JavaScript and jQuery Course
Instructor Teresa Pelkie
Assignment 2

Assignment Description

In this assignment, you will create the necessary variables to hold the values to perform the calculations as described below. You will also use the **Document Object Model** to write the results to the web page, as shown below or similar to it. **You may make modifications to the HTML and add CSS if you wish!**

How to Submit

You will submit an HTML document that includes the necessary JavaScript code in the `<head>` and `<body>` sections.

Name of the file: the file name **must be** `jsjqas02.html`

You **must include** an HTML comment in the file immediately after the `<!DOCTYPE html>` statement. The comment must include your name.

Sample Code

There is a sample HTML file available in the `JS-jQuery_Session_03_sampleCode.zip` file. You can use the HTML in that file as a "starter file" for the assignment. You can make modifications to the HTML and add CSS code, or you can just use the code in the file "as is".

File Name	Description
`jsjqas02_starter_file.html`	HTML file that contains the starting HTML for the assignment. You will add the JavaScript code that you develop to this page.

Sample Output

This is what your web page should look like. It is not important that it look exactly like this, you can change the formatting or apply CSS if you want to. However, you must show all of the values that are shown here and the values must be clearly labelled.

Details of the assignment

Scenario

Sees Candy is having an online Valentines Special. You may purchase a box of chocolates ranging from 1 to 10 pounds. The cost per pound depends on the number of pounds, and so does the shipping. Based on how many pounds the box is, you will calculate:

- the cost of the box
- the cost of shipping
- the total

Given Values

Include the following values in your code. These values should be defined as variables that are initialized, not hard-coded into the calculation statements.

Description	Initial Value	Notes
Box Price, 1 to 5 pounds	$20/pound	This is the cost per pound, based on the number of pounds. Cost example: a 4 pound box will cost $80 (4 pounds * $20/pound) Cost example: a 9 pound box will cost $135 (9 pounds * $15/pound)
Box Price, 6 to 9 pounds	$15/pound	
Box Price, 10 pounds	$10/pound	
Shipping Cost, 1 to 3 pounds	$5.00	The shipping cost is a "flat rate", based on the range of the box weight. Example: it costs $10.00 to ship a 6 pound box Example: it costs $20.00 to ship a 10 pound box
Shipping Cost, 4 to 7 pounds	$10.00	
Shipping Cost, 8 to 9 pounds	$15.00	
Shipping Cost, 10 pounds	$20.00	

What is to be written to the page

You must calculate and write these values to the page:

- the number of pounds ordered (see below)
- the cost of the box, based on the number of pounds
- the shipping cost, based on the number of pounds
- total amount of the order (cost + shipping)

Process

Follow these steps to develop the JavaScript statements to perform the required calculations.

- Generate a random number between 1 and 10. This will be the number of pounds ordered. Sample code to generate the random number is shown below. As an alternative, you can use the `prompt()` function to get the number of pounds from the user. If you use `prompt()`, you must check the value that is entered (it must be a value from 1 to 10).
- Based on that number and the cost per pound, determine the cost of the box of chocolate.
- Based on that number, determine the shipping amount. Shipping is a flat fee based on pounds.
- Add the computed cost amount and the shipping amount to give the total cost of the box.

Write the JavaScript code that performs the calculations in the `<head>` section of the page.

Use the `document.getElementById()` method (Document Object Model) to output the values to the web page.

You may add additional HTML and formatting if you wish. I am not grading your HTML layout, only the JavaScript.

How to generate a random number in the range 1-10

1. Use the `Math.random()` method to generate a random number between 0 and 1. Create a variable to hold the value of that number:

```
var myNumber = Math.random();
```

2. You need to generate a random number between 1 and 10, so multiply the random number times 10:

```
var myNumber = Math.random() * 10;
```

3. Now you need to round the number, which can be a decimal, to a whole number. You need to round up, so that you include the value of 10. You can use the `Math.ceil()` method:

```
myNumber = Math.ceil(myNumber);
```

So this is the code you need to start with (included in the starter file):

```
var myNumber = Math.random() * 10;    // generate a random number between 1 and 10
myNumber = Math.ceil(myNumber);       // round up to whole numbers including 10
```

Alternative method that is allowed in this assignment to get the number of pounds ordered

Instead of using the `Math.random()` method to generate the number of pounds ordered, you can use the `prompt()` function to ask the user to enter the number of pounds ordered. If you use `prompt()`, be sure that you check that the number of pounds that is entered is between 1 and 10.

