**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.

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.

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.length YOURCODINGLANGUAGE

| **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