**Work Cited**

\_ Code:

import math

def round\_up(value, decimal=0):

multiplier = 10 \*\* decimal

return math.ceil(value \* multiplier) / multiplier

def round\_down(value, decimal=0):

multiplier = 10 \*\* decimal

return math.trunc(value \* multiplier) / multiplier

Source and Data:

* 2011-20 KnowledgeHut, 1st of July.

<https://www.knowledgehut.com/blog/programming/python-rounding-numbers>

* 2020, Tutorial Gateway, 1st of July.

<https://www.tutorialgateway.org/python-trunc/>

\_ Code:

Characters\_To\_Replace\_With\_Nothing = [",", "/", "[", "]", "(", ")", "{", "}", "$",

"@", "#", "%", "&", "\*", "\_", "+", "=", ">", "<", "" ", " ""]

Characters\_To\_Replace\_With\_A\_Space = ["-", "'"]

for character in Characters\_To\_Replace\_With\_Nothing:

Raw\_Text = Raw\_Text.replace(character, "", 100000000)

for character\_1 in Characters\_To\_Replace\_With\_A\_Space:

Raw\_Text = Raw\_Text.replace(character\_1, " ", 100000000)

Source and Data:

* 2020, tutorialspoint, simplyasylearning, 1st of July.

<https://www.tutorialspoint.com/python/string_replace.htm>

* Kite, your programming copilot, 1st of July.

<https://kite.com/python/answers/how-to-replace-multiple-characters-in-a-string-in-python>

Quick Notes: The math.trunc and the math.ceil function were used to obtain the results given in the prompt. If the results weren’t given to us in the prompt, I wouldn’t have used that code. The only modifications to make for a text to be correctly analyzed is to gather all the words in a text block and to make sure that there is a space after any dot.