Homework 2 - JavaScript

In this assignment, you will implement some simple JavaScript functions.

In completing this assignment, you will:

- Gain familiarity with JavaScript syntax and writing JavaScript functions
- Get experience working with JavaScript arrays and objects
- See how to execute JavaScript code using a web browser's console

Debugging/Error Note:

If you run into errors/bugs/don't understand the output that Codio is giving you, please post in the Discussion Forum and a TA will assist you! Please do NOT email Codio as they will not review any errors you are getting.

Getting Started

Start by downloading the two files you will need for this assignment. Right-click **this link** and save "petstore.js" to your local computer; then right-click **this link** and save "petstore.html" to your computer.

The petstore.js file contains the starter code for the functions that you will implement in this assignment; their full descriptions are below. You should be able to open this file with a plain-text editor and see the code that we provide.

You will also see that petstore.js includes prototype functions for creating the objects you will need in implementing and testing the functions that you will write. Please do not change these functions.

At the bottom of petstore.js is a simple "hello world" function that you can use to make sure your development environment is set up correctly.

For this assignment, we recommend that you edit petstore.js using a plain-text editor and then use your web browser's JavaScript console to execute your code. If petstore.js is in the same folder/directory as petstore.html, and you open petstore.html, you should be able to access the functions in petstore.js from the browser's console.

We recommend you use **Google Chrome** for this assignment. Open petstore.html in your browser, then open the console by selecting View in the main menu, then Developer, then JavaScript Console.

If you are using **Safari**, you will need to enable the console by selecting Safari in the main menu, then Preferences, then Advanced, then checking "Show Develop menu in menu bar." Once you've done that, open petstore.html then open the console by selecting Develop in the main menu, then Show Error Console.

If you are using **Mozilla Firefox**, first open petstore.html then open the console by selecting Tools in the main menu, then Web Developer, then Web Console.

Once you've opened the console, type " `helloworld()`" in the console (without the quotes) to run the helloworld() function in petstore.js.

Your browser should print "hello world!" in the console. If so, you're ready to proceed!

Activity

The functions that you will implement in this assignment are all related to the operation of a pet store (as you'll see later in the course, the SD4x instruction staff really likes animals!), where shoppers can purchase animals as pets.

In petstore.js, implement these functions as follows:

calculateFoodOrder: This function should calculate the total amount of pet food that the store should order for the upcoming week. The *numAnimals* parameter represents the number of animals in the store, and *avgFood* represents the average amount of food (in kilograms) eaten by each animal each week. The function should return the total amount of pet food that should be ordered for the upcoming week, or -1 if *numAnimals* or *avgFood* is less than 0 or non-numeric.

mostPopularDays: This function determines which day of the week had the most number of people visiting the pet store. If two or more days are tied for the highest amount of traffic, an array containing the days (in any order) should be returned. If the input is null or an empty array, the function should return null. The input is an array of Weekday objects, which are created using the prototype function defined toward the bottom of petstore.js. This function should return a string containing the name of the most popular day of the week if there is only one most popular day, and an array containing the names (as strings) of the most popular days if there are more than one that are most popular.

createAnimalObjects: Given three arrays of equal length containing information about a list of animals – where *names[i], types[i]*, and *breeds[i]* all relate to the same, single animal – this function should return an array of Animal objects constructed from the information provided in the arrays. The parameter *names* represents the array of the animals' names; *types* represents the array of the animals' types (e.g. "Dog", "Cat", "Bird"); and *breeds* represents the array of the animals' breeds. This function should return an array of Animal objects (which you can create using the prototype function at the bottom of petstore.js), each of which contains the animal's information, or an empty array if the arrays' lengths are unequal or zero, or if any array is null.

Your implementations are only expected to handle the "normal" operations of these functions, and any extra conditions listed above. You may think of or encounter other situations or inputs not described here, but you only need to consider the ones listed above for grading.

One more important note:

Please do not change the names or lists of parameters for any of the functions we have provided, as these will be used by our tests during grading. Also, please do not change the function prototypes for the Weekday, Item, or Animal classes, as those are used by the grader as well.

Be sure that all JavaScript is in petstore.js and that you have not created any additional files.

Helpful Hints

Be sure that you're able to run the "helloworld()" function in your browser's JavaScript console before trying to implement the other functions, so that you know you have a development environment in which to work.

For *calculateFoodOrder*, use the JavaScript *Number* function for converting a variable to its numeric form, and consider the *isNaN* function for determining whether the conversion was successful.

For *mostPopularDays*, remember that you can use the JavaScript console as a “REPL” to create variables – including objects – and then use them in invoking your functions. For instance, you can create a Weekday object using the JavaScript “new” keyword with the prototype function (review the lesson on functions if you don’t recall how to do that), and then create an empty array and then use the array’s “push” function to add objects to the array before calling *mostPopularDays*.

You can then use a similar approach in implementing *createAnimalObjects* when it comes to creating new values and putting them into an array.

Before You Submit

Please be sure that:

- You have not changed the names or parameter lists of any of the functions we have provided, or of the function prototypes for the Weekday, Item, and Animal classes
- You have not created any additional files and all of your JavaScript code is in petstore.js

Assessment

Your submission will be assessed using automatic grading scripts that will check that the JavaScript functions are correctly implemented for various inputs. Your score is determined by the percentage of these tests that pass, i.e. that produce the correct output for the given input.

To submit your assignment, click the "BEGIN SUBMISSION" button below and follow the instructions in the Codio Readme file. You only need to upload **petstore.js** for grading.

Homework #2 Submission (External resource) (100.0 points possible)

This link will take you to Codio so that you may submit this assignment.

BEGIN SUBMISSION