**6 kyu**

**Thinkful - Dictionary Drills: Multiple Modes**

14896% of58457[Grae-Drake](https://www.codewars.com/users/Grae-Drake)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/thinkful-dictionary-drills-multiple-modes/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/thinkful-dictionary-drills-multiple-modes/solutions/python)

[Forks (4)](https://www.codewars.com/kata/thinkful-dictionary-drills-multiple-modes/forks/python)

[Discourse (9)](https://www.codewars.com/kata/thinkful-dictionary-drills-multiple-modes/discuss/python)

* Add to Collection
* |
* Share this kata:

You probably know that the "mode" of a set of data is the data point that appears most frequently. Looking at the characters that make up the string "sarsaparilla" we can see that the letter "a"appears four times, more than any other letter, so the mode of "sarsaparilla" is "a".

But do you know what happens when two or more data points occur the most? For example, what is the mode of the letters in "tomato"? Both "t" and "o" seem to be tied for appearing most frequently.

Turns out that a set of data can, in fact, have multiple modes, so "tomato" has two modes: "t" and "o". It's important to note, though, that if *all* data appears the same number of times there is no mode. So "cat", "redder", and [1, 2, 3, 4, 5] do not have a mode.

Your job is to write a function modes() that will accept one argument data that is a sequence like a string or a list of numbers and return a sorted list containing the mode(s) of the input sequence. If data does not contain a mode you should return an empty list.

For example:

>>> modes("tomato")

["o", "t"]

>>> modes([1, 3, 3, 7])

[3]

>>> modes(["redder"])

[]

You can trust that your input data will always be a sequence and will always contain orderable types (no inputs like [1, 2, 2, "a", "b", "b"]).

<https://www.codewars.com/kata/thinkful-dictionary-drills-multiple-modes/python>

def modes(arr):

if len(arr) == 0: return []

diccio = {}

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

if arr[i] in diccio:

diccio[arr[i]] += 1

else:

diccio[arr[i]] = 1

if len(set(diccio.values())) == 1:

return []

m = 0

for key in diccio:

m = max(m, diccio[key])

ans = []

for key in diccio:

if diccio[key] == m:

ans.append(key)

return sorted(ans)

arr = 'redder'

print (modes(arr))