SY306 Lab Three

JavaScript Intro and ClickJacking Defense

Introduction

This week we've introduced the scripting language JavaScript and shown you how it can be used to customize and animate the user's experience. For this lab you will practice using this knowledge on some warm-up exercises and then you will create a rudimentary shopping cart. You will also implement the "frame-busting" defense for the clickjacking attack.

Read this lab completely before you begin typing.

Requirements

You must create a folder in your public_html called "Lab03" (without the quotes) and store your work in that directory. Copy all your work from Lab02 into this directory. Make all modifications in the Lab03 directory. **Do not modify the files in Lab02 folder.**

Tip: an *easy* way to copy your Lab02 files to Lab03 is to run the following command from your `~/public_html$` directory:

```
cp -p -r Lab02 Lab03
```

- **Debugging:** 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!
- **Tip:** to aid in debugging, use `console.log();` to output data to the browser console.
- You *may* use this JavaScript starter page to get started with JavaScript code. Copy the starter JS code to your existing index.html and products.html and use the my_write(), my_writeln(), and my_finish() functions provided in the starter code instead of document.writeln(), to help with debugging. Once everything works, comment out the call to my_finish() in each of your HTML pages that use JavaScript. **Make sure your page still works after you do this!**
- **Note:** 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).

1. **(5 points)** Comments are now especially important – explain at a high level what your JavaScript is doing, and complex parts.

2. **(10 points) greeting:** Modify your existing Lab03/index.html page to include a JavaScript script that displays in the browser window (not a pop-up) a greeting based on the "minute" of the current time: if the current minute is an odd number, then print one greeting, otherwise print a different greeting.

3. **(65 points) rudimentary shopping cart using JavaScript:**

    a. In your products.html page, using JavaScript, add a "shopping cart" variable – this should be an array, to store the product ids/names of all the products in the cart

    b. Modify your HTML code in products.html to add an "Add to cart" button for each product. You can use an input element of type button to create the buttons. When the button is clicked, a JavaScript function addToCart() should be called, with a parameter value that identifies the product to be added to the cart (an integer which will be the productID is best).

    c. Write the addToCart(productID) JavaScript function to add the new productID to the shopping cart array: if the product is not in the shopping cart yet, add it to the shopping cart and display a confirmation pop-up message. If the product is already in the cart, display a pop-up message stating that the product is already in the cart, and do not add it again. The pop-up messages in both cases should contain either the product id or name of the product.

    d. Modify your HTML code to add a "View Cart" button to your products.html page. When the button is clicked, the viewCart() function you will write next should be invoked.

    e. Write the viewCart() function to display the content of the shopping cart as an HTML table. Output a meaningful message if the shopping cart is empty. It is OK for now if the old content of the products.html page disappears once you display the shopping cart content.

    f. Modify your HTML code to add a "Empty Cart" button to your products.html page. When the button is clicked, the emptyCart() function you will write next should be invoked.

    g. Write the emptyCart() function to empty the shopping cart. Display a meaningful message that the cart has been emptied.

4. **(15 points) ClickJacking defense:** In the previous lab you implemented a clickjacking attack. We will see now how such an attack can be prevented.

    a. Copy your index.html from Lab02 to a new file framedindex.html in Lab03

    b. You already created an invisible iframe in your index.html, such that when the user clicks a particular image, the site inside the frame plays some media (this was for the previous lab). For this lab, in Lab03 folder, modify the iframe in framedindex.html, so it loads your own detail.html file that you copied from Lab02.

    c. Modify the detail.html file (in Lab03) so you prevent it from being loaded into a frame, by using a "frame-buster" defense; if detail.html is loaded into a frame, it will "burst" out of the frame into the main browser window: In the <head> part of detail.html, include the JavaScript code below that checks whether the site is loaded into a frame, and if yes, it replaces the main window with itself:

```
if (top != self){
      top.location = self.location; }
```

d. Test your defense: load "framedindex.html" in the browser. What happens?

5. **(5 point) Links:** Add two new links to your default.html page (do not remove the existing links): one link to Lab03/index.html and one to Lab03/framedindex.html

Extra Credit

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

1. Modify your shopping cart to allow you to add the same product more than once – keep track of the quantity added. (Hint. You can use a double indexed array.)

2. Use a CSS/JS Modal to display the pop-up message specified in 3.c above.

3. Modify your code so the style of the "View Cart" button changes when there are items present in the cart.

4. **(+10 points)** Store the shopping cart on the client side, such that you can browse your website and still remember the content of the shopping cart. (Hint: Use HTML5 Web Storage + JSON)

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 "Lab03" (without the quotes) in your public_html folder. **Your instructor will assume that your web pages are viewable at http://midn.cyber.usna.edu/~m21xxxx/Lab03/index.html** (and products.html and framedindex.html). You may want to check that this URL is viewable and that everything works correctly *from a computer where somebody else is logged in.*

3. **Turn in:**

   i. **Paper submission:** turn in the following hardcopy at the beginning of Lab04:

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

      b. On the back of the cover sheet, write your answer to question 4.d

   ii. **Electronics submission:** Submit all Lab03 files via the online system: `submit.cs.usna.edu` by the following deadlines:

   - Section **4341**: 23:59, Wednesday, January 30th
   - Sections **1151, 3351, 5551**: 23:59, Thursday, January 31st