**5 kyu**

**First non-repeating character**

2225688% of 1,2431,158 of8,966[OverZealous](https://www.codewars.com/users/OverZealous)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/first-non-repeating-character/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/first-non-repeating-character/solutions/csharp)

[Forks (7)](https://www.codewars.com/kata/first-non-repeating-character/forks/csharp)

[Discourse (92)](https://www.codewars.com/kata/first-non-repeating-character/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Write a function named first\_non\_repeating\_letter that takes a string input, and returns the first character that is not repeated anywhere in the string.

For example, if given the input 'stress', the function should return 't', since the letter *t* only occurs once in the string, and occurs first in the string.

As an added challenge, upper- and lowercase letters are considered the **same character**, but the function should return the correct case for the initial letter. For example, the input 'sTreSS' should return 'T'.

If a string contains *all repeating characters*, it should return None.

<https://www.codewars.com/kata/first-non-repeating-character/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static string FirstNonRepeatingLetter(string s)

{

// return s.Substring(0, 1);

Dictionary<char, int> dic = new Dictionary<char, int>();

string copia = s.ToLower();

for (int i = 0; i < s.Length; i++)

{

if (dic.ContainsKey(copia[i])) dic[copia[i]]++;

else dic[copia[i]] = 1;

}

for (int i = 0; i < copia.Length; i++)

{

if (dic[copia[i]] == 1)

{

return s[i].ToString() ;

}

}

return String.Empty;

}

static void Main(string[] args)

{

Console.WriteLine(FirstNonRepeatingLetter("sTreSS"));

Console.ReadLine();

}

}

}