JavaScript Assignment #4

Grading Note: Grading will be based on whether your page does what it is supposed to do, but also that it is written properly with correct HTML 5, CSS, and JavaScript. If any of your assignment parts isn’t working right but you are turning it in anyway, be sure to indicate that it doesn’t work properly. I will discover this anyway, but you might save me a lot of time and keep me in a better grading mood!

**Part 1: Feeling Loopy**(Checking for bad input and re-prompting as needed – using A2P2 as a starting point)

Begin by making a copy of your A2P2 file and call it A4P1. To refresh your memory, A2P2 was called "Gimme Some Products" and used "prompts" to get the customer's name, the number of product #1 to purchase, and the number of product #2 to purchase. For this version of the script, you will be checking for bad input and using a "while" loop around each of the 3 prompts (3 loops total) so that the customer is re-prompted for any bad input they enter (but only for bad input, they should never have to re-enter good input). Here is what we will consider to be good (and bad) input that should result in re-prompting of invalid input and accepting of valid input:

**name**: input must be at least 1 character entered for the name otherwise re-prompt (a string with no characters in it is equal to the empty string "" and should be considered bad input)

**quantity #1**: must be a number in the range 0 to 1000 otherwise re-prompt (any input that "is not a number" should be considered bad input)

**quantity #2**: same as "quantity #1" above

To see the “while” loop in action doing some input error checking very similar to what we need to do here, use the link below to this while loop demo script. See how this script works in the browser and then examine its code.

<http://www.faculty.rcc.edu/mcleod/js/class_demos/loops/while.html>

After reviewing that script and understanding how the while loop can be used to error check prompt inputs, you should be in a position to start adding the necessary code to perform the error checking required. As was done in the demo script, if you encounter bad input you should re-prompt for the input by displaying the bad input and indicating there was a problem as you also ask for the to try again to enter good input (inside the while loop).

It is recommended that you not try to write the code for all 3 inputs at once. Work on coding one input at a time. When you get one working, test it thoroughly (with various good and bad input) before moving on to the code for the next input.

**What to turn in:**

No screenshot is needed -- you will just submit your code for this part.

**Code is needed --** into the given assignment 4 “turn it in” Word file, paste a copy of all of your web page’s code into the part 1 code section of the document. It should contain **ALL** of your page’s code. Do NOT paste a screenshot of code, paste the actual text of the code. **Make sure all of your code is visible** within the Word file's margins! You can format the font, spacing, or size of the code after pasting into Word to improve its appearance and readability, just be sure not to actually change any of the code itself.

**Part 2: While Writing A Function…**(Same as part 1 but using a function for the input validation)

Begin by making a copy of your A4P1 (this assignment's part 1) page and calling your new copy “a4p2.html”. In this part, your application will do the same thing as your part 1 application did (from the user's perspective running it in the browser.) But the solution must be coded differently. You must write one function to do the work of the input validation on the product quantity (number) inputs. This one function should be called twice, once for each product quantity input. The function should return the valid input that it gets from the user. (You can leave the code that does the name input as is.) Test your function to make sure the input validation works the same now as it did in part 1.

**What to turn in:**

No screenshot is needed -- you will just submit your code for this part.

