Joshua Bilsland

**Codewars Solutions**

A collection of solutions and things I have learned.

Contents

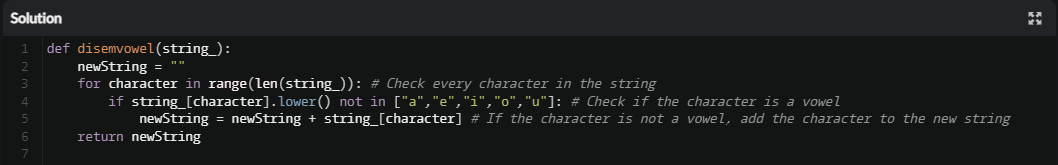
[7 Kyu Difficulty Challenges 2](#_Toc109461869)

[Disemvowel Trolls – 23/01/2022 2](#_Toc109461870)

# 7 Kyu Difficulty Challenges

## The challenge descriptionDisemvowel Trolls – 23/01/2022

In short, this challenge wants me to create a program that can take a string, remove the vowels from it, and return the new string.

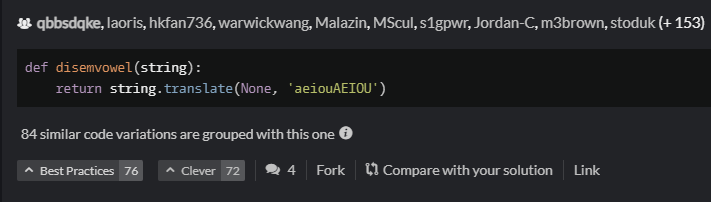
My first idea was to make use of count-controlled iteration paired with selection statements and the ‘not in’ operator. My program would go through each character in the string and check if it’s a vowel. If it is not a vowel, the character would be added to a new string variable which would be returned once every character has been checked. My solution can be seen below.

I tested the code and it worked as intended. I then submitted my solution.

Text

Description automatically generated

I then looked at some of the solutions that the community had come up with. I found a much simpler solution which makes use of the ‘translate’ method. I have not seen this method before, so I will spend some time reading about this method and try to find a way of implementing it into my programs. The solution I found can be seen below.



I also found some solutions that seemed much more complicated than my solution and have lots of unnecessary and inefficient parts to them (not saying that mine is perfect). An example can be seen below.

Text

Description automatically generated

One issue with this solution is that it checks all the vowels in their lowercase and uppercase forms via a list. This will mean that there will be ten comparisons being made for each character in the list. A better way of doing this would be to only put lower case letters in the list and use the ‘lower’ method on each character when making the comparisons. This will half the number of comparisons being made, save memory, and make the program faster.

## 