Hands-On Project 2-1

In this project you will create an application to convert temperature readings between Fahrenheit and Celsius and between Celsius and Fahrenheit.

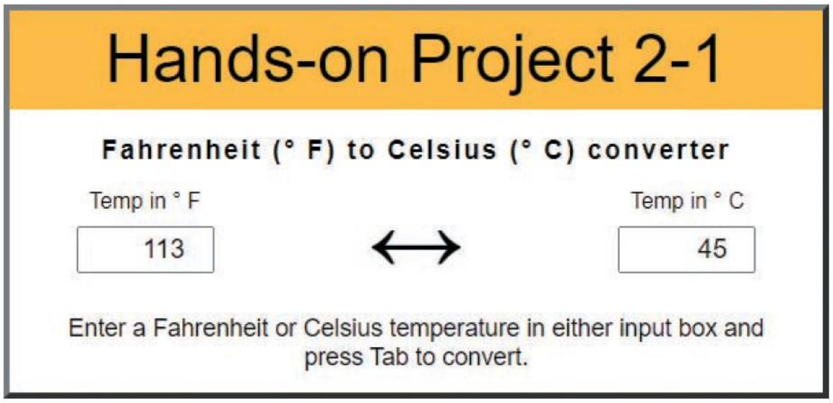
The formula to convert a Fahrenheit temperature to the Celsius scale is

Celsius 5 (*Fahrenheit* 2 32)/1.8

and the formula to convert a Celsius temperature to the Fahrenheit scale is

Fahrenheit 5 *Celsius* 3 1.8 1 32

Users will enter a value in a Celsius or Fahrenheit input box, press the Tab key and have the other input box automatically show the temperature reading in the other scale. A preview of the completed page is shown in **Figure 2-30**.



Hands-on Project 2-1

Fahrenheit (° F) to Celsius (° C) converter

Temp in ° F

113

↔

Temp in ° C

45

Enter a Fahrenheit or Celsius temperature in either input box and press Tab to convert.

Figure 2-30 Completed Project 2-1

Do the following:

1. Use your code editor to open the **project02-01_txt.html** and **project02-01_txt.js** files from the **HandsOnProject_02** folder. Enter your name and the date in the comment section of each file and save them as **project02-01.html** and **project02-01.js**, respectively.

2. Go to the **project02-01.html** file in your code editor and in the head section add a `script` element to load the project02-01.js file. Include the defer attribute to `defer` loading the external script file until the entire page is loaded. Study the contents of the HTML file and then save your changes.

3. Go to the **project02-01.js** file in your code editor. Create a function named `FahrenheitToCelsius()` containing a single parameter named `degree`. Insert a statement that returns the value of `degree` minus 32 and then divided by 1.8.

4. Create a function named `CelsiusToFahrenheit()` containing a single parameter named `degree`. Insert a statement that returns the value of `degree` multiplied by 1.8 plus 32.

5. Add an `onchange` event handler to the element with the id “cValue”. Attach an anonymous function to the event handler and within the anonymous function do the following:
   a. Declare a variable named `cDegree` equal to the value of the element with the id “cValue”.
   b. Set the value of the element with the id “fValue” to the value returned by the `CelsiusToFarenheit()` function using `cDegree` as the parameter value.

6. Add an `onchange` event handler to the element with the id “fValue”. Attach an anonymous function to the event handler and within the anonymous function do the following:
   a. Declare a variable named `fDegree` equal to the value of the element with the id “fValue”.

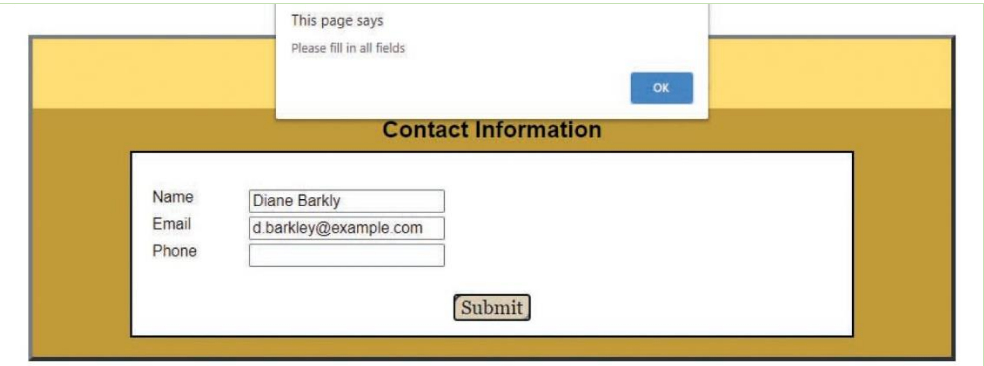
b. Set the value of the element with the id “cValue” to the value returned by the `FarenheitToCelsius()` function using `fDegree` as the parameter value.

