TypeScript Exercise (Activity 1)

Notes

Make sure to follow all the instructions carefully, this exercise will be graded with the following criteria:

Working Code --- 40 %

Followed the Requirements --- 40 %

Error Free – 20 %

Place all your code in your GitHub repository and make sure all necessary files are included.

It's ok to reach Stack Overflow for help but never search for the exact code of the answer!

Most important Rule: **Have Fun!**

1. City Directory

We will create a simple application that allows adding and filtering the list of cities.

Requirements:

Create Form

Create a form with an input of City Name, Country, and Population

City Name: should be an input of type string

Country: should be an input of type string

Population: should be an input of type number

This form will add a new city to the directory

List View

This is a view of all the list of cities added to the directory, it should display the city, country, and population.

Search Input

Once the user types in a value, the list will be filtered out depending on the value, this should be case insensitive.

Example

Current List

{

city: “Manila”,

country: “Philippines”

},

{

city: “Breda”,

country: “Netherlands”

},

{

city: “Tokyo”,

country: “Japan”

}

Current Search Value: an

Result in the list: this will still display all three cities with Manila,

Netherlands and Japan have the “an” value (this means you can search by country or city)

Other Requirements:

Once I refresh the application, this should maintain the list of cities (Tip: try to explore the use of local storage)

UI is optional, you are free on what your application would look.

2. ISBN – 10 Validation

ISBN-10 identifiers are ten digits long. The first nine characters are digits 0-9. The last digit can be 0-9 or X, to indicate a value of 10.

An ISBN-10 number is valid if the sum of the digits multiplied by their position modulo 11 equals zero.

For example:

ISBN : 1 1 1 2 2 2 3 3 3 9

position : 1 2 3 4 5 6 7 8 9 10

This is a valid ISBN, because:

(1*1 + 1*2 + 1*3 + 2*4 + 2*5 + 2*6 + 3*7 + 3*8 + 3*9 + 9*10) % 11 = 0

Examples

1112223339 --> true

111222333 --> false

1112223339X --> false

1234554321 --> true

1234512345 --> false

048665088X --> true

X123456788 --> false

You can use **console.log()** to output the return value of your answer.

3. Change it up!

Create a function that takes a string as a parameter and does the following, in this order:

Replaces every letter with the letter following it in the alphabet (see note below)

Makes any vowels capital

Makes any consonants lower case

Note:

the alphabet should wrap around, so Z becomes A

in this kata, y isn't considered as a vowel.

So, for example the string "Cat30" would return "dbU30" (Cat30 --> Dbu30 --> dbU30)

4. Moving Zeroes to the End

Write an algorithm that takes an array and moves all of the zeros to the end, preserving the order of the other elements.

Example:

moveZeros([false,1,0,1,2,0,1,3,"a"]) // returns[false,1,1,2,1,3,"a",0,0]

You can use **console.log()** to output the return value of your answer.