CS 328 - Week 14 Lab Exercise - 2024-04-26

Deadline

Due by 5:00 pm on Friday, April 26.

Purpose

To get more practice using unobtrusive-style client-side JavaScript, including using it to validate a form.

How to submit

Submit your files for this lab using `~st10/328submit` on nrs-projects, each time entering a lab number of **94**.

Requirements

- You may work individually OR using pair programming for this lab exercise, your choice!
  - If you do use pair programming, make sure BOTH of your names appear in each file submitted!
- Whether working individually or using pair programming, make sure you have a copy of your lab exercise files, and that you submit them using `~st10/328submit` on nrs-projects, with a lab number of **94**.

Part 1

- FIRST: copy the files from **`~st10/for-328lab14-ex`** into your nrs-projects Week 14 Lab directory:

```
cp ~st10/for-328lab14-ex/* .   // don't forget the space and period
```

  - You should now have copies of the files:

    `lab14.php`

    `make_lab14_form.php`

    `make_lab14_response.php`

    `lab14.css`

  - This is a small working two-state PHP postback application, currently with a button that does nothing and no client-side JavaScript.
- Edit your copy of **`lab14.php`** so that its opening comment block includes YOUR name(s), and the URL from which YOUR version of this application can be run.
  - Then -- run it!
  - Confirm that, between PHP and HTML, it has certain, ah, input requirements for its form.
  - Confirm also that the "MUCK..." button currently does NOTHING.

You should now be ready to start adding some client-side JavaScript to this application.

Part 2

The goal here is to add a little unobtrusive-style client-side JavaScript so that the "MUCK..." button does something.

- Create a new file **`lab14.js`**, and put **EXACTLY** the following as its **FIRST** line:

```
"use strict";
```

  - This says to use strict-style for your JavaScript -- this can help you in finding errors.
- put a comment with your name(s) and today's date.
- In this file, make a small function **`changeIt`** that expects nothing, returns nothing, but has the following side-effects as its actions:
  - it declares a JavaScript variable and sets it to the JavaScript DOM object corresponding to some element in the page created by **`lab14.php`**
  - it **CHANGES** this element's content to a noticeable value of your choice -- if this function is successfully run, that element's displayed contents should change to what you put here.
- In this file, also put a JavaScript statement that sets the **`onload`** attribute of the JavaScript DOM **`window`** object to an **anonymous function** that does the following actions:
  - declares a JavaScript variable and sets it to be the JavaScript DOM object corresponding to the **`button`** element in **`lab14.php`**
  - sets this button object variable's **`onclick`** attribute to be the function **`changeIt`**.
- Edit your **`lab14.php`** so that, at the **end** of its **`head`** element, it includes:
  - the class-approved **`script`** element to include your external JavaScript **`lab14.js`** -- and include the attribute **`async="async"`**, since it is safe to load and execute this while loading and parsing this document's HTML.
- See if your button in **`lab14.php`** now changes your chosen element's content when it is clicked! Debug your JavaScript until it does.

Part 3

Fun JavaScript fact #1: JavaScript's **`String`** object has a method **`indexOf`** that expects a string to look for, and returns the index of its first occurrence in the calling string. It returns **-1** if the given string does not appear in the calling string.

Fun JavaScript fact #2: JavaScript has a modulo operator, **`%`** , that returns the remainder from integer division of its operands. So, for example, `(13 % 3) === 1`

The goal here is to now get a little bit of client-side JavaScript successfully validating a form (and refusing to submit it if it doesn't meet its validation requirements).

- In your file **`lab14.js`**, write a second function **`meetsSpecs`**. This function expects nothing, tries to make sure some aspect or aspects of a form about to be submitted are OK, and returns **`true`** if they are, and returns **`false`** if they are NOT.
- DECIDE on **at least ONE** of the following aspects of this form you want to validate:
  - making sure that the user has entered a (non-empty) string **with NO blanks** in the first textfield, and/or
  - making sure that the user has entered an **EVEN** integer in the number field
- declare a JavaScript variable and set it to be the JavaScript DOM object corresponding to each form widget you choose to validate
- write an **`if`** statement to do the desired check of the value in that JavaScript variable's **`value`** data field for

each form widget you choose to validate

  - if it fails that check, use the JavaScript **`alert`** function to print a suitable complaint in a JavaScript pop-up window, and return **`false`** to prevent the form from being submitted.
  - otherwise, return **`true`**

- In your file **`lab14.js`**, in the anonymous function that you set to be the value of **`window`**'s **`onload`** attribute, **also add** the following to what you have from Part 2:
  - declare a JavaScript variable and set it to be the JavaScript DOM object corresponding to the form element in the initial state of **`lab14.php`**
  - what if we are here in this PHP application's second state? Then there will be NO such form, and the previous statement would set that JavaScript variable to **`null`**
  - SO: write an **`if`** statement that checks to see if this JavaScript variable is **NOT** `null`, and only if it is **NOT** `null` should it:
    - set the **`onsubmit`** data field for this JavaScript object representing the form to the function **`meetsSpecs`**.
    - (that is, we don't want to try to change a document element that isn't currently part of the document! But if this form IS part of the current document, add a little event hander to check its entered contents and only allow it to be submitted if it passes those checks.)
- See if trying to submit **`lab14.php`**'s first state's form with a "bad" input into your chosen textfield now causes your JavaScript pop-up to display, and for the form NOT to be submitted! Debug your JavaScript until it does.

Submit all of your files using a lab number of **94**.

- If you work in a pair: I will leave it up to the navigator to decide if they would like to UPDATE their **`lab14.php`** so their opening comment includes the URL to *their* copy, or if they want to leave the URL for the driver's copy.
  - HOWEVER: remember that you *will* lose some credit if this URL does not work when I or the grader paste your submitted file's URL into a browser, in either case.