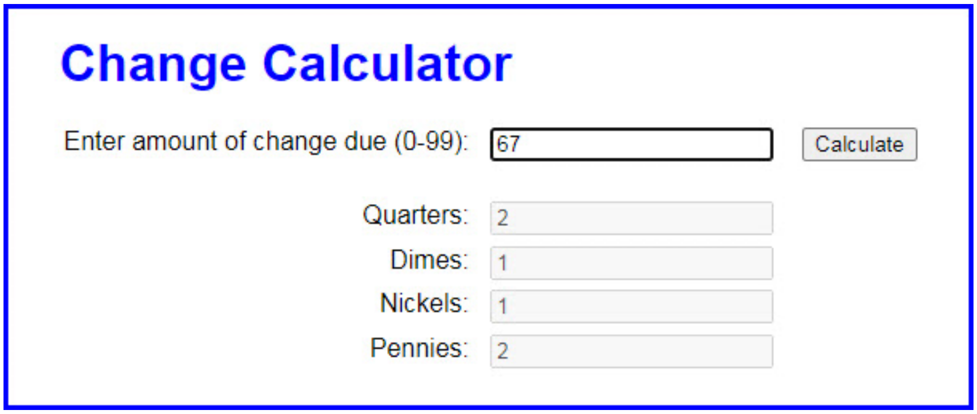
Guidelines for doing the extra exercises

- For all the extra exercises, you will start with the HTML and CSS for the user interface. Then, you supply the JavaScript or jQuery that's required to get the desired results.
- Unless an exercise specifies that you need to modify the HTML or CSS, you won't have to do that.
- Make sure every application is coded in strict mode.
- If you are doing an exercise in class with a time limit set by your instructor, do as much as you can in the time limit.
- Feel free to copy and paste code from the book applications or exercises that you've already done.
- Use your book as a guide to coding.

Extra 4-2 Develop the Change Calculator

In this exercise, you'll develop an application that tells how many quarters, dimes, nickels, and pennies are needed to make change for any amount of change from 0 through 99 cents. One way to get the results is to use the divide and modulus operators along with the parseInt() method for truncating the results so they are whole numbers.



Change Calculator

Enter amount of change due (0-99): 67

Quarters: 2

Dimes: 1

Nickels: 1

Pennies: 2

1. Download the zip file on canvas. It will contain an html file already linked to a css file and js file that are also provided. Make sure to save them all in the same folder.

   Then, run the application to see the user interface shown above, although that interface won't do anything until you develop the JavaScript for it.

2. In the JavaScript file, note that the $() function has already been coded.

3. Code an event handler named processEntry() that gets the user's entry and checks to make sure that it is a number between 0 and 99. If so, call a function named makeChange() and pass it the user's entry. Otherwise, display an alert dialog box for the error.

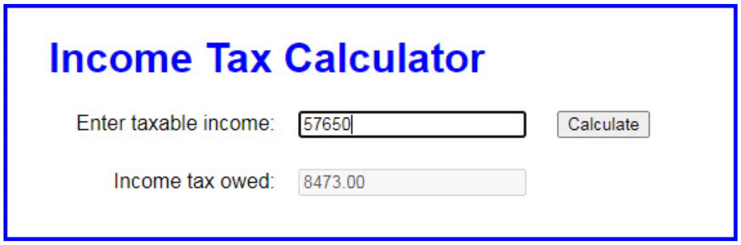
4. Code the makeChange() function, which should have one parameter that accepts the user's entry. This function shouldn't return anything, but it should display the results in the text boxes for Quarters, Dimes, Nickels, and Pennies.

5. Code a DOMContentLoaded event handler that attaches the processEntry() event handler to the click event of the Make Change button. Then, test this application.

6. Once all your files are completed you will upload them to a Github Pages repository and submit a link to your index page so I can test your application. Make sure you test it before submitting the link.

Extra 4-3 Develop the Income Tax Calculator

In this exercise, you'll use nested if statements and arithmetic expressions to calculate the federal income tax that is owed for a taxable income amount.



Income Tax Calculator

Enter taxable income: 57650

Income tax owed: 8473.00

This is the 2020 table for the federal income tax on individuals that you should use for calculating the tax:

Taxable income		**Income tax**	
Over…	**But not over…**		**Of excess over…**
$0	$9,875	$0 plus 10%	$0
$9,875	$40,125	$987.50 plus 12%	$9,875
$40,125	$85,525	$4,617.50 plus 22%	$40,125
$85,525	$163,300	$14,605.50 plus 24%	$85,525
$163,300	$207,350	$33,271.50 plus 32%	$163,300
$207,350	$518,400	$47,367.50 plus 35%	$207,350
$518,400		$156,235.00 plus 37%	$518,400

1. Download the zip file on canvas. It will contain an html file already linked to a css file and js file that are also provided. Make sure to save them all in the same folder.

   Note that the JavaScript file has some starting JavaScript code for this application, including the $() function and a DOMContentLoaded event handler that attaches a function named processEntry() to the click event of the Calculate button and moves the focus to the first text box.

2. Code the processEntry() function. It should get the user's entry and make sure it's a valid number greater than zero. If it isn't valid, it should display an error message. If it is valid, it should pass the value to a function named calculateTax(), which should return the tax amount. That amount should then be displayed in the second text box. The focus should be moved to the first text box whether or not the entry is valid.

3. Code the calculateTax() function. To start, just write the code for calculating the tax for any amount within the first two brackets in the table above. The user's entry should be converted to an integer, and the tax should be rounded to two decimal places. To test this, use income values of 9875 and 40125, which should display taxable amounts of 987.50 and 4,617.50.

4. Add the JavaScript code for the next tax bracket. (use else if statements to add the tax brackets)Then, add the JavaScript code for the remaining tax brackets. Test it as you go.

5. Once all your files are completed you will upload them to a Github Pages repository and submit a link to your index page so I can test your application. Make sure you test it before submitting the link.