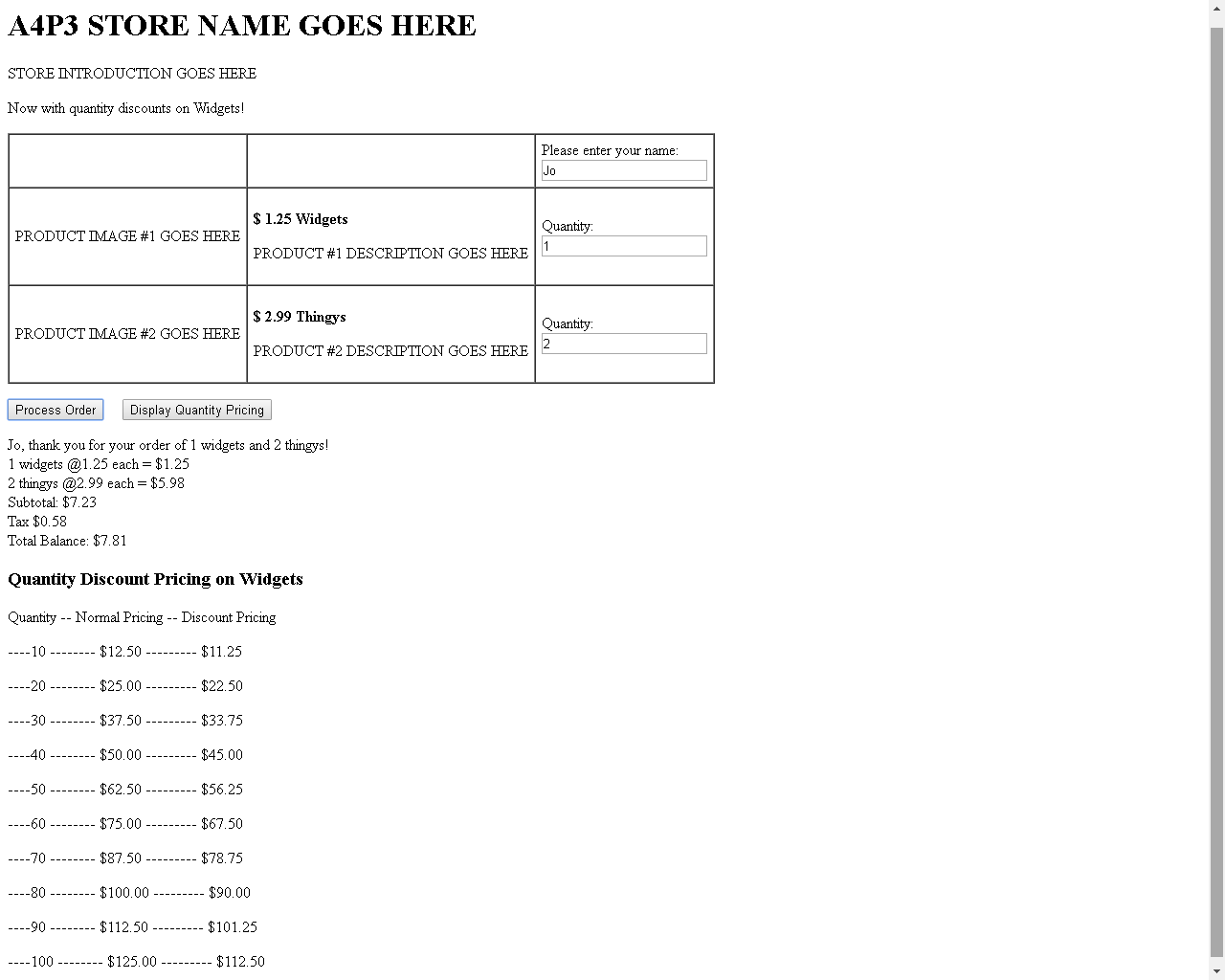
Into the given assignment 4 “turn it in” Word file, paste a copy of all of your web page’s code into the part 2 code section of the document. It should contain **ALL** of your page’s code. Do NOT paste a screenshot of code, paste the actual text of the code. **Make sure all of your code is visible** within the Word file's margins! You can format the font, spacing, tabbing, or size of the code after pasting into Word to improve its appearance and readability, just be sure not to actually change any of the code itself.

**Part 3: For Looping…**(Start with A3P4, then add a button to display a for-loop generated discount pricing list)

Begin by making a copy of your assignment 3 part 4 (A3P4) page and call your new copy “a4p3.html”. This part will focus on the For loop. We will use it to produce a list of discounted prices for large quantity orders for one of your two products. Check out the screen shot below to see what it should look like. The price list display should be triggered by the user clicking a new "Display Quantity Pricing" button that sits next to the existing "Process Order" button. (You should create the HTML for this new button.) When the user clicks that button, a new price list should be placed within a separate "div" or "p" below the existing "div" or "p" for the receipt by using its innerHTML property. (You should create the HTML for this new "div" or "p" with an appropriately named "id" so that it can later be used to access its innerHTML.) The "for" loop should produce one line of the price list each time it iterates through the loop. The loop counter should increase by 10 each time it loops (instead of increasing by 1 as it usually does) so that the price list is for quantities of 10, 20, 30, 40, etc., and then ends at 100. The first column to display is the quantity, the 2nd column is the regular pricing for that quantity, and the 3rd column is for the 10% discounted price for that quantity. Review the sample screen shot shown below to better visualize the intended results. You can either write an "event listener" for the "click" of that new button, or you can directly code the "onclick" event of the "Display Quantity Pricing" button. Either way, a function (that you write) should be called upon to produce the price list.