7. Save your changes to the file.

8. Open **project02-01.html** in your web browser. Verify that when you enter **45** in the Temp in °C box and press Tab a value of 113 appears in the Temp in °F box. Verify that when you enter **59** in the Temp in °F box and press Tab a value of 15 appears in the Temp in °C box.

Hands-On Project 2-2

In this project you will create an application that tests whether all fields within a web form have been completed. In creating this application, you will take advantage of the fact that empty text strings are falsy and in a conditional operator will be treated as having the Boolean value `false` and non-empty strings are treated as truthy with a Boolean value of `true`. If any of the form fields is left empty, the application will display the alert box shown in **Figure 2-31** when the Submit button is clicked, otherwise an alert box with the message “Thank you!” is displayed.



This page says

Please fill in all fields

Contact Information

Name	Diane Barkly
Email	d.barkley@example.com
Phone	

Figure 2-31 Completed Project 2-2

Do the following:

1. Use your code editor to open the project02-02_txt.html and project02-02_txt.js files from the **HandsOnProject_02** folder. Enter your name and the date in the comment section of each file and save them as **project02-02.html** and **project02-02.js**, respectively.

2. Go to the **project02-02.html** file in your code editor and in the head section add a `script` element to load the project02-02.js file, deferring the loading the external script file until the entire page is loaded. Review the contents of the HTML file. Note that there are three input controls with the ids “name”, “email”, and “phone”. Each of these controls must be filled out for the form to be submitted.

3. Go to the **project02-02.js** file in your code editor. Create a function named `verifyForm()` with no parameters. Within the function do the following:
    a. Declare the `name` variable equal to the value of the input control with the id “name”.
    b. Declare the `email` variable equal to the value of the input control with the id “email”.
    c. Declare the `phone` variable equal to the value of the input control with the id “phone”.
    d. Insert a conditional operator that tests the truthy or falsy value of and `name` and `email` and `phone` using the && operator. If the result of this conditional expression is true, use the `window.alert()` method to display the message “Thank you!”, otherwise display the message “Please fill in all fields”.

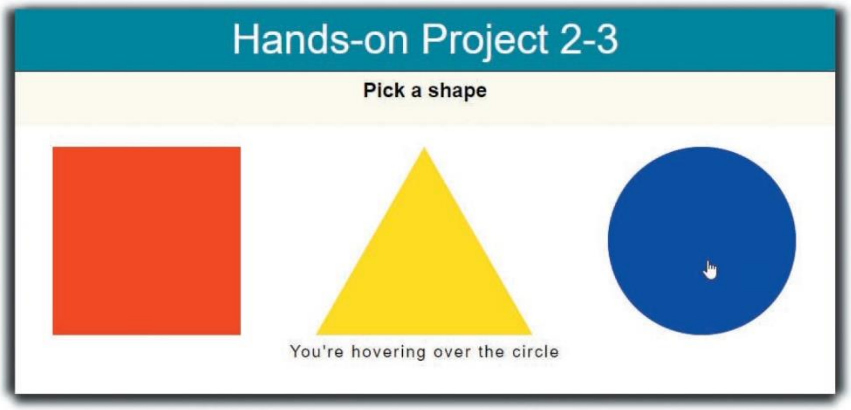
4. Below the `verifyForm()` function insert a statement that attaches an event listener to the page element with the id “submit”. When the click event occurs for this element, run the `verifyForm()` function.

5. Save your changes to the file and then open **project02-02.html** in your web browser.

6. Test the web form by clicking the Submit button with one or all the fields left blank, verifying that an alert box with the message “Please fill in all fields” is displayed. Enter text in all the fields and click the Submit button, verifying that an alert box with the message “Thank you!” is displayed.

Hands-On Project 2-3

In this project you will create an application that responds to the movements of the mouse over and out of a page object. The event that triggers the mouse over the object is called `mouseover` while moving the mouse out from an object triggers the `mouseout` event. In this application you will display a different message depending on the shape the mouse is hovering over or no message at all if the mouse is hovering over no shape. **Figure 2-32** shows a preview of the page with the message for the mouse hovering over the circle.



