**7 kyu**

**Palindrome Pairs**

10283% of 57100 of197[smepple](https://www.codewars.com/users/smepple)

Python

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

Details

[Solutions](https://www.codewars.com/kata/palindrome-pairs/solutions/python)

[Discourse (5)](https://www.codewars.com/kata/palindrome-pairs/discuss/python)

* Add to Collection
* |
* Share this kata:

Given a list of unique words. Find all pairs of distinct indices (i, j) in the given list so that the concatenation of the two words, i.e. words[i] + words[j] is a palindrome.

Examples:

["bat", "tab", "cat"] # [[0, 1], [1, 0]]

["dog", "cow", "tap", "god", "pat"] # [[0, 3], [2, 4], [3, 0], [4, 2]]

["abcd", "dcba", "lls", "s", "sssll"] # [[0, 1], [1, 0], [2, 4], [3, 2]]

Non-string inputs should be converted to strings.

Return an array of arrays containing pairs of distinct indices that form palindromes. Pairs should be reutrned in the order they appear in the original list.

<https://www.codewars.com/kata/palindrome-pairs/python>

from operator import itemgetter

def EsPalin(s):

i = 0

j = len(s) - 1

while(i < j):

if(s[i] != s[j]): return False

i+=1

j-=1

return True

def palindrome\_pairs(words):

# this is where the magic happens

lista = []

for i in range(0, len(words) - 1):

if(words[i] == None): words[i] = ""

for j in range(i + 1 , len(words)):

if(words[j] == None): words[j] = ""

if(EsPalin(str(words[i]) + "" + str(words[j]))):

par1 = []

par1.append(i)

par1.append(j)

lista.append(par1)

if(EsPalin(str(words[j]) + "" + str(words[i]))):

par2 = []

par2.append(j)

par2.append(i)

lista.append(par2)

return sorted(lista,key=itemgetter(0))