ASSIGNMENT 2

Version 21/FA

MAPPINGS:

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

COURSE OBJECTIVES

- 1G: Code, test, and debug an application that uses basic control statements.

COURSE OUTCOMES

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

GRADING:

Task	Points	Criteria
1	2	Pass/Fail
3	5	Each line of the standard opening comment is worth 1 point. Subtract 1 point per missing item.
5	5	Pass/Fail
6	7	Each change is worth 1 point, award 2 points for each correctly labeled constant.
8	2	Pass/Fail
9	5	Pass/Fail
11	5	Pass/Fail
13	2	Pass/Fail
15	2	Pass/Fail
16	2	2 points if the student uses the correct identifier; 1 point if the student does not use the correct identifier.
17a	2	Pass/Fail
17b	2	Pass/Fail
17c.i	2	Pass/Fail
17c.ii	2	Pass/Fail
Total	**45**	

Penalties
Deduct 50% from entire assignment for the use of the `var` keyword in variable declarations.
Deduct 60% from the entire assignment if JavaScript is inline/embedded instead of external.
Deduct a maximum of 2.5% for code that does not comply with the course *Style Guide* and/or which is messy/unorganized, uncommented, or missing semicolons.

TASK:

TEST THE APPLICATION

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 **assignment02_starter.zip** from the *Module 2: Assignment* drop box in Canvas. The file is located beneath the heading *Assignment Resources*.
3. Extract **assignment02_starter.zip.** The file contains two files:
    a. a single HTML document named *index.html*.
    b. a single JavaScript file named *script.js*.
4. Double-click **index.html** to launch it in your browser; examine the application's behavior. Nothing will happen.

MODIFY THE APPLICATION

1. Open **index.html** and link to the **script.js** JavaScript file. (**2 points**)
2. Reload **index.html** in your browser. You should now be prompted to input a weight in pounds. Input a numeric value (e.g., 100) and press enter or click **OK**. The application should display a dialog reporting cargo weight and average item weight (if you input 100, the cargo weight and the average item weight should both be 100).
3. Add the **Standard Opening Comment** to the top of the script **(5 points**).
4. Add "use strict" to the very top of the JavaScript file. Open the JavaScript file. **(5 points)**
5. Double-click **index.html** to launch it in your browser; examine the application's behavior. Note how you are prompted to enter your weight in pounds, then the application stops. This is because "use strict" requires we declare our variables in our code.
6. Modify the application so that each variable identifier shown in Figure 1 is declared using a `let` or a `const` keyword as appropriate. Remember, as stated in the course *JavaScript Style Guide*, "Use `const` by default unless a variable needs to be reassigned. The `var` keyword must not be used." You must use a mixture of `let` and `const` as appropriate **(7 points)**.

```
cargoWeight = [];
maxWeightLbs = 1000;
totalCargoWeight = 0;
itemWeight = 0;
average = 0;
```

Figure 1

7. Find the block of code shown in Figure 2.

```
//add your loop around this
itemWeight = parseFloat(prompt("Please enter the item weight in pounds"));
if(itemWeight >= 0) {
    cargoWeight[cargoWeight.length] = itemWeight;
} else if (itemWeight != -1) {
    alert("Item weight must be a valid number that is greater than zero pounds!");
}
```

Figure 2

8. Modify the prompt "`Please enter the item weight in pounds`" so that it outputs "`Please enter the item weight in pounds, or input -1 to exit.`" **(2 points)**.
9. Modify the block of code shown in Figure 2 to add a do-while statement around the block. The block of code shown in Figure 2 should loop until the user inputs the value **-1** (in other words, the loop will repeat unless the user inputs **-1**). **(5 points)**.
10. Find the block of code shown in Figure 3

```
if(cargoWeight.length > 0) {
    for (let i = 0; i < cargoWeight.length; i++) {
        totalCargoWeight = parseFloat(totalCargoWeight + cargoWeight[i]);
    }
    average = parseFloat(totalCargoWeight / cargoWeight.length);
}
```

Figure 3

11. Modify the for loop so that it is a for-of loop instead (**5 points**).
12. Find the statement shown in Figure 4.

```
alert("Total cargo weight is " + totalCargoWeight +"; average item weight is " + average);
```

Figure 4

13. Comment-out the statement shown in Figure 4. **(2 points)**.

14. Find the statement shown in Figure 5.

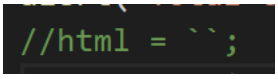

Figure 5

15. Uncomment-out the statement shown in Figure 5. **(2 points)**.
16. Place a `let` or a `const` keyword in front of the statement shown in Figure 5. **(2 points)**.
17. Use template literal syntax to accomplish the following:
    a. Output the total cargo weight, displaying 2 values after the decimal point **(2 points)**.
    b. Output the average item weight, displaying 2 values after the decimal point **(2 points)**.
    c. Based on the rocket weight display one of the following:
        i. If the total rocket weight is less than the maximum rocket weight, let the user know the rocket can take off. **(2 points)**
        ii. If the total rocket weight is equal to or greater than the maximum rocket weight, let the user know the rocket is too heavy to lift off. **(2 points)**

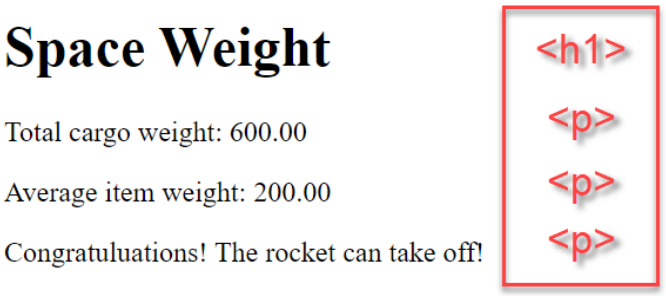

Figure 6

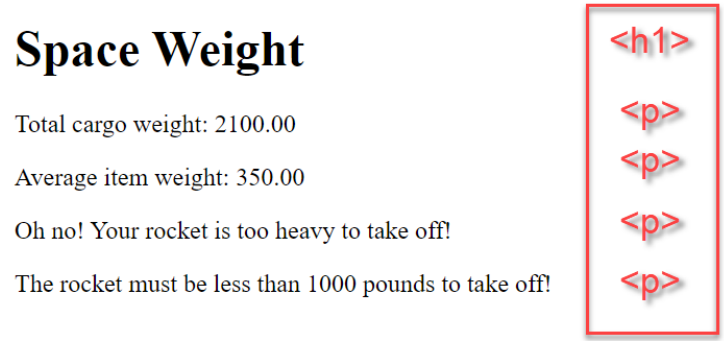

Figure 7

18. Figures 6 and 7 (above) illustrate the output you should have. The box to the right of each image shows the HTML elements you must use in your output.
19. Do not use the `var` keyword for declaring variables. This is grounds for an automatic 50% deduction.
20. Make sure your JavaScript is clean and organized. Ensure spacing is consistent and easy to read. Cleanliness and organization will be assessed beginning in Assignment 2. (2.5% potential penalty).
21. Add comments to your code to explain *your* understanding of what's going on. You must also document where you changed code to satisfy each of the assignment requirements. (2.5% potential penalty)

SUBMISSION

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

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

Upload and submit Assignment 2.

End Assignment.