Hands-on Project 2-3

Pick a shape

You're hovering over the circle

Figure 2-32 Completed Project 2-3

Do the following:

1. Use your code editor to open the **project02-03_txt.html** and **project02-03_txt.js** files from the **HandsOnProject_02** folder. Enter your name and the date in the comment section of each file and save them as **project02-03.html** and **project02-03.js**, respectively.

2. Go to the **project02-03.html** file in your code editor and in the head section add a `script` element to load the project02-03.js file, deferring the loading the external script file until the entire page is loaded. Review the contents of the HTML file and note that the three shapes are placed within `div` elements with the ids “square”, “triangle”, and “shape”. There is also an empty paragraph with the id “feedback”. Save your changes to the file.

3. Go to the **project02-03.js** file in your code editor. Attach an `onmouseover` event handler to the element with the id “square”. In response to the event run an anonymous function containing a statement that changes the `innerHTML` property of the element with the id “feedback” to the text string “You ‘re hovering over the square”.

4. Attach an `onmouseout` event handler to the element with the id “square”. In response to the event run an anonymous function containing a command that changes the `innerHTML` property of the element with the id “feedback” to an empty text string.

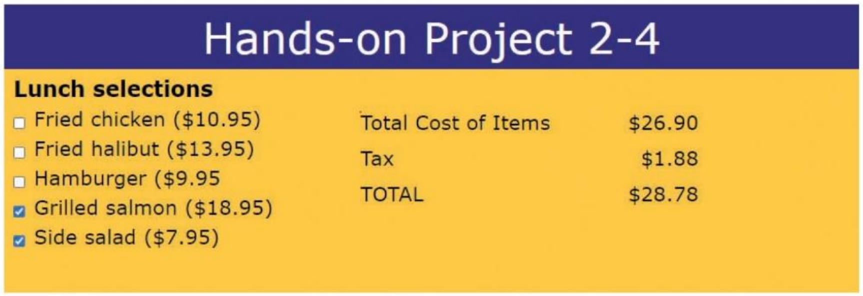
5. Repeat Steps 3 and 4 for the element with the id “triangle”.

6. Repeat Steps 3 and 4 for the element with the id “circle”.

7. Save your changes to the file and then open **project02-03.html** in your browser. Verify that as you hover your mouse pointer over each shape, a message indicating the shape in displayed on the page and when you move your mouse pointer away from the shape the message disappears.

Hands-On Project 2-4

In this project you will calculate the cost plus the tax of ordering items from a restaurant's online menu. So that the currency values are displayed with a leading `$` character, and to two decimal places you will call a function created for you named `formatCurrency()`, which takes a number and returns a text string in the format `$##.##`. The completed project appears as shown in **Figure 2-33**.



Hands-on Project 2-4

Lunch selections

- □ Fried chicken ($10.95)
- □ Fried halibut ($13.95)
- □ Hamburger ($9.95
- ☑ Grilled salmon ($18.95)
- ☑ Side salad ($7.95)

Total Cost of Items $26.90
Tax $1.88
TOTAL $28.78

Figure 2-33 Completed Project 2-4

Do the following:

1. Use your code editor to open the **project02-04_txt.html** and **project02-04_txt.js** files from the **HandsOnProject_02** folder. Enter your name and the date in the comment section of each file and save them as **project02-04.html** and **project02-04.js**, respectively.

2. Go to the **project02-04.html** file in your code editor and in the head section add a `script` element to load the project02-04.js file, deferring the loading the external script file until the entire page is loaded. Review the contents of the HTML file, noting the ids of different page elements. You will display the calculated values in `span` elements with ids of "foodTotal", "foodTax", and "totalBill". Save your changes to the file.

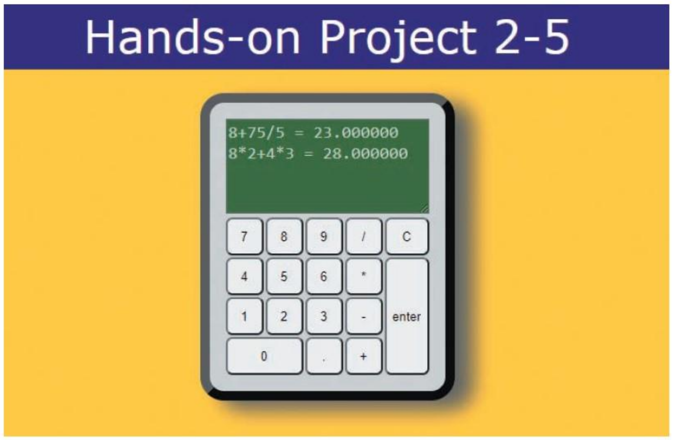
3. Go to the **project02-04.js** file in your code editor. Below the comment section, declare the following constants with their initial values: CHICKEN_PRICE 5 10.95, HALIBUT_PRICE 5 13.95, BURGER_PRICE 5 9.95, SALMON_PRICE 5 18.95, SALAD_PRICE 5 7.95, and SALES_TAX 5 0.07.