Note that the price list area should be separate and below the receipt area so that it doesn't matter which button is clicked first or second, because the information they produce will never interact directly with each other and the resultant page display should look the same either way. Note that you do not actually have to calculate any discounted prices into your order calculations for this part. Just get the display right for now and we'll add those additional calculations into the receipt in a later part. Lastly, keep in mind that you only have to show the discounted prices for one of your two products – so just pick one of your two products and use its current price as the basis for the 10% quantity discounts.

A4P3 Sample Screen Shot:



**What to turn in:**

**Screen shot**: Into the given assignment 4 “turn it in” Word file's part 3 screen shot area, paste a copy of your entire web page's display in the browser after clicking both buttons. The result should be something similar the sample screen shot shown above. Note that with the new discount pricing displaying, it may not all fit on one screen shot. If that is the case, use 2 screen shots as needed and paste both into the the Word file. Whether with one screen shot or two, the entire page's display (from the "h1" to the end of the price list) must be represented in your submission document to get full credit.

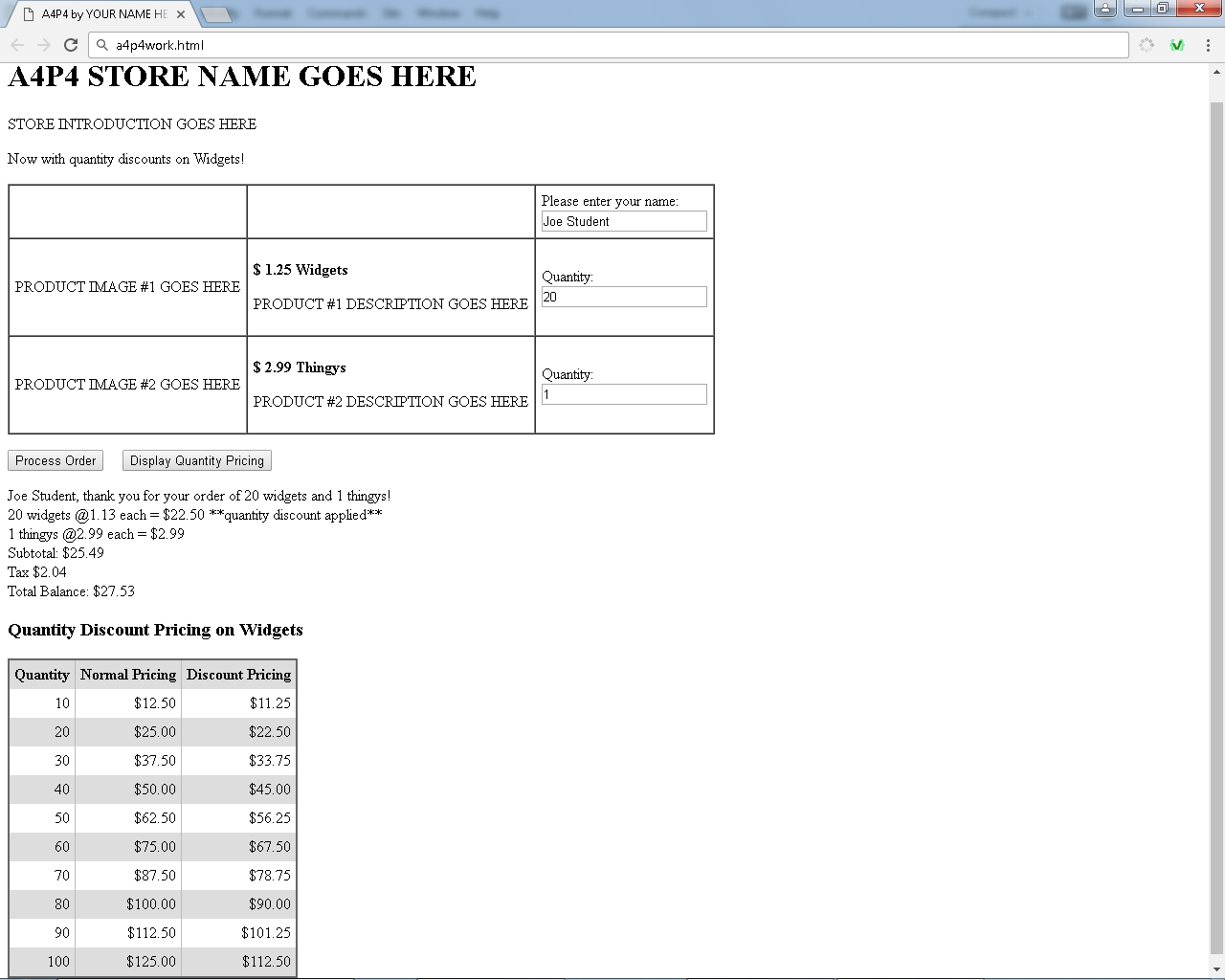
**Code**: Into the given assignment 4 “turn it in” Word file, paste a copy of all of your web page’s code into the part 3 code section of the document. It should contain **ALL** of your page’s code. Do NOT paste a screenshot of code, paste the actual text of the code. **Make sure all of your code is visible** within the Word file's margins! You can format the font, spacing, tabbing, or size of the code after pasting into Word to improve its appearance and readability, just be sure not to actually change any of the code itself.

