

PLEASE USE PYCHARM WHEN SOLVING THESE EXERCISES

Below, a small routine is given for calculating prime numbers between 2 and 100 using for loops. Rewrite this routine with list comprehensions and print out the primes list. (Hint: Use two comprehensions, one for "noprimes" and one for "primes" list) ¶

```
In [ ]: noprimes=[]
        primes=[]
        for i in range(2, 8):
            for j in range(i*2, 101, i):
                noprimes.append(j)

        for k in range(2,101):
            if (k not in noprimes):
                primes.append(k)
        print(primes)
```

```
In [ ]: noprimes = [j for i in range(2, 8) for j in range(i*2, 101, i)]
        primes = [x for x in range(2, 101) if x not in noprimes]
        print (primes)
```

Here is a list of temperatures in Celsius scale, [-32.2,45.3,0.2,-40.0, 1.0, 50.8].

- First, using a list comprehension, convert the list to Fahrenheit scale
- Second, do the same using lambda function

```
In [ ]: celsius=[-32.2,45.3,0.2,-40.0, 1.0, 50.8]
        fahrenheit=[ ((9.0/5.0)*t + 32) for t in celsius ]
        print(fahrenheit)
```

Do the same conversion using lambda function

```
In [ ]: celsius2=[-32.2,45.3,0.2,-40.0, 1.0, 50.8]
        fahrenheit2=list(map(lambda t2:(9.0/5.0)*t2 + 32, celsius2))
        print(fahrenheit2)
```

Change the strings to floats in the list given below and print out the same list with floats. (Hint: use a nested list comprehension)

```
[['30', '12', '5.0', '321.3','67','77'], ['11', '22', '20.6','20.89'], ['45', '0.02']]
```

```
In [ ]: s = [['30', '12', '5.0', '321.3', '67', '77'], ['11', '22', '20.6', '20.89'], ['45', '0.02']]
        f=[float(x) for x in y] for y in s
        print(f)
```

Using a nested list comprehension convert first list to the second:

1st list:

```
[['30', '-5.0', '321.3','77'], ['0.89'], ['45', '0.02']]
```

2nd list:

```
[['3', '0'], ['-', '5', '.', '0'], ['3', '2', '1', '.', '3'], ['7', '7']], [['0', '.', '8', '9']], [['4', '5'], ['0', '.', '0', '2']]]
```

```
In [ ]: s = [['30', '-5.0', '321.3', '77'], ['0.89'], ['45', '0.02']]
        d=[[z for z in x] for x in y] for y in s]
        print(d)
```

Using a list comprehension, find the numbers that are divisible by 3, 5, 7, 11 from 1 to 10000

```
In [ ]: numbers = [x for x in range(1,10000) if x % 3 == 0 and x % 5 == 0 and x % 7 == 0 and x % 11 == 0 ]
        print(numbers)
```

```
In [ ]: numbers_also = [x for x in range(1,10000) if x % 3 == 0 if x % 5 == 0 if x % 7 == 0 if x % 11 == 0 ]
        print(numbers_also)
```

Write the same comprehension using lambda function. Hint 1: Search how to use filter function in connection with lambda function.

```
In [ ]: numbers2 = list(filter(lambda x: (x%3==0 and x % 5 == 0 and x % 7 == 0 and x % 11 ==0), range(1,10000)))
        print(numbers2)
```

Write a lambda function for argument "t" where the arithmetic expression is (-2t) if t>=4 and (2t) for everything else. Evaluate the expression in the range 0 to 6

```
In [ ]: l_function = list(map(lambda t: 2*-t if t>=4 else 2*t, range(7)))
        print(l_function)
```

In this example we will write to a file and read from the same file using list comprehension

- Write the following string to a file: 'Batteries have changed a lot in the past century, but there is still work to do.'
- Close the file.
- Read the file using a list comprehension to obtain the output below. (Hint: use split function)

```
[['Batteries', 'have', 'changed', 'a', 'lot', 'in', 'the', 'past', 'century,', 'but', 'there', 'is', 'still', 'work', 'to', 'do.']]
```

```
In [ ]: my_message='Batteries have changed a lot in the past century, but there is still work to do.'
        fp=open('log.txt','w'); fp.write(my_message); fp.close
        my_list = [[word for word in text.split(" ")] for text in open('log.txt','r')]
        print(my_list)
```

A file object has a writelines() method which expects a list of strings. Search the web for how to use this method.

- Using writelines file object method and a list comprehension to write the even numbers from 0 to 100 to file.
- Each number should be on a new line
- You write this script in two lines of code

```
In [ ]: with open('file.txt', 'w') as textfile:
        textfile.writelines(str(i) + "\n" for i in range(0, 101, 2))
```

File objects are useful for passing into other libraries that know how to use them. The built-in csv (comma separated values) module is a great example. In this exercise you will use DictReader class from csv package. Search the web on how to use DictReader object.

