# **Python Basics Assignments**

There are 10 assignments, each is worth 2 points.

1. Variables and Expressions - Write a Python program which prompts the user for a weight in pounds, and convert the weight to kilograms, and print out the converted weight. (1 pound = 0.4536 kg) - 2 points

Example:



2. Conditional execution - Write a Python program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table: - 2 points

|  |  |
| --- | --- |
| Score | Grade |
| >=0.95 | A+ |
| >=0.9 | A |
| >=0.85 | B+ |
| >=0.8 | B |
| >=0.75 | C+ |
| >=0.7 | C |
| >=0.65 | D+ |
| >=0.6 | D |
| <0.6 | F |

Example:



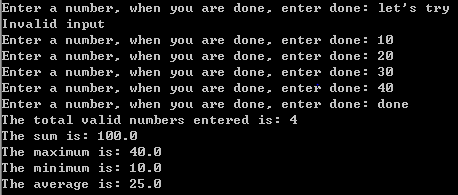




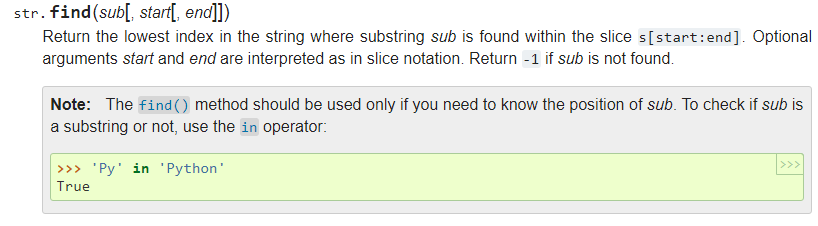
3. Functions - Rewrite the program from 2, and use a function called computegrade that takes a score as its parameter and returns a grade as a string. It should provide the same output as 2. - 2 points

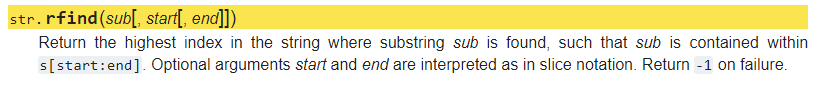
4. Iteration - Write a program which repeatedly reads numbers until the user enters “done”. Once “done” is entered, print out the count, sum, maximum, minimum, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number. - 2 points

Example:



5. String - Use str.find method () to extract an email address from a string (Assuming the email address is any continuous string without space and containing @ sign. If there are multiple emails, please extract the first one). - 2 points



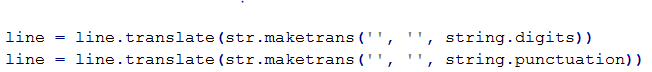


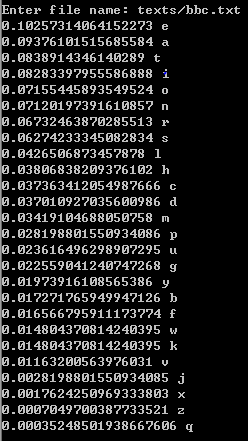
Example:



6. Write a program that reads a file and prints the letters in decreasing order of relative frequency (relative frequency = frequency/total counts of all letters). Your program should covert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, punctuation, or anything other than a-z. Compare your results with <https://en.wikipedia.org/wiki/Letter_frequency> - 2 points

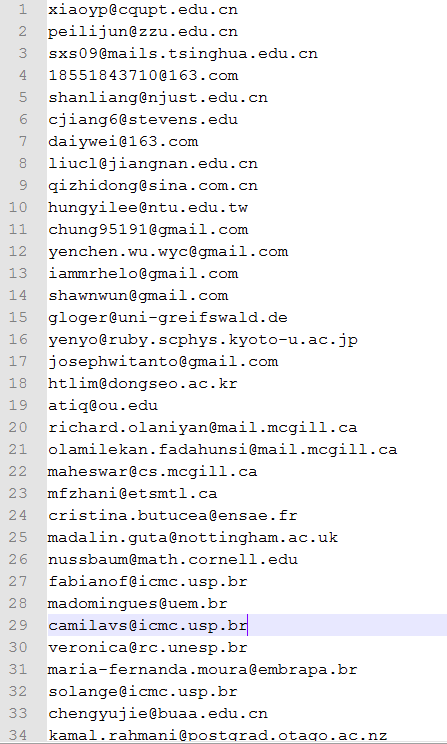
use str.translate method to clean digits and punctuations (page 115 in the Python book)





7. Regular Expression - use regular expression to extract all email address from “wos.txt”, and write it to a file “emails.txt”

emails.txt should look like:

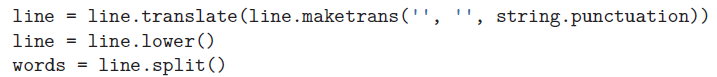


8. File and String - Write a program to find the lines that mention “Hillary” or “Trump” from the example text file. Write those lines into separate files named “Hillary.txt” and “Trump.txt” (The text files are supplied) - 2 points

9. File, String and Dictionary - Download the example text directory from Canvas. Use os.scandir(path) method to read all text files in the directory. Write a program to save words and their counts aggregated across all these text files to 'wordcounts.txt'. - 2 points

list all files in a directory use glob: https://docs.python.org/3/library/glob.html

use str.translate() method to replace punctuations as shown on page 116 of the Python book



10. File, String and Regular Expression - Download the example csv file from Canvas. Extract all hyperlinks from the “text” column and print them out. - 2 points

Use csv.reader() to read in csv file <https://docs.python.org/3/library/csv.html>

Regular expression <http://urlregex.com/>