ASSIGNMENT 6

Version 21/FA

MAPPINGS:

The following course objectives and/or outcomes are measured in this assignment:

COURSE OBJECTIVES

- 6B: Use the equality (i.e., == and !=) and identity (i.e., === and !==) operators in your applications.
- 6C: Use switch statements, including those that use fall through and default cases.
- 6D: Use the conditional (ternary) operator for simple logic requirements.
- 6E: Use try-catch statements to catch errors.
- 6F: Create and throw Error objects.

COURSE OUTCOMES

- 1. Use JavaScript to create interactive web applications.
- 2. Write clean, consistent code.

Continues . . .

GRADING:

Task	Points	Criteria
Part 1		
1.1 Setup		
6	5	Each line of the standard opening comment is worth 1 point. Subtract 1 point per missing item.
7	5	Pass/Fail
1.2 Test the Application		
There are no points in this section of the assignment.		
1.3 Define Reset		
3	2	Pass/Fail
1.4 Check for Zero		
2a.	5	3 points for writing else-if clause (deduct for errors) 2 points for use of identity operator (pass/fail)
2b	3	2 points for displaying error message. 1 point for return statement.
1.5 Switch it Up		
2	18	1.5 points for each case statement (8 total statements; 12 total points): - 1 point for each correctly written statement (deduct for fall through cases) - .5 points for use of correct formulas 2 points for correctly written switch statement 4 points for outputting values (each selection should output a specific conversion message). - .5 points for each output value
1.6 Try to Catch Something		
2	4	Pass/Fail
3a	4	Pass/Fail
3b	2	Deduct if message property not used.
3c	2	Deduct 1 point per missing item, if applicable.
1.7 Throw Me Something		
5	2	Pass/Fail
6	4	Deduct if this is done incorrectly or is missing components (instructor discretion)
1.8 What in Ternary-Nation		
2	6	Instructor discretion.
1.9 Cleanup		
There are no points in this section of the assignment.		
Total	**62**	

Penalties
Deduct 50% from entire assignment for the use of the `var` keyword in variable declarations.
Deduct 30% from the entire assignment is the solution does not load and/or if errors appear in the console that are not generated by explicit methods of the console object (in other words, errors that have not been troubleshot and resolved by the student prior to submission).
Deduct 60% from the entire assignment if JavaScript is inline/embedded instead of external.
Deduct a maximum of 25% for failure to use jQuery methods. This is at the instructor's discretion and may be less than 25% if minor errors/oversight on the student's part.
Deduct a maximum of 12.5% for code that does not comply with the course *Style Guide* and/or which is messy/unorganized, uncommented, or missing semicolons.

Continues…

GENERAL NOTE:

You must use jQuery methods to interact with the DOM.

TASK:

MODIFY THE APPLICATION

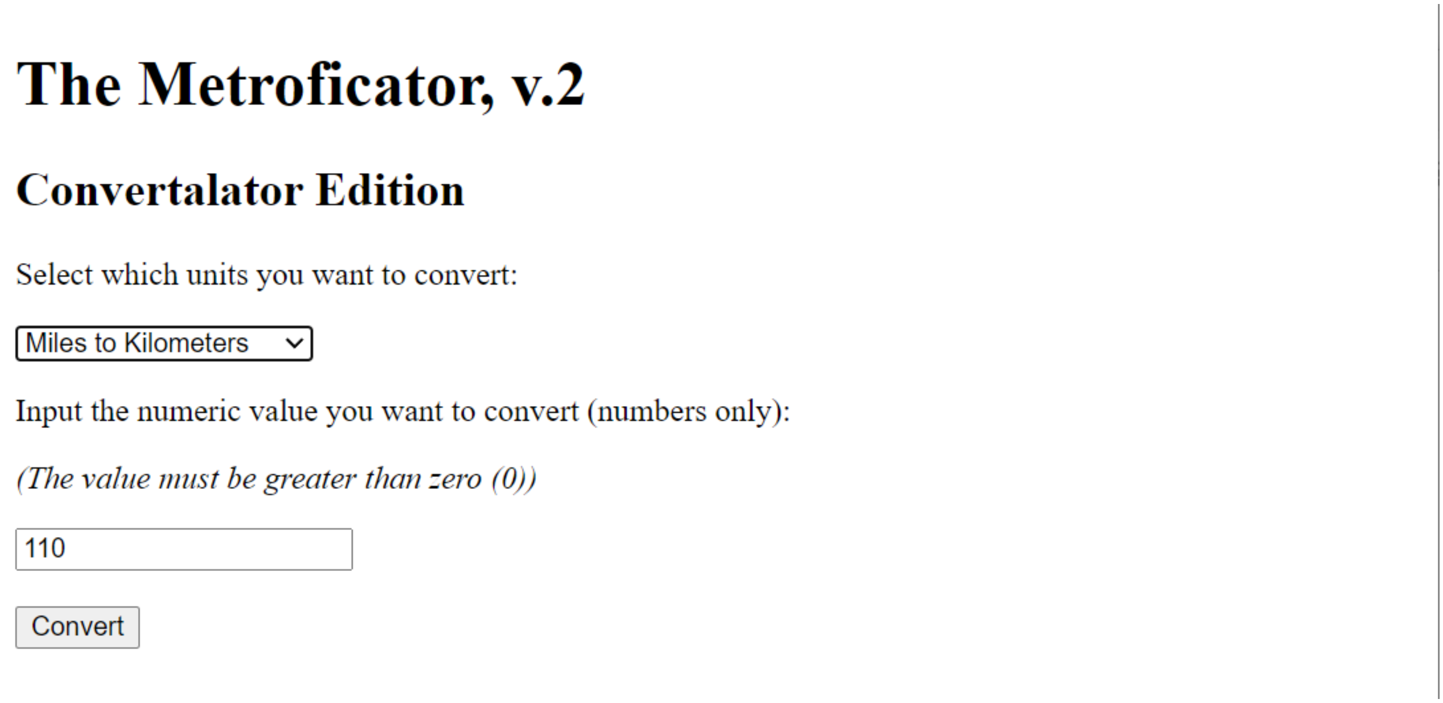
1.1 SETUP

1. Review the course *JavaScript Style Guide* before starting this assignment. Part of the assignment will be graded on your adherence to the *Style Guide*.
2. Download **assignment06_starter.zip** from the *Module 6: Assignment* drop box in Canvas. The file is located beneath the heading *Assignment Resources*.
3. Extract **assignment06_starter.zip.** The file contains two files and one subfolder:
   a. a single HTML document named *index.html*.
   b. a single JavaScript file named *script.js*.
4. **script.js** has already been linked to **index.html** for you.
5. Open **script.js**.
6. Add the **Standard Opening Comment** to the top of the script **(5 points**).
7. Add "use strict" beneath the **Standard Opening Comment**. **(5 points)**.

Continues . . .

1.2 TEST THE APPLICATION

1. Double-click **index.html** to launch it in your browser; examine the application’s behavior.
   a. The page will open, and the content shown in Figure 1 should appear.
   b. Input **0** (zero) into the value field to convert and click **Convert**; nothing should happen.



The Metroficator, v.2

Convertalator Edition

Select which units you want to convert:

Miles to Kilometers

Input the numeric value you want to convert (numbers only):

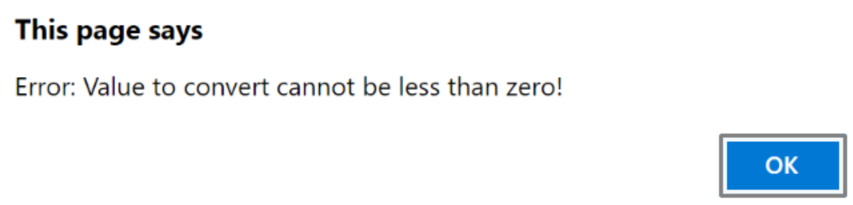
(The value must be greater than zero (0))

110

Convert

Figure 1: Starter version of index.html

   c. Input **-50** (negative 50) into the value field to convert and click **Convert**; an error should appear | (Figure 2).



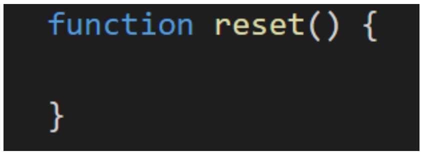
This page says

Error: Value to convert cannot be less than zero!

