

BASIC PYTHON PROGRAMMING EXERCISES № 1

Suman Samui

07/05/2020

Please read the instructions (given at the end) carefully. Treat this problem set as a screening test for the ML project that I have already discussed. Your sincerity will be given more preference over your skillset. So, please the instructions and submit your solutions by 9th May 2020.

Problem 1

Assume that a file with a single column containing either a character or a one-digit integer (may be positive or negative) is named info.txt (File will be provided). Each line of the file contains either a character or a one-digit integer. Write a program that calculates only the average of all the integers stored in the file (by ignoring the characters).

Problem 2

Write a function that will find all such numbers which are divisible by 9 but are not a multiple of 7, between a range of values given as an input to the program. The range should be provided as a comma-separated number (begin, end) in a line. The numbers obtained should be printed in a comma-separated sequence on a single line. The function should print output as 'Wrong input! Try Again' if the user provides a larger begin value compared to the end value.

Sample output: Enter the begin and end of the range: 12,54

Result:18,27,36,45,54

Enter the begin and end:54,12

Wrong input! Try again

Problem 3

(a) Write a function that can create a new dictionary with a flexible number of items (key-value pairs) i.e. your program should always give the option to the user to add a new key interactively. After the formation of dictionary, your function should print the dictionary.

(b) Write a program to create two dictionaries using the function for step (a) and then combine those dictionaries by adding values for common keys.

Sample output: Create the first dictionary

Enter the key: a

Enter the value:100

Want to add another key (y/n)? y

Enter the key: b

Enter the value: 200

Want to add another key (y/n)? y

Enter the key: c

Enter the value: 300

Want to add another key (y/n)? n

The created dictionary: $\{a : 100, b : 200, c : 300\}$

Create the second dictionary

Enter the key: a

Enter the value:300

Want to add another key (y/n)? y

Enter the key: b

Enter the value: 100

Want to add another key (y/n)? y

Enter the key: d

Enter the value: 400

Want to add another key (y/n)? n

The created 2nd dictionary: $\{a : 300, b : 100, d : 400\}$

The merged dictionary after adding the values of common keys:

$\{a : 400, b : 300, c : 300, d : 400\}$

Done!

Problem 4

Write a program or function which will do the following steps (hint: use pandas)

1. Read Data file: contactinfo.csv (this file will be provided) and display the first 10 entries.
2. Print the name of all the columns.
3. Print -> How many different states and cities are there in this dataset?
4. Print -> What is the most frequent city?
5. Print -> How many missing values do we have?
6. Delete the rows that contain missing values and then save the data frame into a new .csv or .xlsx file.

Problem 5

Write a program that will access a folder named *problem5* and scan through all the files. Each of the files contains a one-line string. Your program should return a metafile (.csv or .xlsx) with a name suppose fileinfo.csv which should contain two columns: *filename* and *transcript*.

Sample output:

fileinfo.csv should have a format like this:

filename	transcript
84-121123-0003	Hello Python
84-121123-0012	Nice Code
.....
.....
84-121123-00012	Doing well

Important Instructions:

1. You must create a main folder with a name format: <your-name>Codes e.g. SumanCodes
2. Set up an environment with a name format: <your-name>Env e.g. SumanEnv
3. All your scripts (one for each problem) should be within the main folder that you have created.
4. The beginning of your every script should include problem description (comment out that portion).

Listing 1: Sample Python code

```
1  '''
2  problem description: This Program implements the Fibonacci sequence ←
   analytically.
3  '''
4  from math import *
5  # define function
6  def analytic_fibonacci(n):
7      sqrt_5 = sqrt(5);
8      p = (1 + sqrt_5) / 2;
9      q = 1/p;
10     return int( (p**n + q**n) / sqrt_5 + 0.5 )
11
12 # define range
13 for i in range(1,31):
14     print analytic_fibonacci(i)
```

5. You must include necessary comments for each step of your code.
6. Last but not the least, you must save and include your env file with name format: <your-name>Env.yaml e.g. SumanEnv.yaml
7. After completing the exercises, your main folder should only contain the script files (.py file), environment file (.yaml file), and any additional file or folder that contains data.
8. Finally, compressed your folder into a .rar or .zip file and submit.