**Part 4: Display Table…**(Start with A4P3, then modify code to calculate quantity discounts on the receipt and also display the discount pricing list with table formatting)

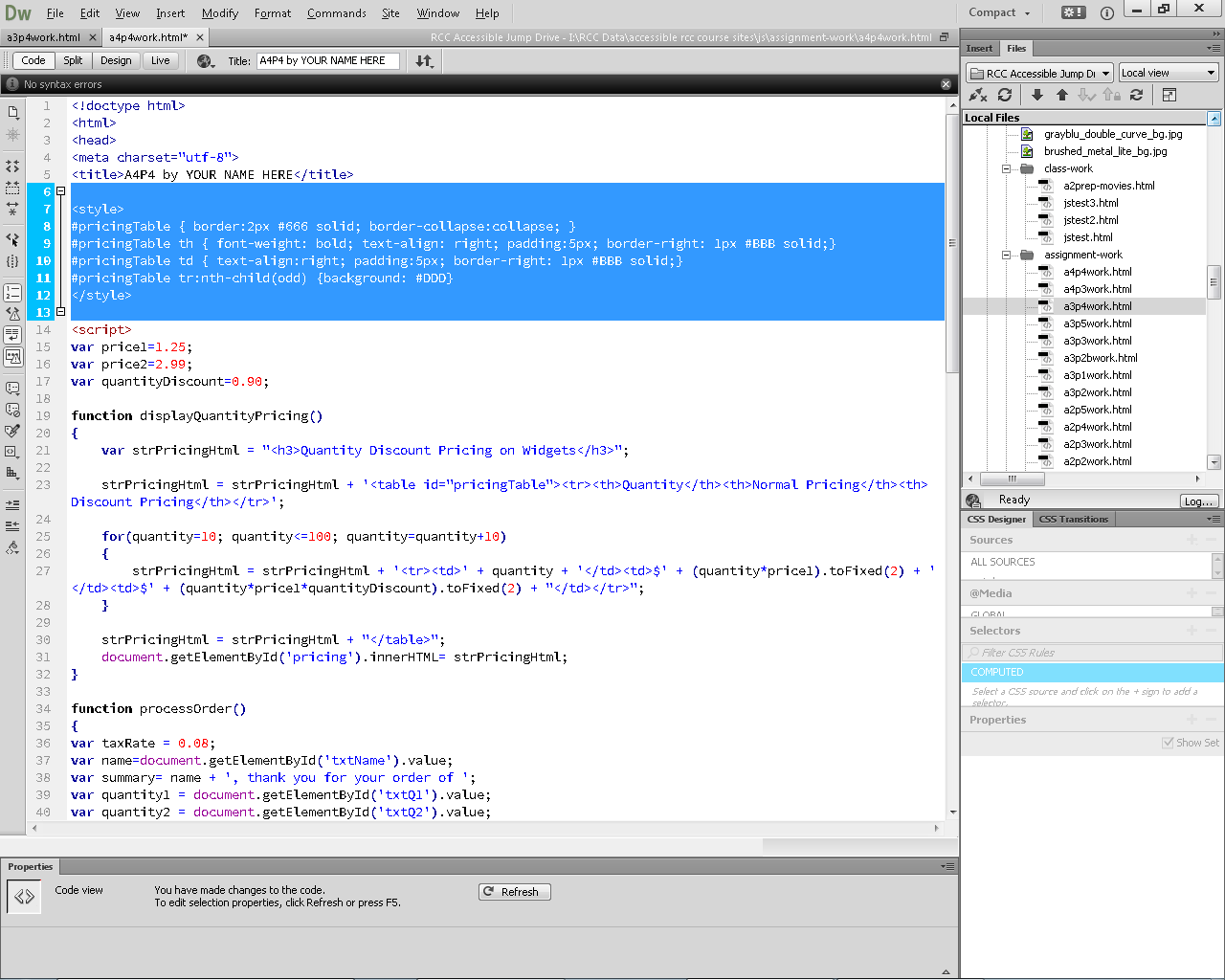
For this last part of the assignment, you will make 2 main changes to your A4P3 page, as detailed below. So begin by copying your a4p3 (part 3) page and naming it a4p4.html. Then work on these two changes.

