9/21/2019

CS303 Object Oriented and Functional Programming in JavaScript

W1D2 Code Quality Assignment

PART I: Complete the following tasks from The JavaScript Language book. You do not have to submit these to GitHub. Try to complete the answers before looking at the solutions.

Coding Style section: Bad style task

Ninja code section: (nothing to implement here, just write the answers)

Read the 'Ninja code' section on your own. Write the real rules implied by the irony examples. E.g.,

- Ninja irony: Make the code as short as possible. Show how smart you are
  - Meaning: do not sacrifice code clarity for brevity.
- Ninja irony: use single-letter variable names everywhere.
  - Meaning: *your answer here*
- Ninja irony: If the team rules forbid the use of one-letter and vague names – shorten them, make abbreviations
  - Meaning: *your answer here*
- Ninja irony: While choosing a name try to use the most abstract word
  - Meaning: *your answer here*

Etc etc

Automated testing with Mocha section: **What's wrong in the test?**

Part II. Implement the following in VSCode, and submit to the GitHub repository for this assignment. Use esLint and JS Doc with your code. The eslint configuration will require you to write JS Doc for each function. See the example JSDoc comment in the instructions for installing esLint.

1. Write unit tests for these 3 functions you implemented yesterday (maxOfThree, sum, findLongestWord). Each test should have at least 3 it use cases.

2. The following is a unit test for an isVowel function. Implement the function so that the unit tests are satisfied.

```
"use strict";
```

```
/* global assert isVowel*/
/*  isVowel() that takes a character (i.e. a string of length 1) and returns true if it is a vowel, false otherwise. */
describe("isVowel", function () {

   it("a is vowel", function () {
      assert.equal(isVowel("a"), true);
   });

   it("e is vowel", function () {
      assert.equal(isVowel("e"), true);
   });
   it("i is vowel", function () {
      assert.equal(isVowel("i"), true);
   });
   it("o is vowel", function () {
      assert.equal(isVowel("o"), true);
   });
   it("u is vowel", function () {
      assert.equal(isVowel("u"), true);
   });

   it("z is not vowel", function () {
      assert.equal(isVowel("z"), false);
   });

   it("5 is not vowel", function () {
      assert.equal(isVowel("5"), false);
   });
});
```

3. Write your own unit tests in the BDD fashion for the following two functions, and then implement the functions to pass the tests.

1. Define a function reverse() that computes the reversal of a string. For example, reverse("jag testar") should return the string "ratset gaj".
2. Write a function filterLongWords() that takes an array of words and an integer i and returns the array of words that are longer than i.

After you finish, make sure all your functions are in ONE JS file