CSCE 102 Lab 10 — Interactive web pages

Preview, Buttons & Event Handlers

## General information

* Read the entire assignment before you start.
* Always maintain backup copies of your work, and work on the H drive during lab.
* Never share your password or memory stick with anyone. Maintain the standards of academic honesty discussed in lecture and in previous labs. The work you turn in must be your work.

**Project Reminder — Do not wait until the last minute to work on your project. The end of the semester is coming up fast.**

## Part I — Due at the end of class

* Open your All\_Index page in scITE and make the image of you that you inserted on week two a hyperlink to open the same image in a new tab. (Use the anchor element with the target attribute).
* Look at the list at the bottom of your index page and be certain that all of the links to your lab assignments are there and working.
* You will show this to your lab instructor and print a copy before you leave.

## Part II (lab10a*xx*.html) — Also due at the end of class,

**Create an Interactive Web Page using Buttons and Textboxes**

Problem: Create an interactive web page with the following specifications.

* Create a web page that contains a blockquote element.
* In an internal CSS set the color of the background of the page to a color of your choice.
* Create a form that contains two buttons and a text box.
* The first button will change the text color of your button to ***purple*** when it is clicked.
* The second will change the color of the text in blockquote to ***green***.
* Add the CSS to make the textbox and all of the buttons be one particular color (any color of your choice) when the page is loaded.
* Save the assignment in your ***All\_102Submissions*** folder.
* You will show this to your lab instructor and print a copy before you leave

## Part III — lab10b*xx*.html

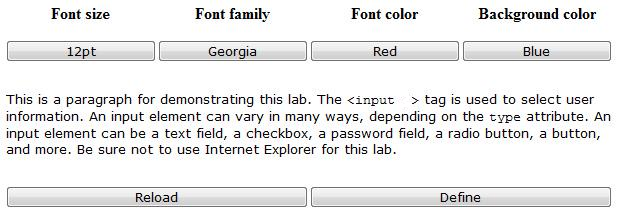
Problem: Create a web page that contains a table of buttons that should apply various styles to a paragraph according to the specifications listed in subsequent sections.

The web page must provide the user a preview of the new style before the change is applied.

The page must also support resetting the style back to the default using event handlers and functions.

You will need use the *getElementById()* JS method to address multiple HTML elements within your webpage.

1. In the body of the page create a **table with 4 rows** such that: the first and second rows have 4 cells, the third row has a single cell, and the fourth row has 2 cells. **Fill in the table as described below** so that it is similar to the following example



* + The **first row** will contain the **column headings.** They contain the following text: Font Size, Font Family, Font Color, and Background Color.
  + The **second row** will contain the **buttons** that will change the property described in the respective column. Give each button a value[[1]](#footnote-2) (attribute) that corresponds to the new value that will be assigned to the respective property. E.g., underneath “Font Size” the button must display the text “14pt”.
  + **Second** Row **must contain** the following **values**: *FontSize:18pt FontFamily:Tahoma FontColor:Green BackgroundColor:Yellow*
  + The **third row** must contain a **paragraph** element**.** You will need a single ***td*** element (*Hint*: use colspan). The paragraph tag must have an *id attribute*, and for the content of the paragraph write about the advantages/disadvantages of the *DuckDuckGo* search engine vs *Google* search engine.
  + The **fourth row** will contain **two** **buttons** with the attribute *value* set to *Reload* and *Define*.

1. Use **an internal style sheet** with CSS rules that accomplish the following:
   * Set the paragraph element’s font color to black, background color to white, font size to 12pt, and font family to “Arial”.
   * Set the td element’s width to 180px, and height to 80px
   * Set the th element’s text align to center.
   * Set the input element’s width to 100% and font family to “Times New Roman”.
   * Change the background color of all **th** elements to any color of your choice.

**Part b-II — Event handlers & JavaScript**

1. For **each button in the second row** add an ***onclick*** event handler that will update the style of the paragraph to the value1 indicated by the button. Use *document.getElementById()* function to refer to the paragraph by the id you gave it earlier.
2. For **each heading in the first row** add an ***ondblclick*** event handler **to each th element** that will reset the style of the paragraph to the defaults you defined in the style sheet for the paragraph element.
3. For **each button in the second row** use the ***onmouseover*** event handler to give the user a **preview** of the style that will be applied by modifying the style of the button itself to reflect the change.[[2]](#footnote-3) Use the ***onmouseout*** event handler to reset the style back to the default values

(*Hint: The default values are the same as what you have explicitly specified in your internal CSS*. *Use this.style.property=value to change the button’s style for the onmouseover and onmouseout event handlers.* )

1. Forthe **Reload button** the ***onclick*** event handler should reload the page by calling the *window.location.reload()* function.
2. Forthe **Define button** the onclick event handler should define a word selected or highlighted by the user, using a custom function named *defineSelection()*, that will be described next. (*Note: you must use either Chrome or Firefox for this part, IE will not render this properly.*)
3. **Add a script element to the head** element that contains the defineSelection() **function declaration**:

* The basic syntax for declaring a function can be found in your notes, on pg. 24 in the DOM Scripting textbook, or by Googling “JavaScript function.”
* This function must have a single statement (highlighted in yellow) that will change the window location to a Google search results page. In the statement we are **concatenating two strings** (a URL and the string returned by the getSelection() window method) and **assigning** the result to the window location property.

*window.location='http://www.google.com/search?hl=en&q=define:'+window.getSelection();*

**Part b-III — Test cases**

1. When you **click** on a button in the second row it should change the style of the paragraph in the way indicated by the button.
2. When you **double click** on a property name in the first row (event handler in the th element) it should change the style of the paragraph back to the default (specified by you in the style sheet).
3. When you move your **mouse over** a button in the second row the button’s style should change in the way indicated by the button. (i.e., in the way that the paragraph’s style would change if you clicked on the button (test case 2).) When you move your **mouse off** of the button the button’s style should change back to its default.
4. When you **click** on the reload button the page should reload.
5. When you **select a word** in the paragraph (highlight it with your cursor by clicking and holding as you move the cursor over a word) and then **click** the define button, a Google page with the definition of the word should load. (Note: you cannot use IE for this step, IE does not correctly implement the getSelection() window method.)
6. You must validate your code prior to submission at <https://validator.w3.org/>

(See pages 46-47 in your web design textbook)

#### Part c — Update *All\_Index.html*

1. Add relative hyperlink references to *lab10axx.html* and *lab10bxx.html* at the end of your list of weekly lab assignments.

1. value in this context refers to the value attribute of the input element, i.e., the text displayed on the button, not value in the CSS sense of property and property value. [↑](#footnote-ref-2)
2. E.g., if the button will change the color of the paragraph text to red when clicked on, then moving the cursor over the button should change the color of the **button** text to red (e.g., this.style.color=’red’). When the user moves the cursor off of the button the color of the button should change back to the default color (black). [↑](#footnote-ref-3)