Grading

This assignment is worth 100 points. You will be graded based on the following items.

Item	Points	Description

1. File name is correct	5	The name of the HTML file must be `jsjqas02.html`
		
		This is so that both you and I can clearly identify which assignment the file is for.
		
		If you submit the HTML file in a ZIP file, the name of the ZIP file must be `jsjqas02.zip`
		
		If you split the JavaScript and/or CSS into separate files, the files must be named:
		`jsjqas02.js`
		`jsjqas02.css`
		
		You are not required to split the code into separate files. It is preferable (for this course) that you include all of the code in the single HTML file.

2. File includes an HTML comment with your name	5	You must include an HTML comment with your full name immediately after the `<!DOCTYPE html>` statement. This is so that I can quickly identify who created the file. If you split the code into separate files (HTML, JS, CSS), you must include a comment at the beginning of each file with your name.
3. All variables are defined with the `var` keyword once	5	Although JavaScript allows you to initialize and use a variable without defining it with `var`, that technique can lead to problems. In this course, all variables must be defined with `var`. You only need to define a variable with `var` once, the first time it is used.
4. All variable names are meaningful names and accurately reflect their usage	10	Creating meaningful variable names is one of the most important parts of coding. In addition to a meaningful name, the variable name should accurately reflect the purpose of the variable. For example, a variable name of "sanDiego" is meaningful, but it is not accurate if it is used to represent the pounds ordered.
5. Calculations are clearly written and can be easily understood, are efficient and correct	15	There are many ways to write calculations in JavaScript. The goal is to have calculations that are correct and that can be easily understood. For this assignment, you will most likely use a series of `if/else if/else` statements, or you can use a `switch` block. There are three sets of calculations that you will perform: generate or prompt for the number of pounds ordered calculate the cost of the box based on the number of pounds calculate the shipping based on the number of pounds

Item	Points	Description
6. Correct and accurate results. Control structures are used logically	20	The calculations that you will perform are intended to provide specific results. Check your output. Before submitting the assignment, work through the calculations "by hand" (you can use a calculator to help check your work). Verify that the results of your JavaScript calculations are correct and accurate. If you use the `Math.random()` method to generate the number of pounds ordered, be sure you reload the page so that it generates all of the values between 1 and 10. You need to check to see that the code handles each number of pounds correctly. If you use the `prompt()` function, you can quickly test your code by entering each value (1 to 10).
7. DOM is used to write values to the page	10	You must use the `document.getElementById()` DOM method to write your calculated values to the page. You cannot use `document.write()`.
8. Code is well-formatted	10	Indent your JavaScript statements under the `<script>` tag. Usually, you will indent statements by 2 to 4 spaces. In this assignment, all of the statements will be intended by the same number of spaces. You should include a blank line as a separator between sections of code. For example, you can put a blank line after the variable initializations and before you start the calculations. Do not "mechanically" put a blank line after each statement unless it makes sense to do so. The idea of using "whitespace" (blank lines) in code is to provide visual separation of important sections of code. Be sure you are using a consistent formatting style for opening and closing brace ({ }) characters.
9. Assignments and operations are consistently expressed	10	When you write an expression using a math symbol (+, -, *, / or others) you may or may not put a blank before and after the operator. It is generally preferable to use a blank before and after. Similarly, when you use the assignment operator (the = sign), you can put a blank before and after or not. Again, it is generally preferable to put a leading and trailing blank. When you use relational operators in an `if` statement (for example, ==, <=, >=, <, >) you may or may not put a space before and after the operator. It is usually preferable to use a leading and trailing blank. Regardless of which option you choose (use blanks/do not use blanks), you must be consistent in your style.
10. All other code (HTML, CSS) is well-formatted	10	You can lay out the page any way that you want, using HTML and CSS, or you can use simple HTML code that just displays the results. Your HTML code should be well-formatted. Do not include extra blank statements where they are not needed, but use a blank statement to separate sections of the HTML. Be sure you have closing tags where they are required. If you use CSS, the CSS must be well-formatted. Be sure you have correct { and } characters in your CSS.