CS591SA1 MEAN Summer 2019

Problem set 1: functions (three problems)

Due: 26 May 11:59pm

Part 0

This and future code assignments will be turned in through your Gradescope account. The Gradescope interface will ask you to identify a repository and branch on GitHub when you submit an assignment.

This means that you'll need a GitHub repo to work from. Here's some advice on organization to make things easier:

- You'll need to have git installed on your local machine. git comes with the command line tools in XCode on MacOS (you can verify git is installed with 'git —version'). If you need to install, look for instructions for your OS at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

- If you don't already have a github account, create one at github.com, otherwise log in to your account.

- On github, create a new public repository to hold your course files. On this page, choose to initialize the repo with a README file (this helps with the next step). Also add a .gitignore file…choose Node from the dropdown menu.

- Once the repo is created, click the green 'Clone or download' button and copy the URL displayed to the clipboard.

- Open a terminal window on your local machine and move to a directory that will hold your local course files (for example, 'cd ~; mkdir cs591sa1; cd cs591sa1'.

- Clone the new repo onto your local machine: git clone repo-url-copied-in-previous-step; the repo will be created as a directory with the repo's name, located in your current directory.

This directory holds your course repo, and each assignment will be saved in a subdirectory of it. For example, this assignment should be save in a directory called PS1. You can create new directories and add them to your repo right in your IDE (I'll demo what it looks like in Webstorm).

Create a new branch called PS1 to track files for this assignment (it'll be this branch that you submit on Gradescope).

Part 1

For this assignment, write JavaScript code to satisfy each of the following problems. Each solution should be in a separate file which is named to reflect the problem number. For example, PS1/PS1.P1.js would be the file holding your solution to Problem Set 1, Problem 1.

You'll end up pushing your entire directory and all of the solution files to github, which can be done either from the command line or from your IDE. After your files are pushed, submit your assignment on Gradescope.

For these problems, you'll need to both write the function (and any helper functions you require), and also a line or two that executes the function with sample input, printing the result on the console. You'll also need to dig around to find ways to solve them; for example, when working with strings, take a look at the docs for the String library to get ideas of what can be done. The 'official' docs are at developer.mozilla.org. Most solutions require chaining a few functions.

Note: For each problem, write your solution using 'fat arrow' (=>) notation.

Example 1
Write a function that returns the largest integer of an array of integers

```
const biggie = items => Math.max(...items);
console.log(`Biggest int is: ${biggie([4,8,1,4,3,9,2])}`)
```

Problem 1 (PS1.P1.js)
Write a function that takes a string as its input and returns a new string that contains all of the letters in the original string, but in alphabetical order. Ignore punctuation and numbers. Test your function using the string 'supercalifragilisticexpialidocious'.

Problem 2 (PS1.P2.js)
Write a function that takes as input each of the following strings in turn:

'4+2'
'5*7'
'6-1'
'9/2'

This function should
    Determine the operator (+, *, -, or /) embedded in the string
    Return a function to implement the input operator that returns the result

For example, if the input string is '8%3', return   (left, right) => left % right

Execute the returned function to evaluate and print each expression. For example,

```
const expression = '8%3';
let operator = evaluate(expression);
console.log(`${expression} = ${operator(expression)}`)
```

You can assume that the format of the input string is fixed (i.e. a digit, an operator, and a digit, always the same length).

Problem 3 (PS1.P3.js)
Write a function that accepts two input parameters: a string, and a function. The function should execute the passed function with the passed string and return the result.

Next, write two expressions that call this function. For the first, pass the string ‘supercalifragilisticexpialidocious’ and a lambda function that returns an array containing fragments of the input string broken on the character ‘c’. For the input string ‘supercalifragilisticexpialidocious’, you should get

```
[‘super’, ‘califragilisti’, ‘cexpialido’, ‘cious’]
```

For the second, pass the string ‘supercalifragilisticexpialidocious’ and a lambda function that replaces all of the ‘a’ characters with ‘A’ characters. Return an object that contains the original string, the modified string, the total number of As in the modified string, and the overall length of the modified string:

```
{
        originalString:         xxx,
        modifiedString:         xxx,
        numberReplaced:         xxx,
        length:                 xxx
}
```

Print the data from the returned object on the console.