Js Quest07

Subject | 1 Solution | Additional Resources (1)

Js Quest07

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under `submit directory` and specific file names for each exercise are also provided at the beginning of each exercise under `submit file(s)`.

Introduction

This is the final quest, and we will conclude with one of the big piece of coding is about using the `correct` data structure.

Control Center

roup formation

Progress

ubmitted

est review

Finished: approved

Go To DoCode

Access:

READ WRITE

Go To Gitea

Keep Working On This Soluti

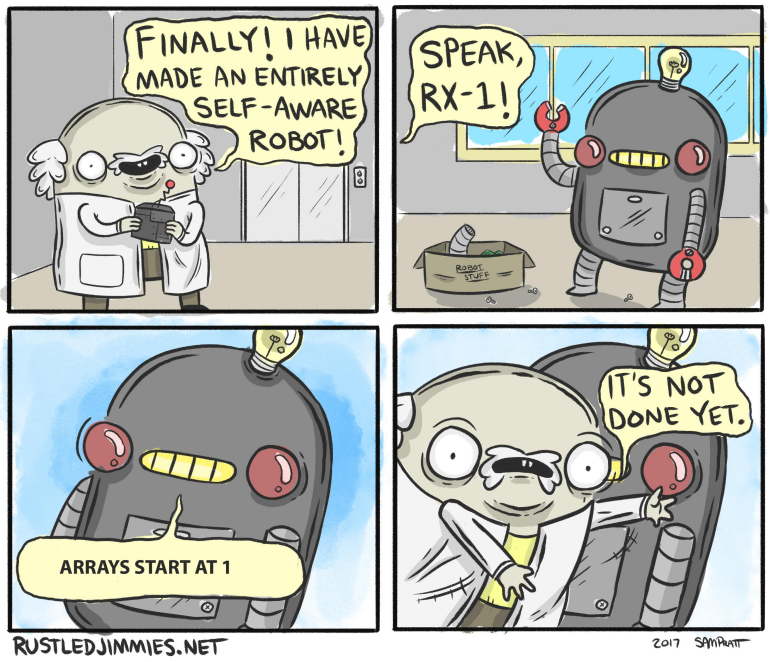
What is a data structure?
An array is a data structure.
But it's not the only one, `hash`, `queue`, `stack`, `linked list`, `tree`, ... they are a lot of different data structure. :D

They can perform different type of operation and specially at different speed OR/AND use of memory.

For example an `Array` is faster to search (and even more if it's sorted) than a `Linked list` but a `Linked list` is easier to insert an element in the middle of it.

For this quest you will only need to know how to use an `array` and a `dictionary` (or often called a `Hash`).

Good luck :-)
PS:



Js Quest07	My Map Mult Two
Submit directory	ex00
Submit file	my_map_mult_two.js

Description

Multiply by 2 each elements of an array.

Looking for a group

lucas_v

Also working on the project


abdullay_b

rasulov_d

suxrobov_s

risalieve


xujamura_d

to-xtasi_a

rahmono_az

isakova


asomov_u

axmadxo-_j

sodiqov_o

alpeisso_n


sultonov_x

nigmatu_mu

begimqul_g

xudaboye_s


norxojay_o

stupakov_k

sodikov_ar

sultanba_y

Just finished


parpiev_s

diyarova_s

muhiddin_s

uktamov_s




Create a function `my_map_mult_two` which receives an integer array as parameter and iterate over the array, perform a multiplication by 2 on each value and return the new array collected.

Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_map_mult_two
**
**
** @param {Integer[]} param_1
** @return {integer[]}

**
*/


function my_map_mult_two(param_1) {

};
```

Example 00

```
Input: [1, 2, 3, 4, 5]
Output:
Return Value: [2, 4, 6, 8, 10]
```

Example 01

```
Input: []
Output:
Return Value: []
```

Tips
Google while YOURCODINGLANGUAGE
Google for YOURCODINGLANGUAGE
Google array YOURCODINGLANGUAGE

zharas_g orifjono_a g-ofurov_s isakova




abdurahi_a yusupov_f abduhaki_b qayumov_s




radjapov_i xudaboye_s musayev_s tsay_d

Type
Project

Group Size
1 Participant

Review system
Test Review (Gandalf)

Difficulty
Initiation

Average duration
1 Week

Project's Metadata

Project

id: 242

name: js-quest07

visible: True

Js Quest07	My Count On It
Submit directory	ex01
Submit file	my_count_on_it.js

Description

Count the size of each elements in an array.

Create a function `my_count_on_it`, which receives a string array as parameter and returns an array with the length of each strings.

Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_count_on_it
**
**
** @param {String[]} param_1
** @return {integer[]}

**
*/


function my_count_on_it(param_1) {

};
```

Example 00

```
Input: ["This", "is", "the", "way"]
Output:
Return Value: [4, 2, 3, 3]
```

Example 01

```
Input: ["aBc", "AbcE Fgef1"]
Output:
Return Value: [3, 10]
```

Example 02

```
Input: ["aBc"]
Output:
Return Value: [3]
```

Tips
Google while YOURCODINGLANGUAGE
Google for YOURCODINGLANGUAGE
Google array.length YOURCODINGLANGUAGE

Js Quest07	My Array Uniq
Submit directory	ex02
Submit file	my_array_uniq.js

Description

Create an array without any duplicates.

Create a function `my_array_uniq`, which receives an integer array as a parameter and returns an array with those integers but without any duplicates.

Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_array_uniq
**
**
** @param {Integer[]} param_1
** @return {integer[]}

**
*/


function my_array_uniq(param_1) {

};
```

Example 00

```
Input: [1, 1, 2]
Output:
Return Value: [1, 2]
```

Example 01

```
Input: []
Output:
Return Value: []
```

Example 02

```
Input: [1, 1, 1, 2, 3, 4, 1]
Output:
Return Value: [1, 2, 3, 4]
```

Tip
Google filter/uniq YOURCODINGLANGUAGE

Js Quest07	My Average Mark
Submit directory	ex03
Submit file	my_average_mark.js

Description

Hash is a data structure which works like an array but instead of having numerical value as index you can specify "any type" of key:

```
# ARRAY
my_array = [1, 2, 3]
my_array[0] # 1

# HASH
my_hash = {'age' => 34, 'name' => 'Luke'}
my_hash['age']  # 34
my_hash['name'] # Luke
```

Assignment:
You have a test for a class and we want to know the average results for the class. Write a function that takes an array of hash with all of the grades/marks for a given test and returns an average grade for the entire class.
Input are in JSON (be careful ruby users, "string": is not a symbol)

Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_average_mark
**
**
** @param {String_integer[]} param_1
** @return {float}

**
*/


function my_average_mark(param_1) {

};
```

Example 00
Example 00: (In Javascript)
John, Margot, Jules, and Marco are in a class together.

```
Input: [
        {"string": "John", "integer": 7},
        {"string": "Margot", "integer":
8},
        {"string": "Jules", "integer":
4},
        {"string": "Marco", "integer":
19}
       ]

Output: 9.5
```

Example 01

```
Input: [
        {"string": "Quentin", "integer":
1},
        {"string": "Fred", "integer": 1},
        {"string": "Julia", "integer":
18},
        {"string": "stephanie",
"integer": 13}
       ]
Output: 8.3
```

Example 02

```
Input: []
Output: 0.0
```

Tips
(In Javascript)
For javascript user in order to get the right precision of the floating number, use toFixed().
Google the following: Float variable type
Google the following: what is a Hash (look at dictionary or technical definition and use cases)
Google the following: Object.keys(hash)