**6 kyu**

**Unique In Order**

5578687% of3,3839,895 of 27,714[antrianis](https://www.codewars.com/users/antrianis)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/unique-in-order/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/unique-in-order/solutions/python)

[Forks (8)](https://www.codewars.com/kata/unique-in-order/forks/python)

[Discourse (136)](https://www.codewars.com/kata/unique-in-order/discuss/python)

* Add to Collection
* |
* Share this kata:

Implement the function unique\_in\_order which takes as argument a sequence and returns a list of items without any elements with the same value next to each other and preserving the original order of elements.

For example:

unique\_in\_order('AAAABBBCCDAABBB') == ['A', 'B', 'C', 'D', 'A', 'B']

unique\_in\_order('ABBCcAD') == ['A', 'B', 'C', 'c', 'A', 'D']

unique\_in\_order([1,2,2,3,3]) == [1,2,3]

<https://www.codewars.com/kata/unique-in-order/python>

*'''*

*Created on 14 oct. 2018*

**@author:** *Usuario*

*'''*

from test.crashers.mutation\_inside\_cyclegc import lst

*'''*

*Created on 13 jun. 2018*

**@author:** *Usuario*

*'''*

import string

import sys

def **unique\_in\_order**(iterable):

l = []

ans = []

prev = *""*

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

l.append(iterable[i])

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

if l[i] != prev:

ans.append(l[i])

prev = l[i]

return ans

print( unique\_in\_order(*"AAAABBBCCDAABBB"*))