**Find All The Unique Substrings**

5086% of 3224 of51[fvcproductions](https://www.codewars.com/users/fvcproductions)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/find-all-the-unique-substrings/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/find-all-the-unique-substrings/solutions/python)

[Discourse (24)](https://www.codewars.com/kata/find-all-the-unique-substrings/discuss/python)

* Add to Collection
* |
* Share this kata:

**The Problem**

You have a string and you want to find all the possible unique substrings this string has.

**Specifically, you should be *returning the total number of unique substrings*.**

**What You Have Now**

A function called getSubstrings() with the string str as a parameter.

**JavaScript Methodology | What Can Help Out**

* JavaScript has many methods that could possibly come in handy here, such as:
  + [substring()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/substring) (surprise, surprise)
  + [sort()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)
  + [.length](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length)
  + [splice()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice)
  + [.push](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push)
  + and finally [.toLowerCase()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase)

It's up to you how to figure out how to make them work in your favor!

* You also might want to define a helper function for eliminating the duplicates.

**Python Methodology | What Can Help Out**

Thanks [Unnamed](http://www.codewars.com/users/Unnamed" \t "_blank)!

* Python has some built in methods that could come in handy here, such as:
  + [.lower()](https://docs.python.org/3/library/stdtypes.html?highlight=lower#str.lower)
  + [len()](https://docs.python.org/3/library/functions.html?highlight=len#len)
  + [set()](https://docs.python.org/3/library/stdtypes.html?highlight=set#set)
  + and finally [range()](https://docs.python.org/3/library/stdtypes.html" \l "range" \t "_blank)

**Java Methodology | What Can Help Out**

Thanks [CodeVortex](http://www.codewars.com/users/CodeVortex" \t "_blank)!

* Java has many methods/classes that can help out, such as:
  + [ArrayList](http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html)
  + [List](http://docs.oracle.com/javase/7/docs/api/java/util/List.html)
  + [toLowerCase()](https://msdn.microsoft.com/en-US/library/aa989676)
  + [length()](http://www.tutorialspoint.com/java/java_string_length.htm)
  + [substring()](http://www.tutorialspoint.com/java/java_string_substring.htm)
  + [contains()](http://www.tutorialspoint.com/java/lang/string_contains.htm)
  + [add()](http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html#add(E))
  + [isEmpty()](http://www.tutorialspoint.com/java/util/arraylist_isempty.htm)
  + [size()](http://www.tutorialspoint.com/java/util/arraylist_size.htm)

**To Note**

* There should be no repeated substrings/duplicates, each substring should be **unique**.
* The substrings do not have to be actual words.
* Your solution should not be case-sensitive, meaning that the function should ignore cases and return all your strings in *lower case*.
* Your substrings can have the same characters repeated within them, *i.e. "mmm" and "ooooo"*.

**Happy coding! ^\_^**

<https://www.codewars.com/kata/find-all-the-unique-substrings/python>

def get\_substrings(s):

if(len(s) == 0): return 0

s = s.lower()

#hs = Set()

lista = []

for le in range(1, len(s)):

i = 0

while (i + le - 1 < len(s)):

#hs.add(source[i: i + le])

subs = s[i: i+ le]

#print(subs)

if not subs in lista:

lista.append(subs)

i+=1

if not s in lista: lista.append(s)

return len(lista)

print(get\_substrings("YOLO"))