**Is it an isogram?**

2184% of 2825 of49[DiegoSalazar](https://www.codewars.com/users/DiegoSalazar)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/is-it-an-isogram/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/is-it-an-isogram/solutions/python)

[Discourse (11)](https://www.codewars.com/kata/is-it-an-isogram/discuss/python)

* Add to Collection
* |
* Share this kata:

**An [isogram](https://en.wikipedia.org/wiki/Isogram" \t "_blank)** (also known as a "nonpattern word") is a logological term for a word or phrase without a repeating letter. It is also used by some to mean a word or phrase in which each letter appears the same number of times, not necessarily just once.

You task is to write a method isogram? that takes a string argument and returns true if the string has the properties of being an isogram and false otherwise. Anything that is not a string is not an isogram (ints, nils, etc.)

**Properties:**

* must be a string
* cannot be nil or empty
* each letter appears the same number of times (not necessarily just once)
* letter case is not important (= case insensitive)
* non-letter characters (e.g. hyphens) should be ignored

<https://www.codewars.com/kata/is-it-an-isogram/python>

*'''*

*Created on 13 jun. 2018*

**@author:** *Usuario*

*'''*

import string

def **is\_isogram**(word):

#word = lower(word)

if word is None:

return False

word = str(word)

if word == *""*:

return False

lower = *""*

#for i in range(0, len(word)):

# lower += word[i].lower()

for ch in word:

lower += ch.lower()

frec = {}

for i in range(0,len(lower)):

if lower[i].isalpha():

if frec.get(lower[i]) != None:

frec[lower[i]] += 1

else:

frec[lower[i]] = 0

*'''*

*aux = frec.get(lower[0])*

*for key in frec.keys():*

*if frec.get(key) != aux:*

*return False*

*aux = frec.get(key)*

*return True*

*'''*

if len(list(set(frec.values()))) != 1:

return False

return True

print (is\_isogram(*"isogram"*))