

C M S I 3 8 6

# Homework #2

Due: 2013-09-26

---

For this homework assignment, you'll be warming up to JavaScript. **Readings:** Read as many on-line tutorials or book chapters on JavaScript as is reasonable without seriously impacting your personal life.

**INSTRUCTIONS:** Submit all solutions on paper, but provide the code online as well in two [jsfiddles](#): one for problems 1-9 and one for problem 10. You may use jQuery.

1. Write a JavaScript function that accepts a number of U.S. cents and returns an array containing, respectively, the smallest number of U.S. quarters, dimes, nickels, and pennies that equal the given amount.

```
> change(96)
[3, 2, 0, 1]
> change(8)
[0, 0, 1, 3]
```

2. Write a JavaScript function that takes in a string *s* and returns the string which is equivalent to *s* but with all ASCII vowels removed.

```
> stripVowels("Hello, world")
'Hll, wrld'
```

3. Write a JavaScript function that *randomly* permutes a string. By random we mean that each time you call the function for a given argument all possible permutations are equally likely (note that "random" is not the same as "arbitrary").

```
> scramble("Hello, world")
'w,d!roH e!o!'
```

4. Write a JavaScript function that yields successive powers of two starting at 1 and going up to some limit. Consume the values with a callback.

```
> powersOfTwo(70, function (p) {console.log(p);})
1
2
4
8
16
32
64
```

5. Write a JavaScript function that yields powers of an arbitrary base starting at exponent 0 and going up to some limit. Consume the values with a callback.

```
> powers(3, 400, function (p) {console.log(p);})
1
```

3  
9  
27  
81  
243

6. Write a JavaScript function that interleaves two arrays. If the arrays do not have the same length, the elements of the longer array should end up at the end of the result array.

```
> interleave(["a", "b"], [1, 2, true, null])
["a", 1, "b", 2, true, null]
```

7. Write a JavaScript function that doubles up each item in a list.

```
> stutter([5,4,[3],9])
[5,5,4,4,[3],[3],9,9]
```

8. Write a JavaScript function that produces a word count object from a string. Consider the words of the string to be any sequence of one or more Latin letters plus the apostrophe. (Yes, that means you can "throw away" all non-Latin letters, because, well, ES5 JavaScript doesn't have `\p{L}` in its regexes. Argh.) Your script should treat the words case-insensitively, so just lowercase everything.

```
> wordCount("If you dog a dog, you'll\nbe DOG-TIRED.")
{"if":1, "you":1, "dog":3, "a":1, "you'll":1, "be":1, "tired":1}
```

9. Flesh out the unit tests for problems 1-8 that I have started here for you.

```
$(function () {

    var anagram = function (s, t) {
        if (s.length !== t.length) {
            return false;
        }
        var a = s.split(""), b = t.split("");
        return $(a).not(b).length === 0 && $(b).not(a).length === 0
    };

    test("Change Tests", function () {
        deepEqual(change(97), [3, 2, 0, 2]);
        deepEqual(change(8), [0, 0, 1, 3]);
        // More needed
    });

    test("Strip Vowels Tests", function () {
        deepEqual(stripVowels("Hello, world"), "Hll, wrld");
        // More needed
    });

    test("Scramble Tests", function () {
        var data = ["", "a", "rat", "zzz", "^*&^*&^&ÄËËgnΩ"]
        data.forEach(function (s) {
            ok(anagram(s, scramble(s)));
        });
    });

    test("Powers of Two Tests", function () {
        // Implement these
    });
});
```

```

    });

    // Test powers here

    // Test interleave here

    // Test stutter here

    // Test word count here
  });

```

Use the following HTML driver for the tests:

```

javascriptwarmup.html

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript Warmup Test</title>
    <link rel="stylesheet" href="http://code.jquery.com/qunit/qunit-git.css">
  </head>
  <body>
    <div id="qunit"></div>
    <div id="qunit-fixture"></div>
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <script src="http://code.jquery.com/qunit/qunit-git.js"></script>
    <script src="javascriptwarmup.js"></script>
    <script src="javascriptwarmuptest.js"></script>
  </body>
</html>

```

10. Write a JavaScript script that writes successive prefixes of the value entered into a web page text field into new `div` elements, one after another, starting with the first prefix, which is zero characters long. The new `divs` should be added approximately one second apart.

Assuming the string "matsumoto" were entered, the `divs` would be:

- o
- o m
- o ma
- o mat
- o mats
- o matsu
- o matsum
- o matsumo
- o matsumot
- o matsumoto