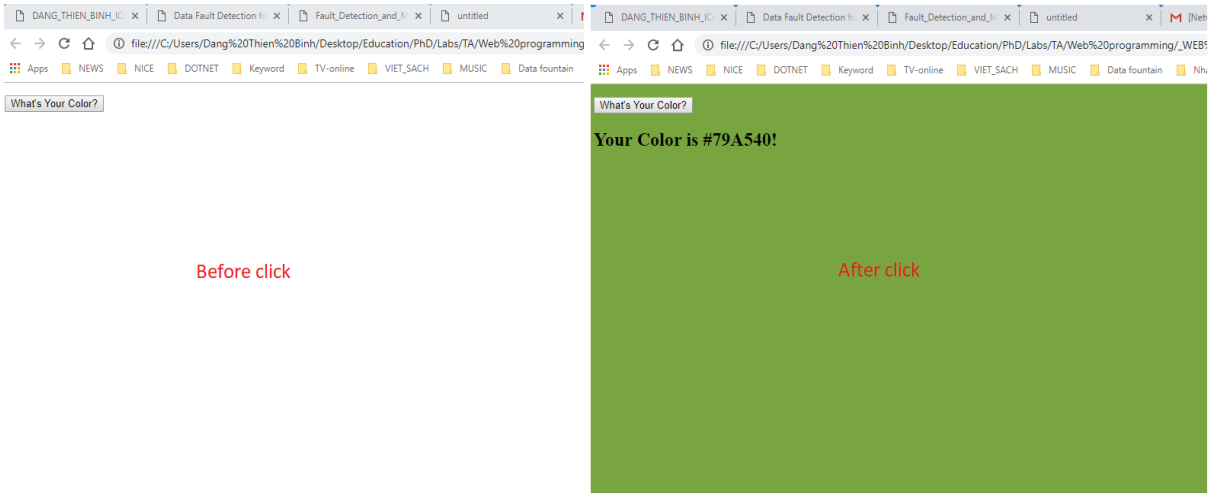
Web Programming Lab
Practice Week #9
Introduction to JavaScript

Problem #1 (10 points). Practice the examples in the lecture note.

Follow the instructions in the lecture note and submit source codes and snapshots of results with HTML files. (p9, p29, p32, p37).

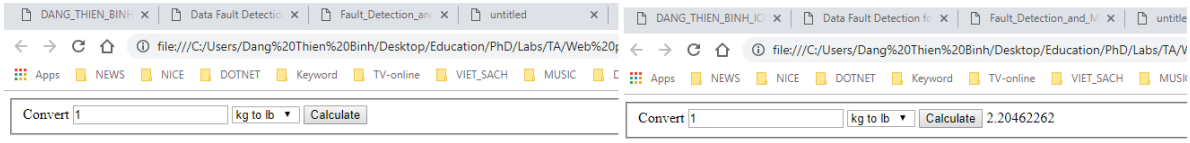
Problem #2 (20 points). What's Your Color?

Modify **colorifyme.html** so that when the button is clicked, it randomly decides the color of the page background. That is, it should choose a random hex value for the background between #000000 and #FFFFFF. In addition, it should replace any text in the heading tag that has the ID "mycolor" with the text, "Your Color Is <random-color>", (where random-color is the color randomly-generated for the background).



Problem #3 (20 points). Weight Conversion

- Create the file conversions.js which is referenced by conversions.html. Write a function convert() in conversions.js that takes the value in the text input and converts it from either kilograms (kg) to pounds (lb) or from pounds to kilograms, depending on the value selected in the dropdown box.
- The result should be displayed in the empty span with the id of #answer.
- The conversion factor from pounds to kilograms is 0.45359237, and the conversion factor from kilograms to pounds is 2.20462262.
- You should edit the HTML file **conversions.html** to add id's to the elements as necessary and you may assume valid input.

