CS212: Web Programming – Spring 2019

Homework	Points	Announced	Due
#6	20[1]	Feb-25	Mar-08

Introduction

In this homework, you will practice combining HTML, CSS, and JavaScript. You will design a page with a simple restaurant bill calculator. Some required features may have not been covered in detail in class; you will have to explore resources such as your textbook, lecture slides, or online references to learn how to complete these features.

References

1. Chapters 2, 3 and 8 of the textbook; Web Programming Step by Step
2. HTML: HyperText Markup Language; MDN Web Docs
3. HTML Element Reference; W3Schools
4. CSS: Cascading Style Sheets; MDN Web Docs
5. CSS Reference; W3Schools
6. JavaScript; MDN Web Docs
7. JavaScript Tutorial; W3Schools

General Guidelines

Read the following guidelines carefully before working on this assignment.

1. This is an *individual* homework assignment. You may discuss ideas, ask questions or explain things to your colleagues. Nevertheless, you should solve the problem(s) independently.
2. You should submit your *own work*. Material brought from elsewhere (e.g. the Internet[2], a classmate, submission at a previous offering…) is not acceptable.

Submission Instructions

1. Submissions via email will not be accepted. Your work should be submitted via BBLearn by the due date.
2. For question #1, submit one PDF file, named Q1.pdf, that contains the answers for parts a to d.
3. For question #2, upload index.html, index.css, and index.js files to your CEFNS homepage inside a folder named hw6. If your CAS ID is id123, then navigating to (https://cefns.nau.edu/~id123/hw6) should display your index.html file.
4. Also, submit the files index.html, index.css, and index.js via BBLearn *and* type the link of your hw6 folder in the submission comment box. You must compress your files and upload a single archive. Only .zip archives are allowed.

[1] The homework will be graded out of 20 points, but it is worth 2% (i.e. 2 points) of the overall course score.
[2] Unless explicitly asked to do so.

CS212: Web Programming – Spring 2019

Questions

[4 points] Question #1

Answer the following questions by consulting any of the references mentioned above. Submit your answer in a one PDF file (Q1.pdf).

a) The HTML <input> element is used for capturing input from the user. How to set it to accept numeric values? How to configure it to accept values within a specific range? Write an example of an <input> element with id "input1" that allows numbers between 0 and 10.
b) When an <input> element has validity constraints such as minimum and maximum values, which JavaScript function can be used to check whether the <input> element data complies with the validity constraints? Write a JavaScript function that displays an alert box with an error message when an <input> element with id "input1" has invalid data.
c) When an <input> element has validity constraints such as minimum and maximum values, what CSS selector can be used to select the <input> element when its data is invalid (i.e. violates the validity constraints)? Write a CSS rule that changes the background color of an <input> element with invalid data, to pink.
d) What is the purpose of the <label> element?

[16 points] Question #2

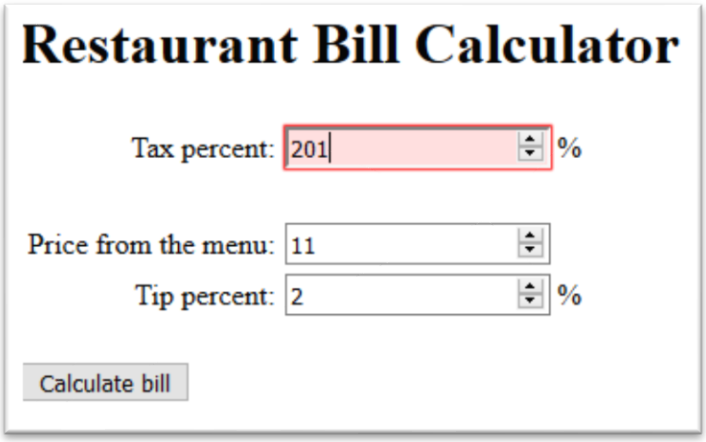
Implement a web page (idnex.html) that generates a restaurant bill. Use numeric input boxes for tax rate percentage, the meal price, and the tip percentage. After the "Calculate bill" button is clicked, if the input values are valid, display the bill below the button. The bill should show the meal price, the tax amount, the tip amount, and the total payment due. The screenshots below show how the page should behave.

1. You must use appropriate attributes for your <input> elements, including attributes for input validation. The tax rate percentage <input> element should accept a decimal number between 0 and 100, with a maximum of 2 digits to the right of the decimal point. The price <input> element should accept a decimal number between 0 and 1000, with a maximum of 2 digits to the right of the decimal point. The tip percentage <input> element should accept an *integer* between 0 and 200.
2. Use JavaScript to check for input errors and to display *one* error message in an alert box. For each user input that is outside of the ranges described above, there should be a separate line in the error message. If there is invalid input, remove the previously displayed restaurant bill before displaying the error message alert box.
3. The code that displays the error message should not use hard-coded values for the valid ranges of the <input> elements. Instead, retrieve those values by using the <input> element min and max attributes.
4. Implement CSS rules such that when an <input> element has focus, the element background becomes light green for valid input and light red for invalid input.
5. Use <label> elements for the three <input> elements.

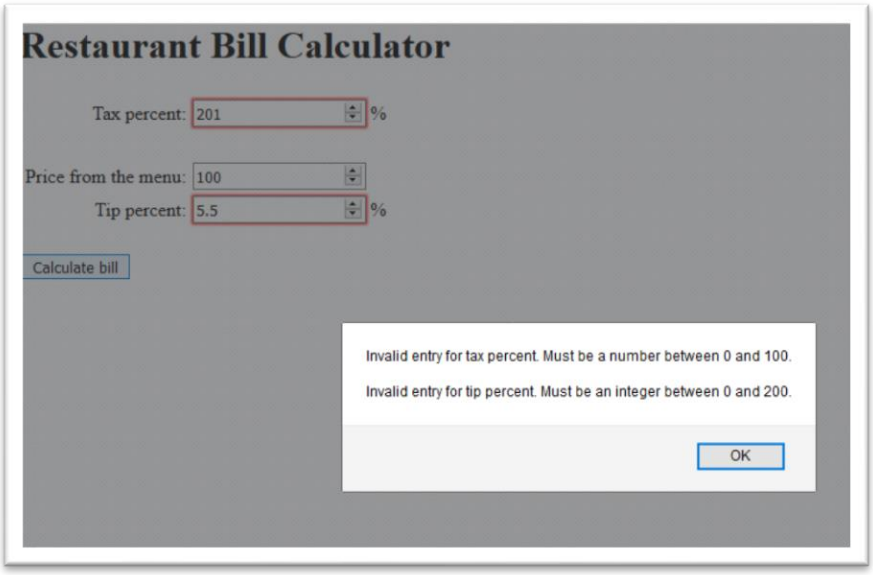
Grading Rubric	
Component	**Points**
Proper HTML elements	4
Proper CSS styles	4
Correct JavaScript validation & calculation	8

With best wishes
Dr. Mohamed Elwakil

After entering 201 in the tax percent input box, note the light red background color:



The "Calculate bill" button is clicked after entering a valid value for the price and invalid values for the tip and tax boxes. Note the error message with two lines for the two input errors.



Entering all valid values, and clicking "Calculate bill":