Continues…

1.3 Define Reset

1. Open **script.js**
2. Find the stub for the **reset()** method (Figure 2).



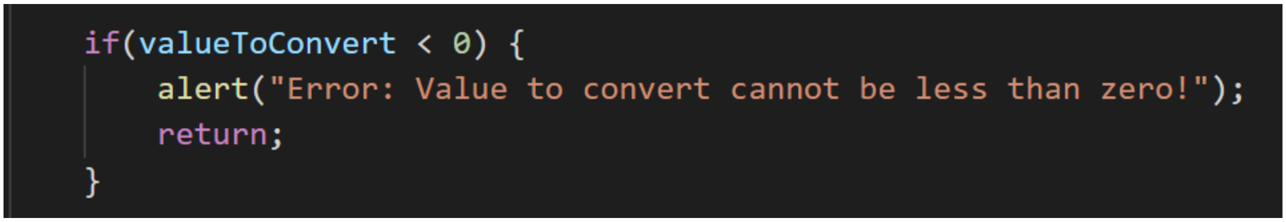
```
function reset() {

}
```

Figure 2

3. The **reset()** method should set the value in the **valueHolder** input back to **0** (zero). (**2 points).**

1.4 Check for Zero

1. Find the block of code illustrated in Figure 3 (below).



```
if(valueToConvert < 0) {
    alert("Error: Value to convert cannot be less than zero!");
    return;
}
```

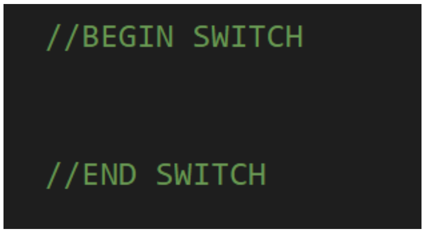
Figure 3

2. Modify the block of code shown in Figure 3:
   a. Add an **else-if** clause to the **if** statement; use the **identity** operator to check if the user input **0** (zero)**. (5 points)**
   b. If the user input zero, display an error message indicating that the value to convert must be greater than zero and invoke the **reset()** method, then exit the **else-if** by using a **return** statement. **(3 points)**

Continues…

1.5 Switch It Up

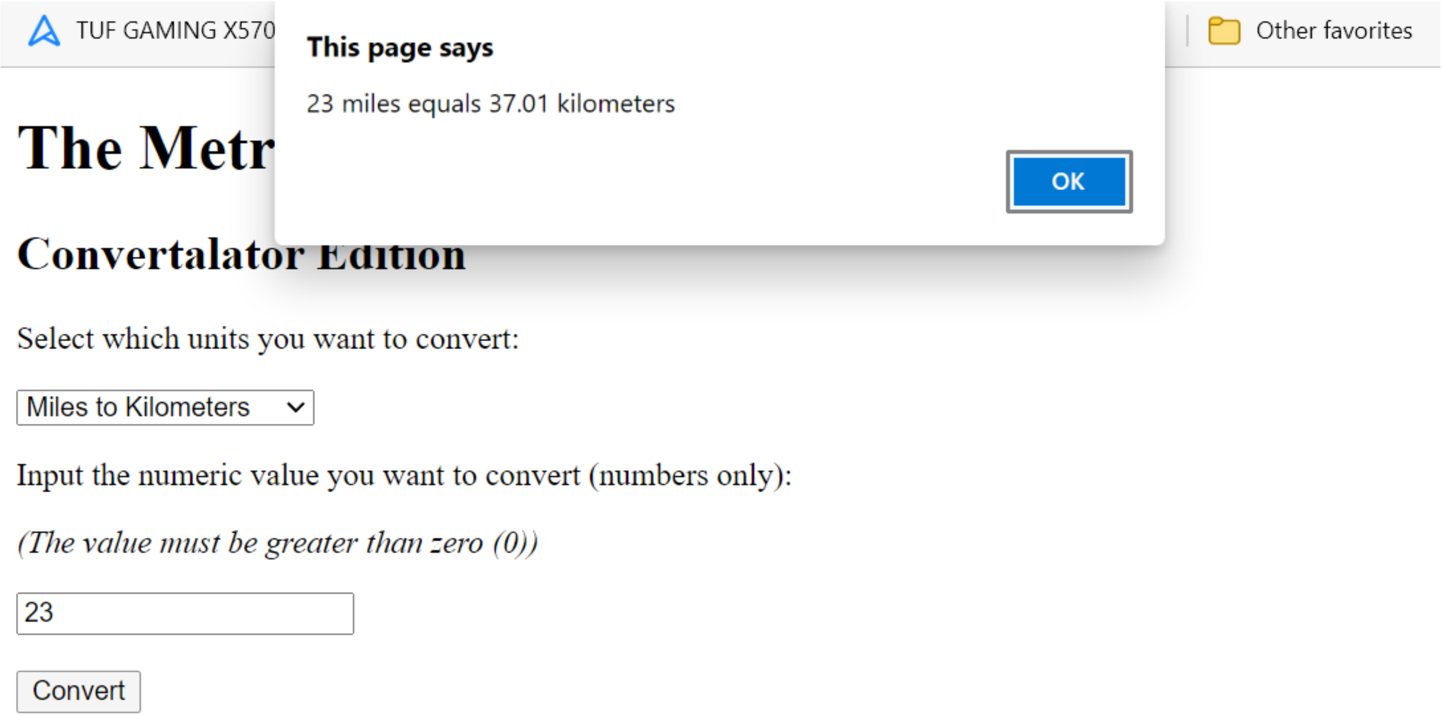
1. Find the comments shown in Figure 4 (below).



```
//BEGIN SWITCH


//END SWITCH
```