4. Create the `calcTotal()` function containing the following:
   a. Declare the `cost` variable with an initial value of 0.
   b. Declare the `buyChicken` variable equal to the checked property of the element with the id "chicken". In the same way, declare the `buyHalibut`, `buyBurger`, `buySalmon`, and `buySalad` variables equal to the `checked` property of elements with ids of "halibut", "burger", "salmon", and "salad", respectively.
   c. Use a comparison operator to increase the value of the `cost` variable by the value of the CHICKEN_PRICE constant if `buyChicken` is `true` or by 0 if otherwise (see **Figure 2-21** as an example of your code). Do the same for the `buyHalibut`, `buyBurger`, `buySalmon`, and `buySalad` variables, increasing the value of total cost by the value of HALIBUT_PRICE, BURGER_PRICE, SALMON_PRICE, and SALAD_PRICE, respectively.
   d. Set the `innerHTML` property for the element with the id "foodTotal" to the value returned by the `formatCurrency()` function using `cost` as the parameter value.
   e. Declare the `tax` variable, setting its value equal to the `cost` variable multiplied by SALES_TAX.
   f. Set the `innerHTML` property for the element with the id "foodTax" to the value returned by the `formatCurrency()` function using `tax` as the parameter value.
   g. Declare the `totalCost` variable, setting its value equal to the `cost` variable plus the `tax` variable.
   h. Set the `innerHTML` property for the element with the id "totalBill" to the value returned by the `formatCurrency()` function using `totalCost` as the parameter value.

5. Directly above the `calcTotal()` function, insert an event handler that runs the `calcTotal()` function when the element with id “chicken” is clicked. Repeat this for the elements with the id “halibut”, “burger”, “salmon”, and “salad”.
6. Save your changes to the file and then open **project02-04.html** in your web browser. Verify that when you click each of the menu items the calculated cost and tax is automatically updated to reflect your choices.

Hands-On Project 2-5

In this debugging challenge you will fix mistakes in code for an online calculator. The code has already been written for you but there are several syntax mistakes you will have to locate and correct. You can use your browser’s debugging console to assist you in locating the errors. When the code has been fixed, you will be able to run the online calculator shown in **Figure 2-34** by clicking the calculator buttons and viewing the results in the calculator window. To erase the contents of the window, click the C button.



Hands-on Project 2-5

```
8+75/5 = 23.000000
8*2+4*3 = 28.000000
```

Figure 2-34 Completed Project 2-5

Do the following:

1. Use your code editor to open the **project02-05_txt.html** and **project02-05_txt.js** files from the **HandsOnProject_02** folder. Enter your name and the date in the comment section of each file and save them as **project02-05.html** and **project02-05.js**, respectively.

2. Go to the **project02-05.html** file in your code editor and in the head section add a `script` element to load the project02-04.js file, deferring the loading the external script file until the entire page is loaded. Review the contents of the HTML file. Notice that the calculator buttons are arranged in a web table with each calculator button having a separate id related to the button’s value. Save your changes to the file.

3. Go to the **project02-05.js** file in your code editor. The first part of the code contains several event handlers for running functions in response to the `click` event (in future chapters you will learn a more efficient way of specifying these event handlers). Within this section there are four syntax errors. Locate and fix those errors.

4. The next section in the file contains the `runCalculator()` function used to edit the contents of the calculator window in response to the clicking of calculator buttons. The `calcValue` variable will be used to store the text string of the expression in the calculator window. There are two syntax errors in this function. Fix them both.

5. The next section contains the `clearCalculator()` function to clear contents of the calculator window. There is one syntax error in this function. Locate and fix the error.

6. Save your changes to the file and then open **project02-05.html** in your browser. Test your calculator by clicking the calculator buttons, verifying that you can enter expressions into the calculator window and

evaluate those expressions by clicking the Enter button. Also verify that you can clear the calculator window by clicking the C button. If the online calculator does not work correctly, use the browser console to locate and fix any undiscovered errors.

NOTE

1. **Use the W3C Markup Validation Service to validate the index.html document, and then, if necessary, fix any errors that the document contains.**

2. **Make sure you organize the content of the project in different folders: images, styles, and scripts folders.**