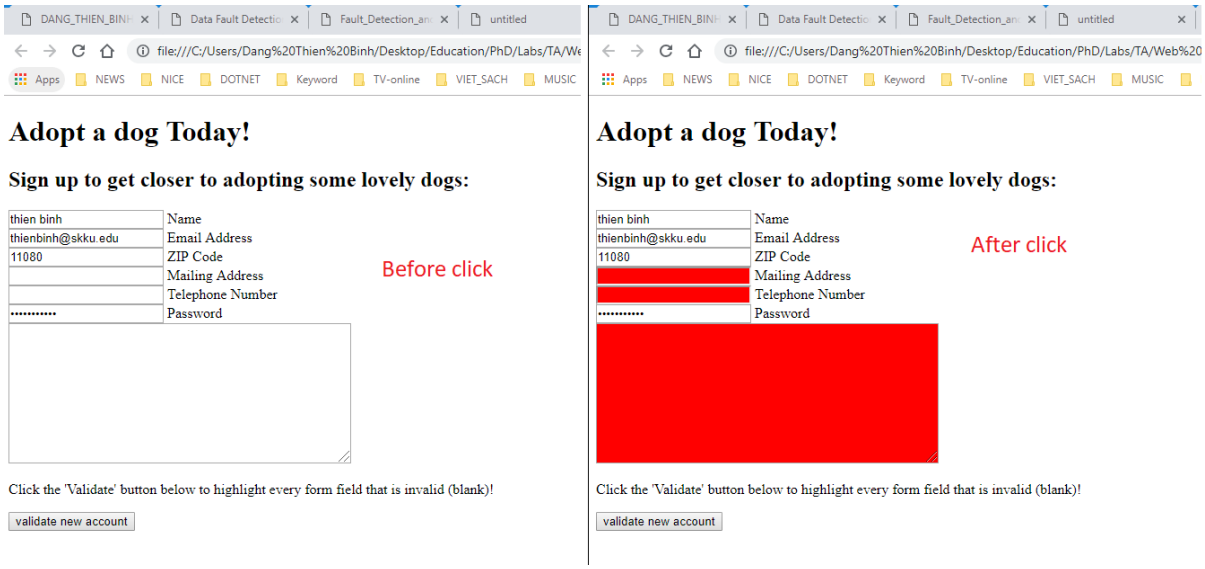


Before click

After click

Problem #4 (25 points). **Adopt a Dog *Today!***

Write a JavaScript file dog.js to be used with the given adopt-a-dog.html file so that when the Validate button is clicked, every input in the form including the textarea that has a blank value (empty string) will be highlighted with a "red" background color, as shown in the screenshots below.



Problem #5 (25 points). **Encryption**

Part I: Create the UI Elements

- The first task is to extend encrypt.html by adding UI controls. Add HTML code for the following:
  - A field for users to enter large (multi-line) amounts of text. It should be 10 rows by 60 columns in side and wrapped in a bordered field set box with the label "Text to Encrypt".
  - Add a second bordered field set box labeled "Encrypt Options" that contains:
    - ★ A button labeled: "Encrypt!"

- ★ A dropdown menu with the options, "Shift Cipher" and "Randomized", preceded by the boldly-emphasized label: "Cipher Type:"

Part II: Write the start of encrypt.js

- Now you'll write a bit of JavaScript testing code that pops up an **alert** box. This is just a test to make sure that your browser is running your JavaScript file, before we move on to the main exercises.
  - Create a new file and save it as **encrypt.js**
  - Put the following line of code into the file: **alert("Hello world!");**
  - Link your HTML page to your JavaScript file using a **script** tag
  - Refresh your page in the browser. Do you see the **alert** message? If so, move on. Otherwise, double-check your **script** tag syntax

Part III: Hello World Button

- Now let's set up a very basic JS event handler. Modify your JS code and HTML so that the "Hello, world!" alert message won't pop up until the user clicks the "Encrypt!" button.
  - Modify your JS file to wrap the alert into a function
  - Write a **window.onload** function to set up an onclick event handler for the "Encrypt!" button. This handler should call your new function
  - Refresh your page in the browser. Click the button. Do you see the alert? If so, move on. Otherwise, double-check your onclick and **window.onload** syntax and function, or ask a TA for help

Part IV: Implement a Basic Shift-Cipher

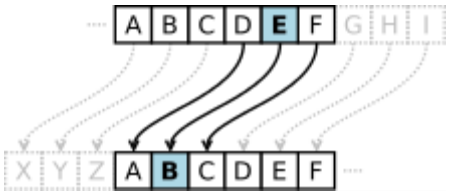
- Modify your JS code so that when the user clicks "Encrypt!", the text in the input text area will be encrypted using a basic shift-cipher, and output into the page's div with the ID output.
  - Make sure that your text area has an ID attribute so your JS code can talk to it
  - Modify your JS function so that it now takes the text area's text and generates a shift cipher (algorithm discussed later)

Part V: Implementing the Randomized Cipher

- In this part, you will implement the "Randomized" cipher option. Each letter in the English alphabet is mapped to one of the 25 other letters, but no two letters may have the same mapping (e.g., "a" and "c" cannot both map to "d").
- First, you should modify your button event handler to determine whether the "Shift Cipher" or "Randomized Cipher" option is selected in the dropdown menu. If the "Randomized Cipher" option is selected, it should call a new function which outputs a randomized encrypted version of the input text area to the output div.
- Just encrypt letters

Part VI: Implementing the Shift Cipher

- In this part, you will implement the shift cipher option. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.



- Add a dropdown menu to select a value to shift by in the Shift Cipher, the shift value should be from 0 to 9.
- Output the shift cipher encrypted version of the input text area to the output div.
- Just encrypt letters

Submission Guidelines

- A compressed file named W9_ID_NAME.zip via icampus
  - For each problem, create a folder named Problem#, and put the corresponding file(s) into the folder.
- Grading policies:
  - Your comments (if any) must be written in ENGLISH. o Clean code, good commenting are important components of your grade.