Figure 4

2. Within the comments shown in Figure 4, write a switch statement that switches **selectionOptionValue** to perform a conversion calculation based on the value selected in the dropdown with the ID value **conversionSelector**. The application should display a dialog informing the user that the conversion the first unit is equal to the converted value and units (an example is shown in Figure 5). The formulas to perform the conversions are listed in *Appendix A*, which is on page 10 of this writeup. **(36 points)**
   a. None of the cases of the switch statement shall be a fall through case.



This page says

23 miles equals 37.01 kilometers

The Metr

Convertalator Edition

Select which units you want to convert:

Miles to Kilometers

Input the numeric value you want to convert (numbers only):

(The value must be greater than zero (0))

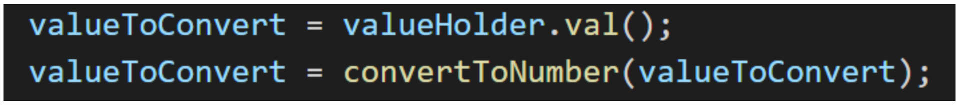
23

Figure 5

Continues . . .

1.6 Try to Catch Something

1. Find the block of code shown in Figure 7.



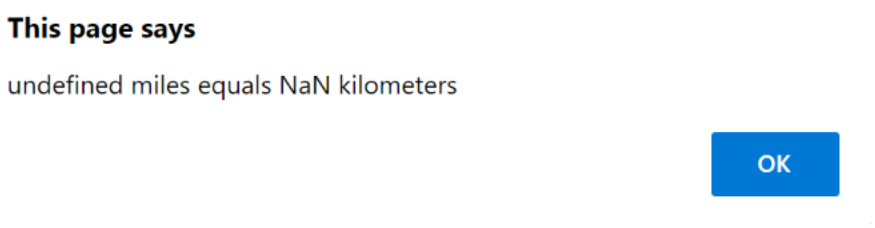
```
valueToConvert = valueHolder.val();
valueToConvert = convertToNumber(valueToConvert);
```

Figure 6

2. Wrap the block of code shown in Figure 7 in a **try** block. (**4 points**).
3. Write a **catch** block:
   a. In the **catch** clock, capture the Error object **(4 points)**
   b. Display an alert, which shows the **message** property of the Error object **(2 points**)
   c. Then, invoke the **reset()** method and exit using the **return** statement (**2 points)**

1.7 Throw Me Something

1. Save all your changes and reload **index.html** in your browser.
2. Enter a non-numeric value into the text field and click **Convert**. The app should display a wonky error message similar to what's shown in Figure 7.