Firstly, adjust your order processing calculations so that you actually give the quantity discount of 10% for orders of 10 or more for your chosen product like your quantity pricing list is advertising. The receipt should reflect this new pricing and should total up the receipt accordingly, as shown in the screen shot below where I am showing the 10% quantity discounted purchase for 20 widgets.

Secondly, adjust your quantity price list display so that the prices display in a nice table based columnar format using a styled HTML table as shown in the screen shot below. Below the screen shot, there is suggested CSS given that you can use to help format your table. In your script that produces your existing price list code (the "for" loop from part 3), you will need to produce the beginning of the table before the "for" loop, then the "for" loop code can be adjusted so that each iteration of the for loop produces one row of the table (each row has 3 columns, or td's), then below the "for" loop code should be added to complete the table. This code should result in a correctly generated (and valid) table with all of the proper tags opened and closed as if you were just writing the table with HTML and no JavaScript. It might help for you to actually create a few rows of this table with HTML to help you visualize what code you need to do above the loop, what code you need to do inside the loop, and what needs to still be done to finish off the table below below the loop.



To help you style the table as shown above, you can use the styles shown below. Write this CSS code in your "head" section. Then when you create your discount pricing table, give the table an HTML id of "pricingTable" to trigger these styles. That way your table code does not need to use "style" attributes, which can complicate the coding quite a bit.



**Challenges:**

If you want to challenge yourself a little further, use a second table to display the receipt in a nicely aligned format so that the receipt's numbers line up on the decimal point (right align them within the table cell).

Alternative challenge: Show a price list and also calculate discounted prices for your second product as well as your first product.

Note that these challenges are not required, and should only be submitted if it works properly. It will be worth a few extra credit points if it does work properly but it will only be used to offset other points you might be missing elsewhere on this assignment. You cannot score over 100 points.

**What to turn in:**

**Screen shot**: Into the given assignment 4 “turn it in” Word file's part 4 screen shot area, paste a copy of all of your web page displaying in the browser after clicking both buttons and using a quantity of 20 for your quantity discounted item. The result should be something very similar the sample screen shot above. Note that it may not all fit on one screen shot. If that is the case, use 2 screen shots as needed. All of the discount price list table should be visible.

**Code**: Into the given assignment 4 “turn it in” Word file, paste a copy of all of your web page’s code into the part 4 code section of the document. It should contain **ALL** of your page’s code. Do NOT paste a screenshot of code, paste the actual text of the code. **Make sure all of your code is visible** within the Word file's margins! You can format the font, spacing, tabbing, or size of the code after pasting into Word to improve its appearance and readability, just be sure not to actually change any of the code itself.