Overview:

In this lab, you shall create a JavaScript animation page named 03_js_intro/index.html that uses object composition of at least two animation objects inside of a background object. Your animation objects will respond to user events. This is a lab about learning JavaScript, so you MAY NOT use jQuery. As always, you cannot submit code that is overly similar to any examples presented in class or on my web site.

Basic Requirements:

1. The first line of your JavaScript code shall have the "use strict" directive as its first line of code (see below). By having this directive, you won't be able inadvertently declare a new variable by misspelling a variable name. It seems odd, but if you forget to use the keyword "var" when first using a variable, a global variable is created, regardless of where the variable was first used. So it is very important that you always use the keyword "var" when declaring variables/objects. Note that you will actually have the double quotes in your JavaScript code when you "use strict". Test that you have this right by trying to use a variable that was never declared with the "var" keyword.

   "use strict";

2. Your code shall use the object oriented JavaScript techniques covered in class, meaning that there will be very few globally declared objects and very few globally defined functions.
   - Your code shall include a function called makeBackground() that is passed an id to a div that will hold your background image. Your makeBackground function can “hard code” some of its attributes or have them passed in as input parameters, but makeBackground shall set whatever CSS properties are needed for JavaScript calculations (and these same CSS properties shall NOT be set within the internal CSS style sheet). Any other CSS properties for the background div (e.g., that are not needed by JavaScript) can be set in the internal CSS style sheet.

     // Main program
     var bgObj = makeBackground("myBgDiv");

   - makeBackground( ) shall have a method like makeAnimation( ) that creates a new animation object as a property of the background object (object composition). The main program will invoke the makeAnimation method at least twice like this (but of course, use object names that are representative within the context of what you are coding). With input parameters to the makeAnimation method, you can make the objects have different properties and/or behave differently.

     var catObj = bgObj.makeAnimation(...);
     var dogObj = bgObj.makeAnimation(...);

   - This makeAnimation method shall:
     - Create the animation object by using document.createElement( ) – not by getting a reference to a div that already exists in the HTML.
     - Implement animation of the sprite set image variety (poses that are automatically advanced as time progresses), using an array (declared within makeAnimation) of backgroundPosition values to select out the poses from the sprite set image. You can choose to have one of the animation objects implement the sprite set image animation and ha
• Associate an event with the animation object (like click or onmouseenter or onmouseleave). For a list of DOM events, visit https://www.w3schools.com/jsref/dom_obj_event.asp

**Originality and Other Requirements:**

1. **Topic.** You can use virtually any idea you have for background object with at least two visual animation objects on it, but try to come up with something that is related to your web site topic. Remember that each of the animation objects must respond to a user event and use sprite set image type animation (where poses automatically advance).

2. **Originality.** You must come up with something original. Points will be deducted if you submit work that is overly similar to any sample code provided from my web site. Also, we will be able to tell if you have just copy/pasted an example from the internet because the code will likely be overly complicated and not meet the object oriented specifications of this lab. Finally, your code shall not be overly similar to any other student’s code. If we find this, we will have to determine who wrote the code and who copied and this is no fun for anyone, so please do not do this.

3. **Professionalism.** To get full credit, you have to create a component that is professional and visually appealing. You have had two design labs so you should be in a position to come up with something that looks good.

4. **Look and Feel.** You can use the look and feel either of your initial layout lab or your advanced layout lab (or you can create a new or modified look and feel that meets the requirements of either lab 1 or lab2).

5. **Blog.** For each lab, you will add a blog to your labs page and this lab is no exception. In your blog, link to the work you did ("03_js_intro/index.html") and describe your experience doing this lab. What aspects were easy? Which were hard? What were the most important things you learned in this lab? For this particular lab, you shall **also include instructions for us to run your animation.**

6. **Good Programming Style.** At the top of the 3344 labs page, there is an entry entitled "Requirements For All Labs". Make sure you adhere to the "good programming and design practices" listed in there.

**Grading and Submission:**

1. After completing all the requirements, test locally (and syntax check), then publish and test what you published.

2. Then submit into blackboard a zip file of your whole web site (including all the work you have done in all the labs so far). Make sure to include all the necessary files and folders in your zip file.
Here is How We Will Evaluate Your Lab:

1. We will visit your published web site, link to your labs page then read the blog for this lab which should explain how we are to test your animation page (03_js_intro/index.html).

2. After linking to your animation page, we should see a background image with two visual objects on it. We will invoke events on each of the objects (as per your instructions) and see if we get the results your instructions indicated we should get.

3. We will view source on your HTML page and check for the program style requirements (listed in the "Requirements For All Labs" section at the top of the 3344 labs page). We will also check that you have followed the basic requirements listed for this lab:
   - Your first line of js code is the "use strict;" directive (so you can't inadvertently create a new variable by misspelling).
   - Your js code has a function like makeBackground(), e.g.,
     
     ```javascript
     var bgObj = makeBackground("myBgDiv");
     ```
   - The object returned by makeBackground has one or more makeAnimation methods, e.g.,
     ```javascript
     var catObj = bgObj.makeCat();
     var dogObj = bgObj.makeDog();
     ```
   - The makeAnimation type functions (like makeCat or makeDog) have used document.createElement() and not document.getElementById() to create the animation objects.
   - The makeAnimation type functions (like makeCat or makeDog) have associated events with the animation objects that were created.

4. We will make a subjective evaluation regarding your topic selection (relevance to your web site topic), originality, professionalism, look / feel, quality of blog post with instructions. Remember that even if there are some web design gurus in the class, you are not expected to measure up to this, just show mastery of the topics presented in class so far.

Suggested Approach

1. Carefully study the links that were provided (on the 3344 labs page under this lab).
2. Remember that this lab is about learning JavaScript, so you may not use jQuery for this lab.
3. As you work on this lab (or any lab that involves JavaScript), work in Chrome, press F12, and watch the console tab - this is the only place that you'll see JavaScript error messages. To debug, add console.log() statements to print information to the console. As things begin to work, you might want to comment out your debug statements instead of deleting them (might be useful again).
4. Always use the NetBeans menu option: Source – Format. Otherwise, it will be very hard to understand your own code. When you click on a starting brace, NetBeans should highlight that brace and its matching ending brace so this also helps you to understand nesting.
5. When you are trying to associate events with the animation objects, remember to appropriately use the “this” keyword appropriately and probably have to copy “this” to “self” (needed because the setInterval and/or setTimeout JavaScript methods change the value of “this”). Sample code was provided that shows a correct implementation for this.
6. As with ANY programming project (and especially if you are new to JavaScript), code just a few lines of code between testing.