**Unit 10 Lab – Dynamic HTML**

**Exercise 1 – Slideshow**

1. Create a new folder called “unit 10” containing a folder called exercise-1-slideshow
2. Copy the demo slideshow code to the exercise-1-slideshow folder
3. Replace the images with three new images of your own choosing
4. Add two extra images. Note you will need to add an extra “slide” div for each new image. Plus you will need to add a corresponding “navigation-dot” span
5. Make the slider full width. You need to modify just the CSS
6. The navigation dots are currently hard coded into the html. Change this to generate the navigation dots using JavaScript

**Exercise 2 – Slideshow**

Using the images and overlay text from exercise 1, recreate the slider using the Swiper (<https://swiperjs.com/get-started#installation>) JavaScript library

1. Create a new folder called exercise-2-shideshow
2. Create a basic version by following the instructions above and using the code supplied
3. Once you have the basic version working add the images to the slides
4. Add the html and CSS for the overlay text
5. Fix any remaining issues

**Exercise 3 – Lightbox**

1. Create a new folder called exercise-3-lightbox
2. Copy the demo lightbox code to the exercise-3-lightbox folder
3. Generate a set of navigation dots for the lightbox. The code for the navigation dots code in exercise 1 can be used as your starting point
4. Add a click event handler to each navigation dot so that that dot can be clicked and the corresponding image displayed

**Exercise 4 – VenoBox Lightbox**

Using the html and images from exercise 3, recreate the lightbox using VenoBox <https://veno.es/venobox/> JavaScript library

1. Create a new folder called exercise-4-venobox-lightbox
2. Create a basic version by following the instructions above and recreate the demo using a few images
3. Hint: add a link (<a> tag) to each <div> with a ‘image-box’ class. The link should have a href to the respective image, a ‘data-gall’ attribute and a class called ‘image-box-link’

**Exercise 5 – VenoBox Lightbox (advanced)**

1. Create a new folder called exercise-5-venobox-lightbox-advanced
2. Copy the html file from exercise-3
3. Add the VenoBox CSS link and JS script tags to the html file
4. VenoBox requires all images that you want in the lightbox to be wrapped in an <a> tag
5. In the JS file:
   1. Get a list of nodes with the ‘.image-box’ classname
   2. Loop thru the list of nodes
      1. Create a new <a> element
      2. Add the 'image-box-link' class to the <a> element
      3. Add a ‘data-gall’ attribute with a value of ‘gallery01’
      4. Add a ‘href’ attribute with a value equal to the ‘src‘ attribute of the ‘img’ element
      5. Add the new <a> element to the ‘.image-box’ node using appendChild()
   3. Create a new VenoBox object
   4. Set the numeration, infinigall, share and spinner options for the VenoBox object
6. In the CSS file:
   1. Add a 'image-box-link’ class
   2. In order for the lightbox to work this class needs to set the height. Otherwise it will not be clickable
   3. Hint: Set the ‘position’ property to overlay the link element on top of the image element and give it a height equal to 80%

**Exercise 6 – Build a table dynamically from data**

1. Create a new folder called exercise-6-table-from-data
2. Copy the code and data from demos/table-data-json
3. Modify the render function to use template strings (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals>)
4. Add two additional data fields to the table:
   1. countryCode
   2. types

**Exercise 7 – Build a grid of images dynamically from data**

1. Create a new folder called exercise-7-images-from-data
2. Copy the code and data from demos/grid-images-json
3. Modify the render function to use template strings (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals>)
4. Using the VenoBox JS library, display the larger version of each image in a lightbox
5. Set the numeration, infinigall, share and spinner options for the VenoBox object