

CPSC 1045: Assignment 2

Due date set in D2L. 1% penalty per hour late if submitted the next day.

TO BE DONE INDIVIDUALLY

Overview

This assignment focuses on writing and using functions. For this assignment, the functions you need are carefully spelled out. In the future, you may be expected to figure out when it is useful/necessary to place code in its own function. Pay attention to why the functions in this assignment are useful. As you write each function, use the console to test if it is working correctly.

Code Style

A portion of your mark is based on writing your code with good style. A document on D2L will give you the full list of style requirements. Here is a partial list.

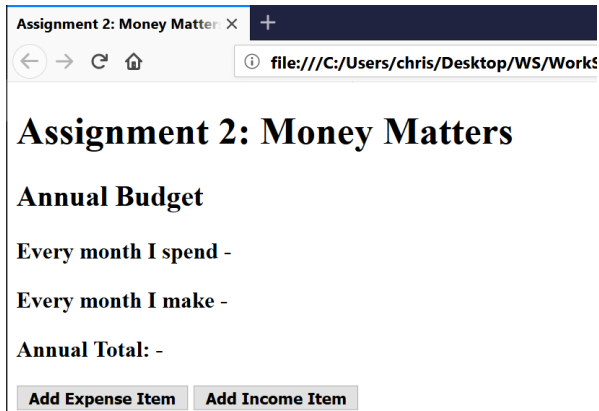
- Functions, variables, and parameters must have descriptive names
- Each function should have a comment above it explaining the function
- Code inside a function should be indented
- See the Style Guide on D2L for all requirements

Part 1: Annual Budget

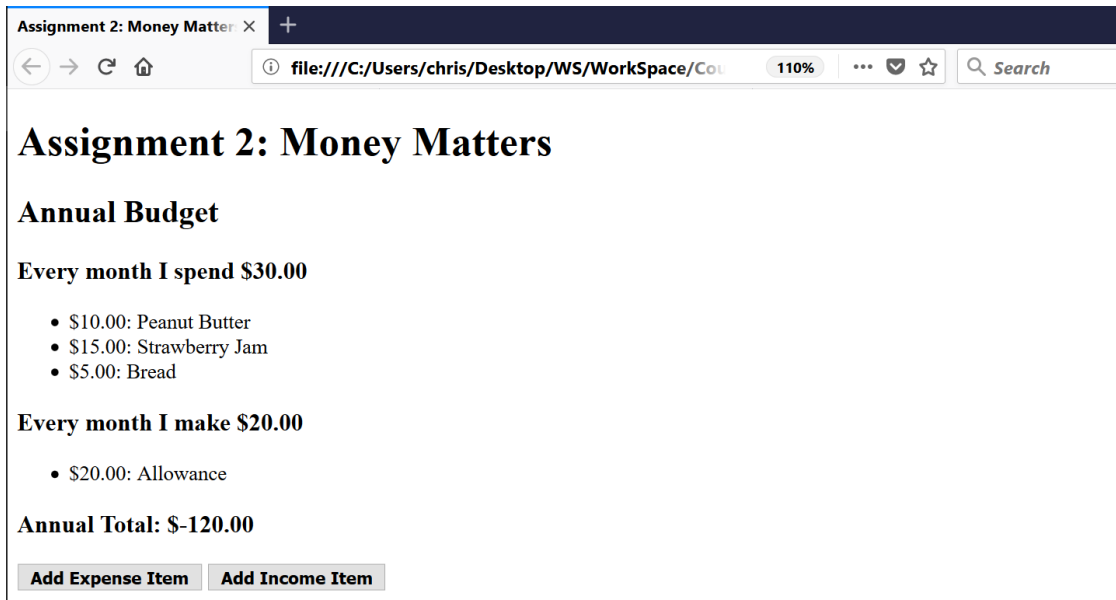
1. Download a2.zip and extract a2.html. It contains the html code you will need for Part 1. Create an a2.js file for the code for this assignment.
2. At the top of your javascript file, create **two global variables** initialized to 0. These will store the monthly expense total and the monthly income total. As a user adds monthly expenses and incomes, these variables should be updated. **DO NOT create any further global variables.**
3. Write a function called formatDollarAmount
 - a. **Parameters:** One numeric parameter representing an amount of money
 - b. **Operation:** Format the dollar value by adding a \$ sign and displaying the number with two decimal places. Use the .toFixed() function to do this.
https://www.w3schools.com/jsref/jsref_tofixed.asp
 - c. **Returns:** The string containing the formatted dollar amount. For example, formatDollarAmount(5.2) should return "\$5.20"
formatDollarAmount(1) should return "\$1.00"
 - d. Use the console to test that the function is working. It's a good idea to test all of your functions this way.
 - e. For the rest of this assignment, use this function for every dollar amount displayed on the webpage. If you wanted to change how they are all displayed, you would only have to change this function.
4. Write a function named addListItem
 - a. **Parameters:** Two string parameters. The first is the id of an HTML list element and the second is the text for the new item.
 - b. **Operation:** Adds the supplied text as a new list item to the list with the given id.
 - i. To add an item to the list it must be a complete list item element. Add the and tags to the text before adding the item to the list. The text will already be formatted.

1. If the text was "New Item", it would become "New Item" before being added to the list's innerHTML.
 - ii. You aren't replacing the innerHTML, you are adding something to what is already there
 1. Just like you can use `x += 5`; you can use `+=` here.
 - c. **Returns:** Nothing
 - d. For example, this function might be called with parameters like this
 - i. `addListItem("expenses", "$30.00: Peanut Butter")`
 - e. Use this function when adding items to the expense/income lists.
5. Write a function named `updateTotals`
 - a. **Parameters:** None. The global variables store the information it needs
 - b. **Operation:** The two global variables store the total monthly income and total monthly expenses. Calculate the annual total ($12 * (\text{income} - \text{expenses})$) and display the three values in the three span elements on the webpage. **This is the only function that should change the values displayed on the webpage.**
 - c. **Returns:** Nothing
 - d. This function will be used after an item has been added to one of the lists.
6. Write the event handler functions that will be called for clicks of the Add Expense/Income buttons. They will prompt the user for the item and the amount of money.
 - a. **Parameters:** None. These are event handlers.
 - b. **Operation:** Prompt the user for an item and the amount of money associated with it. Then use the `addListItem()` function to add the new item to the appropriate list. Adjust either the global variable for expense or income. Call the `updateTotals()` function to display the updated values.
 - c. **Returns:** Nothing

Initial Screenshot



Final Screenshot



Submission

- Submit to D2L ONE zip/archive file containing a2.html and a2.js
- **If you do not zip your files, you will receive a 2 mark penalty**

Grading

Marks will be assigned as follows:

- 15 marks for Part 1
- 5 marks for code style (Javascript and HTML). See the style guide on D2L