Advanced Web Development and Web Scraping
Fall 2018
Assignment #3 -- Javascript

1. It is recommended that you complete the Javascript tutorial from **JS Home** to **JS Mistakes** at https://www.w3schools.com/js/default.asp *except* for the following:
    a. JS Array Sort
    b. JS Dates, JS Date Formats, JS Date Get Methods, JS Date Set Methods
    c. JS Random
    d. JS Switch
    e. JS Bitwise
    f. JS RegExp
    g. JS Errors

2. Complete the following Exercises from w3schools.com (you are responsible for these concepts, but these Exercises will not be graded):
    a. Javascript Variables, Exercises 2, 4, 6
    b. Javascript Functions, Exercises 3, 4
    c. Javascript Objects, Exercises 1 – 3
    d. Javascript Events, Exercises 1- 3
    e. Javascript Arrays, Exercise 2

3. Create a webpage that uses Javascript to store three web page URLs in an array, and uses a loop to output the links. The page should look as follows, with the header generated in HTML and the links generated from your Javascript array. Note that links must be active links (the URLs are output using the proper <a> … tags), and the list items are numbered.

    Some useful webpages are the following

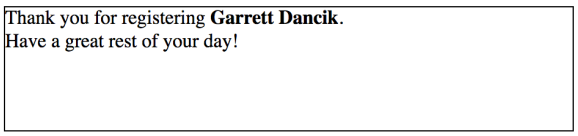
    1. http://www.easternct.edu
    2. http://gdancik.github.io/
    3. https://easternct.blackboard.com/

4. ***Multi-page registration site.*** Modify the *registration.html* page as described below. This assignment must be completed using Javascript, without modifying any of the HTML code, unless stated otherwise.

    a. Use CSS to change layout of the "page" div so that it is displayed on the top right of the "login" div as shown below: Hint: should the "page" div be "static", "relative", "fixed", or "absolute", with respect to the "login" div? Once set, where should its position be? Does the "login" div position need to be changed? Note that for this question, you must answer by changing the position properties and by adjusting the location (but note that you can also accomplish this using the *float* property).

    b. When the user enters their first name and clicks the Continue button, they are taken to the next "page" and prompted to enter their last name. The page number, label, and placeholder are all updated:

    c. After entering their last name, a confirmation is displayed:



    d. If at any point the user clicks *Continue* without entering a name, an alert is issued that a name must be entered, and we stay on the current page.

5. Modify the web page you created for Assignment #2 as follows (you may simplify your web page if you like, but a table with at least 4 rows and 3 columns (containing a checkbox, a numeric input, and the item name) is required. Optionally, you may include tooltips and links to more information.

    a. Your table should be created entirely in Javascript by first creating an array for the items/products in your table and then using a loop to generate the HTML to display the table. Your array should contain either item names (as strings) or objects that describe the different properties of each item that will be displayed (such as its name, tooltip text, and link).

    b. The numeric inputs should be *disabled* by default (simply include *disabled* inside the <input> tag).

    c. When the user checks a checkbox, the corresponding numeric input is enabled (set the *disabled* attribute of the element to *true* in Javascript) and given the value of "1".

    d. When a numeric input loses focus (which can be detected with the *onblur* event), check whether or not the input is empty. If so, uncheck the checkbox and disable the numeric box.

    e. When the user Submits the form, the submit button's *onsubmit* event should call a Javascript function which summarizes the order, including the items and the quantity of each. This summary can be displayed in an *alert* or can be written to the window. As discussed previously, all inputs should be part of a single form, and invalid inputs should not be allowed (this will be the case if the form is set up correctly).

    f. The logic of your Javascript code requires that each check box is linked to its corresponding numeric input (so that you can enable the input when checked, for example). This can be accomplished by giving each pair of inputs matching id numbers, e.g., "check1" and "num1".