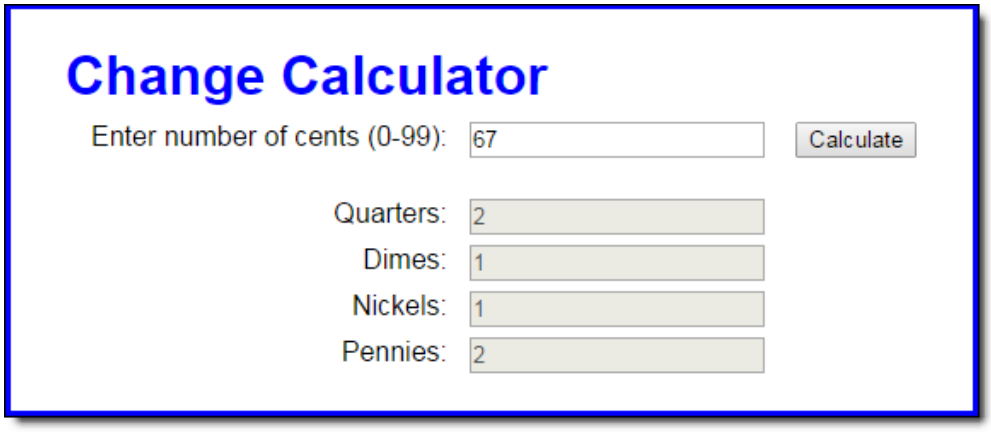
# Assignment #5 - Use an object literal and a constructor with the Change Calculator

In this assignment, you'll modify a Change Calculator application so it uses an object literal and a constructor function. (NOTE: in a real-world project you probably wouldn't use both an object literal and a constructor, but this is just for practice ☺ )




1. Open the HTML and JavaScript files in this folder:

   **`Assign5_start\`**

   Note that there are two JavaScript files for this application: the main JavaScript file (calculate.js) and the start of a library file (library_coin.js). You should not have to modify the HTML or CSS files in this assignment.

2. In the **calculate.js** file, note that three functions are supplied. The **$** function. The **calculateChange** function that contains all of the code for the application. And an **onload** event handler that attaches this function to the click event of the Calculate button and sets the focus on the first field.

3. In the library_coin.js, note that just the strict declaration has been provided.

4. In the index.html file, add the script tag for the library_coin.js file.

5. In the library_coin.js file, code an object literal named **coins** that has a cents property and two methods:

   The **isValid** method should determine whether the cents property is valid.

   The **getNumber** method should accept a divisor parameter (like 25 for quarters), calculate the number of coins of that type that are required, update the cents property with the remaining cents, and return the number of coins.

6. Change the code in the **calculate.js** file to use the object literal to get the cents entered by the user, validate the user's entry, and calculate the number of coins. (validation and calculations will now be in the object literal, so calculate.js will be much shorter.

7. Test your application to make sure it works well so far.

8. Now, in the library file, code a constructor function named **Coins** that accepts a parameter named **cents**, which is the number of cents entered by the user. Then, code a

cents property and two methods: (these 2 methods are the same as in the object literal above, so you can copy-and-paste the code)

The **isValid** method should determine whether the cents property is valid.

The **getNumber** method should accept a divisor parameter (like 25 for quarters), calculate the number of coins of that type that are required, update the cents property with the remaining cents, and return the number of coins.

9. Comment out the line of code from calculate.js that you wrote in step 6 to create the object literal. Now add a line of code in the calculate.js file to create an instance of the Coins object type to validate the user's entry and calculate the number of coins.
10. Test your application to make sure it works with different values.

**Rubric** (10 points) The assignment will be assessed according to the following criteria:

-- Problem Has Been Solved and meets all requirements based on the Written Instructions (10 pts)

-- Mostly Meets Requirements (7-9pts)

-- Somewhat Meets Requirements (4-6pts)

-- Rarely Meets Requirements (1-3pts)

-- Not Attempted (0 pts)