CSCI 311 Spring 2020: Lab 3

In this lab, students will use JavaScript to validate form data, compute values, and generate HTML and CSS.

Learning Objectives:

- Write JavaScript helper functions to validate form data
- Write external JavaScript to compute values based on user input
- Write external JavaScript to generate HTML
- Write external JavaScript to modify HTLM and CSS on user input

What to hand in:

- zip and submit the following files (as a single zip file) to VIU Learn no later than Feb 8, 18:00:
    - index.html
    - Lab3Form.html
    - Lab3Compute.html
    - Lab3Dynamic.html
    - Lab3Dynamic.js
    - Lab3CSS.css
    - any media files

Note:

- late submissions will be penalized 20% per day
- All work must be individual
- Plagiarized work will result in a mark of 0. Further penalties may apply

Marking Scheme:

- Specifications: 4 marks
- Requirements: 4 marks
- Code standards: 4 marks

Instructions:

All html files:

All files must include the following:

- A header element with a title
- A nav element with:
    - Links to all other Lab 3 html files
- A main div that contains the portion specified below for each file
- A footer with your name, the date, and copyright information

index.html

This file should contain links to this lab's html files: Lab3Form.html, Lab3Compute.html, Lab3Dynamic.html. You can add any other content you wish.

Lab3Form.html

This page will contain the same form as last lab, but will add some client-side validation. We will run through this code in lab.

- Create a copy of the Lab2Form (or grab a copy of my solution of the course site)
- Modify it as follows:
  - In the two inputs of type number
    - remove the min and max attributes
- In a script tag in the head of the file, add the following functions:
  - checkNumber(var min, var max, var id)
    - gets the value of the element with the matching id, and if the value is not within min and max (inclusive) change the border of the element to dashed red line
    - use the onchange attribute to call the function from each of the inputs of type number, passing in the appropriate min and max values
  - fixName()
    - gets the value of the element with id name, and modify it as follows:
      - convert to all lower case
      - remove all spaces
      - replace all digits with _
    - use the onblur event (when the element loses focus) to trigger the function call
- When submitted, the form should generate an html file such that:
  - the page's background colour matches the colour selected
  - the larger of the two numbers is given

Lab3Compute.html

This page will act like a simple calculator where the user can enter two different numbers (integer or floating point) and an operator, and then see the answer.

Create a page that meets the following specifications:

- All JavaScript is located in a script element at the bottom of the body element
- A form lets the user enter:
  - Two integers
  - Choose an operator (+, -, /, *, %, **)
  - Compute and display the output
- NOTE: do not allow the user to do any of the following:
  - Divide by 0
  - Raise 0 to a power
  - In both these cases, highlight the offending input box, and open an alert explaining what was wrong. Do not compute anything or output it.
- A reset button should clear all values from the form
- Hints:
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/switch
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/try...catch
  - https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/number#step

Lab3Dynamic.html

In summary, this page will contain an image, and a list of one or more animals. The user can add more animals to the list by clicking a button, or reset the list just contain "dog" using a reset button. When an animal's name in the list is clicked, its picture will be shown (replacing the original picture), and its text colour will be changed to reflect that it is select.

Create a page that meets the following specifications:

- All JavaScript is located in Lab3JavaScript.js
- link to it in the `head`, after the title:
    - `<script type="text/javascript" src="Lab3Dynamic.js"></script>`
- The basic starting html matches the Lab3DynamicStart.html that is given (you will have to add attributes to some elements, but do not add html elements to the file).
- When the page loads, open a prompt that gets the user's full name. Modify the contents of the level 1 heading to welcome the user
- When the "click me" button is clicked, add a new animal to the unordered list in the div with id "animals".
    - Get an animal at random from an array of animal names that is located in Lab3Dynamic.js (You'll need to declare an array that has some animals in it...
        - let animals = "Dog", "Cat", ...].
    - add this animal as a list element to the list
    - Support at minimum 6 different animals being loaded into the list.
- When any of the animal's names (in the list) are clicked
    - change the text colour of that animal to Red (for a challenge, use a class to do this, and css to style all elements with that class. The trick will be to remove the class from all other elements.)
    - change the image to the animal in question.
    - Get the animal's picture from an array of picture file names. Again, you'll need to create a static list of animal-file name strings
        - let fileNames = ["Dog.jpg", "Cat.jpg", ...];
- When the "undo" button is clicked reset the list to contain only "dog"
- It's up to you how you handle adding an animal if it is already in the list. Duplicates are ok, or you could try to avoid them...
- Hints:
    - https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement
    - https://developer.mozilla.org/en-US/docs/Web/API/Element/setAttribute
    - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

Specifications:

- An index.html file is located in the Lab3 directory, as specified above
- Lab3Form.html produces correct results when form is submitted.
- Lab3Form validates the user input as described above
- Given php file is not modified
- Lab3Compute.html computes +, -, *, /, %, and ** (power) for the two values entered
- Lab3Dynamic.html lets the user add animals to a list, and show an animal when the list item is clicked

Requirements:

- All errors are handled gracefully in JavaScript (no divide by 0, user clearly informed of error's location)
- All submitted code must be error free

- All permissions for all files and folders correctly set
- all media is correctly attributed and sourced (also put the source link in a comment in the html)
- Submission followed all instructions

Code Readability, and Comments:

- Code adheres to all course code standards