# Homework: Update a website to add tabbed content.

**Step 1: Understand the features current website**

First open the index.html to see the site in a browser and take a look at what is displayed.

**Step 2: Understand the source code of the current website**

Next explore the html, css, js, and image files of the website to see how the site is put together.

You may notice that you have a *styles.css* file which is included on line 3 of the *index.html* file. There is a property display:none which may be new to you. All this style does is hide an element. display:block does the opposite, it will show an element (and make it a block element). Using these two properties you can show and hide an element on the page. You can try changing the property in the css and see what happens to the page. Make sure you return the css rules to the way they were originally so that you can do the rest of this exercise.

You may also notice the *lib/classy.js* file and its two functions called addClass and removeClass. These functions are part of a library called "classy". A library is just a JavaScript file that someone else has built that you can use in your website. A library usually makes functions available to you that you can use in your own code. You can add it with a script tag just like your own js files. In this website, the library is already added on line 18 of the *index.html* file.

In the *lib/classy.js* file, you'll see an explanation of how the addClass function works and also its opposite, the removeClass function. It is common in libraries to include a little documentation with the functions. Reading these lets you see how they work without having to understand the code itself.

Now take a look at the *index.js* file. This is the website's main JavaScript file and is also already included in the *index.html* file. It is executing the function addClass which is adding the class "active" to the element with id of "animal" when the page loads. You can inspect the element to see that the class "active" has been added to the element with the id of "animal". See what happens to the page if you change addClass('#animal', 'active') to use vegetable instead of animal: addClass('#vegetable', 'active'). Look back at the *lib/classy.js* file to see the explanations of how this works again.

**Step 3: Add the tab feature.**

Now you can start the assignment

Requirements: You must add a "tabbing" feature to this web site so that each section (animal, vegetable, and mineral) is shown when you click the corresponding button with the same name. Watch the *working-tabs-demo.mov* that is included to see how the tabbing feature is supposed to work.

You will need to use a few things you've learned so far to make this work: declaring your own function, using the two functions in the classy library, using conditional statements, and adding your function to the onclick attribute of the button elements on the page.

1) Write a function called "show" that takes one argument and does the following:

show('animal'); // --> should show ONLY koala picture

show('vegetable'); // --> should show ONLY carrot picture

show('mineral'); // --> should show ONLY mineral picture

Hint: Use the addClass and removeClass functions you learned earlier and conditional statements we learned in class.

Another Hint: You can start by just running this function in the index.js to see if it does what you expect.

2) Now you can use the show function in the onclick attribute of the button elements to make each button run your function. For instance <button onclick="show('animal')"> which will run that code whenever the button is clicked. Remember that it should show ONLY its corresponding picture when clicked, not the others. You may need to go back and fix your function if it isn't working right. Keep changing it and testing all the buttons until they all work.