you move your mouse off of the paragraph, the innerHTML should change back to, “Secret Message”
2. (5 pts) 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.
3. (10 pts) Create a web page (or modify the existing web page) with a header, some paragraphs, and an image on it. Add a button that says, “Click here for better accessibility”. Make the button call a function. The function should change the style of the page to adapt to the needs of people with low vision (e.g., bold font, large font, arial font, high contrast, make the image(s) on the page larger, and make the image have a longer alt tag).
4. (8 pts) a. Create a web page (or modify the existing 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 to your array, this function will work regardless of how many pictures you add.

   (8 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).

   (5 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”

   …

   Now modify the function you wrote in 4a so that it also changes the paragraph on the web page’s text to what is in your array of text at the variable

   (8pts) d. Finally, modify function 4b 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).

5. (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.*

6. (20 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).

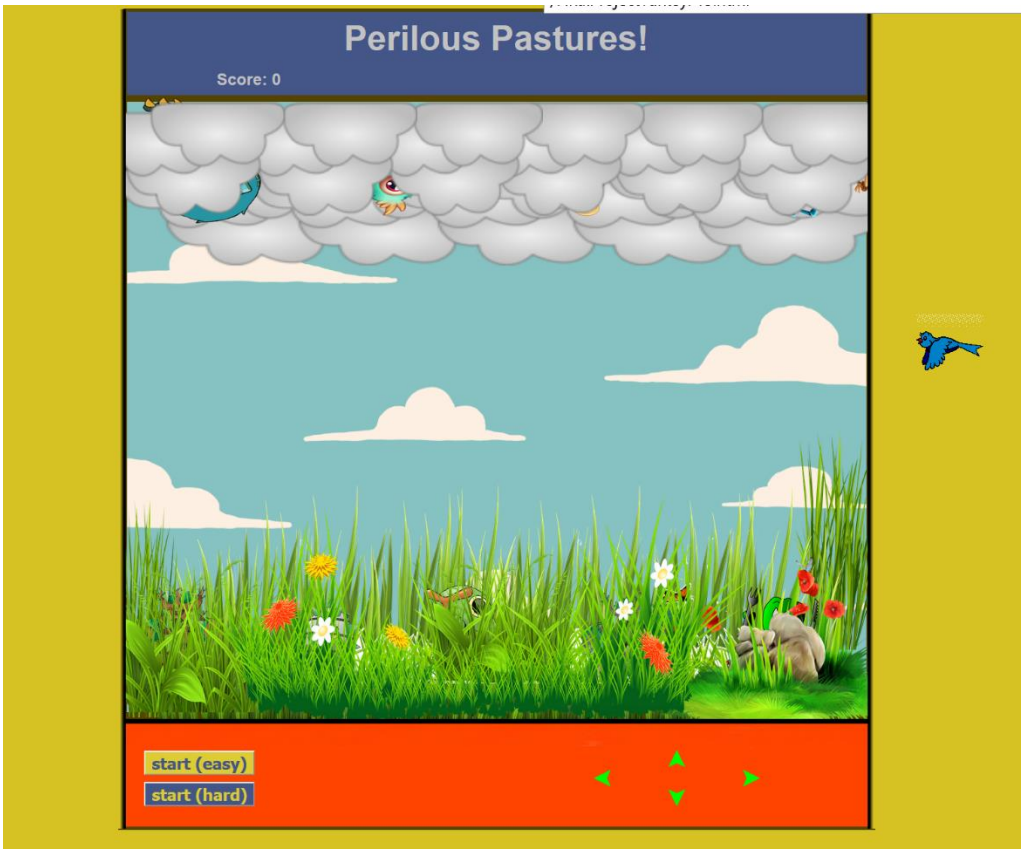
   Now when you click on the button, the image should move to the left by 10 pixels each time.

7. (6 pts) Add comments /*…*/ throughout your code describing what you are doing (notes to yourself and to the TA so your code is easier to read. These notes should explain what you were trying to do throughout your code).

Final Project:

NOTE: The final project will be due on the last day of class. Different parts of the lab will be due before that (as part of the remaining labs).

For your final project you will be creating a game, in which a bird hero(or animal of your choice) must move from one end of the screen to the other end while scary creatures (again, monsters are of your choice). If the bird hero successfully makes it from one end of the screen to the other, the score increases by 10 points. If, however, the bird gets caught by a monster, then the score goes down by 10 points and the bird returns to the right side of the screen to start over. Below is a screenshot of my game before it starts. Please feel free to choose your own images, scary creatures, and hero as you please. For instance, you could have a spaceship trying to avoid shooting stars and planets (and maybe alien spaceships), or maybe a human trying to escape from zombies and ghosts in a graveyard. I just went with scary creatures in nature.



Part A:

HTML and CSS:

For my project I set up the html so that I had 2 rows of clouds, positioned absolutely, with the z index of the first row of clouds set to 0 and the second row set to 20. I then added 4 monster images, each positioned absolutely, and with a z-index of 10. The goal was to have the monsters appear to be between the back row of clouds and the

front row of clouds. If you look carefully you can see the monsters peaking out among the clouds. Again, you may pick the creatures/images you like, but you will need to have 3 layers of images and approximately 4 monsters in the middle layer.

I also had a row of grass images positioned absolutely with a z index of 20, a row of 5 woodland monster images positioned absolutely with a z index of 10, and a row of grass images with a z index of 0. Again, the idea was that the woodland monsters would appear to be between the two layers of grass in the foreground and background. You may choose the images of your choice, but again, you want to have 3 layers and approximately 5 images in the middle layer

At the top of my html page I had a header saying "Perilous Pastures" and a paragraph, that I initialized with text saying, Score: 0. (You choose the name of the game, and the creature to whom you are assigning a score – it should not be any of the creatures in the middle layers, above.

At the bottom of my html page, I had a table with 4 arrows, one pointing left, one pointing right, one pointing up, and one pointing down. These arrows will be used to call a function that will move the bird up, down, left, and right.

Below that, I had a button for starting the game (Note: I actually have 2 buttons, one for a version of the game that may make it easier to see what is happening, and one that requires almost no extra work but is harder to follow visually).

And finally, I added an image of a bird (my hero), positioned absolutely. I positioned my bird over to the right of the board. You may pick your creature, and you may pick where you want the creature to start. Mine is an animated gif, so the bird appears to fly.

The monsters (cloud monsters and woodland monsters) and the bird (or your equivalent), and the paragraph all must have unique ids.

I used a table with 3 rows and 0 columns to make the top and bottom of the game line up nicely.

NOTE: I used an animated gif of moving clouds for the background and animated gifs for the grass as well. It made it look more interesting.

USE CSS to make the game board appear as you wish.

To Turn In:

Upload your folder called JSLab1, containing:

- web page(s) for problems 1-10 and the corresponding images
- A first draft of your Final Project Html and CSS

Submit the URL via sakai.