This page says

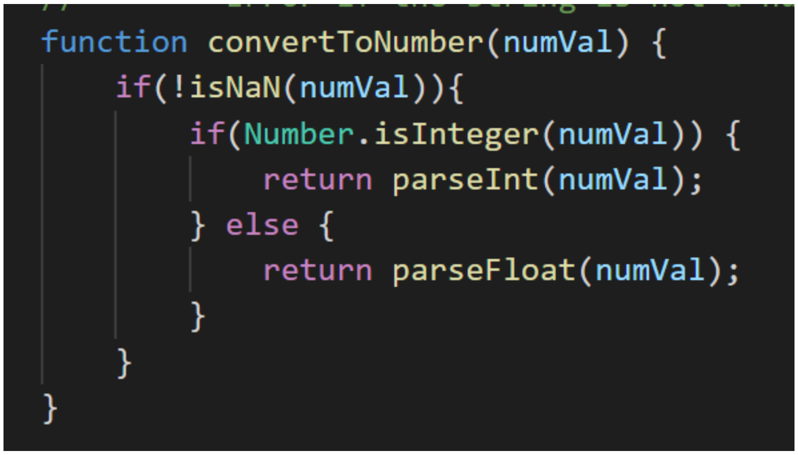
undefined miles equals NaN kilometers

Figure 7

3. Let's clean this up.

Continues…

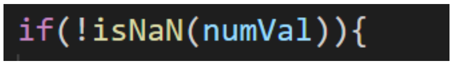
4. Find the code block shown in Figure 7.



```
function convertToNumber(numVal) {
    if(!isNaN(numVal)){
        if(Number.isInteger(numVal)) {
            return parseInt(numVal);
        } else {
            return parseFloat(numVal);
        }
    }
}
```

Figure 8

5. Add an **else** clause to the **if** statement shown in Figure 8 (**2 points**).



```
if(!isNaN(numVal)){
```

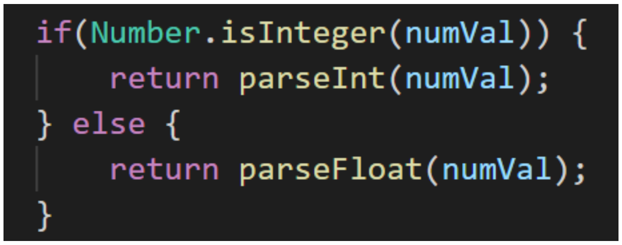
Figure 9

6. In the body of the **else** clause, throw an error, which contains a message indicating the value is not a numeric value. (**4 points)**.
7. Save all your changes and reload **index.html** in your browser.
8. Enter a non-numeric value into the text field and click **Convert**. The app should display the text of message in the Error you throw.

Continues…

1.8 What in Ternary-Nation?

1. Find the block of code shown in Figure 10.



```
if(Number.isInteger(numVal)) {
    return parseInt(numVal);
} else {
    return parseFloat(numVal);
}
```

Figure 10

2. Comment-out the entire block of code shown in Figure 10. Then, rewrite the statement using a **ternary operator.** (**6 points).**

1.9 Cleanup

1. Test your application. If necessary, use the browser’s developer tools to troubleshoot any errors you may have.
2. Comment and clean up your code:
    a. Make sure to document what functions and blocks of code do.
    b. Ensure consistent alignment, spacing, and carriage returns.
    c. Make sure to remove non-used code; comments should describe your code only.

Submission

When complete, create a single ZIP file containing your solution for this assignment. The ZIP file should contain all files included in the original starter code or added as part of this assignment.

Attach and upload the ZIP file to Assignment 6 and submit.

NOTE: Canvas is configured to only accept ZIP files, DOC files, and DOCX files; it will not accept ZIPx, 7ZIP, pZip, RAR, etc.

End Assignment.

APPENDIX A

Conversion	Formula
Miles to Kilometers	km = (n x 1.609344)
Kilometers to Miles	m = (n x 0.62137)
Pounds to Kilograms	kg = (n * 0.45359237)
Kilograms to Pounds	lb = (n / 0.45359237)
Feet to Meters	meters = (n * 0.3048)
Meters to Feet	feet = (n / 0.3048)
Celsius to Fahrenheit	Fahrenheit = (n * 1.8) + 32
Fahrenheit to Celsius	Celsius = (n - 32) / 1.8

NOTE: *n* is the value the user has input.