# SY306 Lab Four

## Dynamic HTML and JavaScript for good and evil

## Introduction

This week we continued JavaScript and learned about how to truly make our pages come alive by changing the content dynamically. For this lab you will practice using this knowledge by making your products page more interactive and adding form validation to your sign-up form.

## Requirements

**You must create a folder in your `public_html` called "Lab04" (without the quotes) and store your work in that directory.** Copy your work from Lab03 into this directory. Do NOT modify the contents of the Lab03 directory. **Tip:** an *easy* way to copy your Lab03 files to Lab04 is to run the following command from your `~/public_html$` directory:

```
cp -p -r Lab03 Lab04
```

Using Firefox or Chrome is highly recommended for this lab. Make sure you know how to use the Web Console in Firefox (Go to "Web Developer-> Web Console `[CTRL+SHIFT+K]`) or the JavaScript Console in Chrome (Tools --> Developer tools `[CTRL+SHIFT+I]`) The Web/JavaScript Console will help with finding syntax errors in your JavaScript code! **Remember**, to aid in debugging, use `console.log();` to output data to the browser console.

1. **(10 points)** Comments are now especially important – explain at a high level what each function is doing and what you are trying to accomplish.
2. **(30 points) View shopping cart - a better solution** - modify the JavaScript code for the viewCart() function from Lab03, and, if needed, the HTML code in products.html, such that when the user clicks on the "view cart" button, the content of the shopping cart is added to products.html as an HTML table, without replacing the old content of the products.html (the list of products normally shown by products.html should still be available). Output a meaningful message if the shopping cart is empty. Hint: 1) you can create an empty element in HTML, and later change the content of that element with JavaScript (DHTML).
3. **(40 points) Form validation for sign-up form.** In your sign-up form for your website, which you did in lab 2, make sure the action is http://courses.cs.usna.edu/SY306/tools/FormChecker/submit.cgi. If you do not have an input type text, input type password, input type email and textarea in your form, or if any input fields are missing a name, make sure you add them. Now add the following JavaScript validation checks for some of the fields in your form. Note that HTML5 can do

some of the checks. However, not all browsers support the latest features of HTML5, so you need to perform these checks using JavaScript. The form should **not be submitted** until these fields are valid. If any field is invalid, the user should be somehow notified of the error and should be given the opportunity to fix it (the form is not submitted).

a. Ensure that the **password** size is at least 6 characters - notify the user if it is not long enough. This check should be performed as soon as the user moves out of the input field (using tab or clicking somewhere outside the input). The check should also be performed before the form is submitted.

b. Ensure that the **password** contains at least one number – notify the user otherwise. This check should be performed as soon as the user moves out of the input field (using tab or clicking somewhere outside the input). The check should also be performed before the form is submitted.

c. Ensure that the **email** is valid – notify the user otherwise. This check should be performed as soon as the user moves out of the input field (using tab or clicking somewhere outside the input). The check should also be performed before the form is submitted.

d. The first **text** field on your form is filled out – notify the user otherwise. This check should be performed as soon as the user moves out of the input field (using tab or clicking somewhere outside the input). The check should also be performed before the form is submitted.

e. The **textarea** field: let's assume that the server side script does not support "<" in the input. So, all the "<" in the textarea field value, if they exist, should be replaced with "&lt;" (using JavaScript). Hint: The String object provides some useful functions.

**NOTE 1:** The user notifications for part **a-d** above must be displayed with text to the right/left of the input fields. You cannot use an `alert();` call for this requirement. Additionally, the form should submit only if all checks are OK and the textarea string has any "<" characters replaced by "&lt;".

**NOTE 2:** While HTML5 has a required attribute for form inputs, and the type can be set to e-mail, these actions alone are insufficient to meet the various requirements of part 3 above. The form validation must be performed using JavaScript.

4. **Attack:** There are many ways to redirect a user to a new page (and you've already implemented some). We'll try one more way here (see extra credit for another one):

- **(15 points) fakeindex.html** Research document.location to see how you can use it to redirect with JavaScript. Then copy your index.html into a new file called fakeindex.html. Modify fakeindex.html such that when the user moves the mouse anywhere on that page, the user is redirected to the official SY306 course page.

5. **(5 points) Links:** Add 2 new links to your default.html page. One to Lab04/products.html, and anoter link to Lab04/fakeindex.html.

**Note:** Note that in JavaScript, global variables are global for the life of the page. This means that **variables** are persistent and available to other parts of a script (or other scripts, if you have more than one).

## Extra Credit

For a nominal amount of extra credit do some/all of the following:

1. **Attacks:**

    a. Injection attacks - write all the answers on your coversheet: Change the action for your form to be http://courses.cs.usna.edu/SY306/tools/FormChecker/submit_unsafe.cgi

        i. Bring up your form in the browser. In the input type text on your form, instead of plain text, insert some HTML code (ex. <h1>This is a header</h1>). Complete the rest of the form satisfying the checks so the data is submitted. Submit the form. What happens? Why do you think it happens that way? Write your answer on the coversheet.

        ii. Bring up your form in the browser. In the textarea input on your form, instead of plain text, insert some HTML code (ex. <h1>This is a header</h1>). Complete the rest of the form satisfying the checks so the data is submitted. Submit the form. What happens? Why do you think it happens that way? Write your answer on the coversheet.

2. Add some interaction to your list of products to display more or less of the product description/information: initially the page should display less information and have a "+" sign. If the user clicks on "+", the longer description/information is displayed, with a "-" sign too. If the user clicks on "-", we come back to the original state (less info and + sign). Hint:

    a. You can have 2 elements with similar ids for the + an - versions, and you can use the CSS display property to show/not show an element.

    b. You can change the style with JavaScript

## Deliverables

1. Your pages should contain all of the elements described in the requirements section above.

2. All of your files should be in a folder called "Lab04" (without the quotes) in your public_html. **Your instructor will assume that your web pages are viewable at http://midn.cyber.usna.edu/~m21xxxx/Lab04/index.html** and products.html, and signup.html (if appropriate).

3. **Turn in (due before lab on Friday):**

    a. **Paper submission:** turn in the following hardcopy at the beginning of Lab05.

        i. A completed assignment coversheet. Your comments will help us improve the course.

    ii. **Electronics submission:** Submit all Lab04 files via the online system: `submit.cs.usna.edu` by the following deadlines:

        - Section **4341**: 23:59, Wednesday, February 6th
        - Sections **1151, 3351, 5551**: 23:59, Thursday, February 7th