- You will find fbi_crime_statistics.csv file in the GitHub repo. Read this file and pass it to a DictReader object.
- Using a for loop read the rows with fieldnames "Violent crime" and "Population". At each iteration, pass the values under the given field names to integer variables
- Using the integer variables, calculate ratio (violent crimes/population) and print out this ratio for each year (year is another fieldname in the input file) Hint: By default the numbers in the rows will be read as strings. Also be careful about the commas in the numbers

```
In [ ]: # First import `DictReader` class from csv
from csv import DictReader
with open("fbi_crime_statistics.csv", 'r') as f:
    # This creates a csv "DictReader" object that will let us access individual
    # fields in the csv file by name.
    dr = DictReader(f)

    # DictReader objects can be looped over just like file objects.
    for row in dr:
        # The values in this csv contain commas, so we have to strip those out
        # and then convert to integer (by default every field will be read in as
        # a string).
        violent_crimes = int(row["Violent crime"].replace(',', ''))
        population = int(row["Population"].replace(',', ''))
        violent_crime_ratio = (violent_crimes / population)

        print("The violent crime rate in", row["Year"], "was", violent_crime_ratio)
```

Write a short script to calculate and print out the circumference and area of a circle with only diameter as input. You can use math module which includes constant pi. For circumference and area calculations develop 2 functions.

- First develop the code to read the diameter interactively (input via keyboard) and print the results to screen. Hint: Interactive input is read as a string
- Second, change the code to read the diameter from a file and print the results to screen as well as to an output file. Hint: file.write() method accepts a string

```
In [ ]: # This is the script that interactively reads the input

from math import pi

def circumference(diameter):
    return pi*(float(diameter))

def area(diameter):
    return pi*(float(diameter)/2.0)**2

diameter=input("Enter the diameter of the circle:")
print(type(diameter))

print('The circumference is:', circumference(diameter))
print('The area is:', area(diameter))
```

```
In [ ]: # This is the script that reads the input from a file

from math import pi

def circumference(diameter):
    return pi*(float(diameter))

def area(diameter):
    return pi*(float(diameter)/2.0)**2

## Second part of the exercise

finput = open('circle_area.input','r')
diameter2=float(finput.readline())
finput.close

print('The circumference is:', circumference(diameter2))
print('The area is:', area(diameter2))

foutput=open('circle_area.output','w')

foutput.write('The circumference is: ')
foutput.write(str(circumference(diameter2)))
foutput.write('\n')

foutput.write('The area is: ')
foutput.write(str(area(diameter2)))

foutput.close
```

You developed a python code including functions to calculate circumference and area of a circle. Now, divide the code into 2 different parts:

- First file should be a module that includes the functions.
- Second file should be the main python code that takes the input from a file and writes the results to both screen and an output file.

```
In [ ]: # This the the module file called area.py
# Circle circumference and area module

from math import pi

def circumference(diameter):
    return pi*(float(diameter))

def area(diameter):
    return pi*(float(diameter)/2.0)**2
```

```
In [ ]: # This is the main python script

import area

finput = open('circle_area.input','r')
diameter2=float(finput.readline())
finput.close

print('The circumference is:', area.circumference(diameter2))
print('The area is:', area.area(diameter2))

foutput=open('circle_area.output','w')

foutput.write('The circumference is: ')
foutput.write(str(area.circumference(diameter2)))
foutput.write('\n')

foutput.write('The area is: ')
foutput.write(str(area.area(diameter2)))

foutput.close
```

Develop a short python script that calculates the shell area between a larger and a smaller circle.

- The code will read an input file that contains the diameter of the large and small circle. The input includes many sets of large&small circle radii.
- The code will have one main script and a module file. The main script will do the reading, writing and calling the module functions. The module file will contain the calculator for the shell area
- The code will write the output to both screen and an output file
- Your input file should look like this:

```
6 4
3.3 5.6
2 9
3.1 4.4
6 5
9 11.2
```

```
In [ ]: # Circle circumference and area module
```

```
from math import pi

def circumference(diameter):
    return pi*(float(diameter))

def area(diameter):
    return pi*(float(diameter)/2.0)**2
```

```
In [ ]: # This module calculates the shell area
```

```
def shell(area,diameter1,diameter2):
    return abs(area(diameter1)-area(diameter2))
```

```
In [ ]: # This is the main python script
```

```
from shell import shell
from area import area

finput = open('shell.input','r')
foutput=open('shell_area.output','w')

for line in finput:
    diameter1=float(line.split()[0])
    diameter2=float(line.split()[1])
    print('The shell area is:', shell(area,diameter1,diameter2))
    foutput.write('The shell area is: ')
    foutput.write(str(shell(area,diameter1,diameter2)))
    foutput.write('\n')

foutput.close()
finput.close()
```

Write a python module that includes a function to calculate the average of a given list of numbers.

- The module function should ask you to enter numbers interactively
- The module function should print out the average of the given list
- Use the module function by calling it from another python script
- Modify the module to make sure that the module can be used just by itself (i.e. by simply running the module)

```
In [ ]: # This module calculates the average of
# a list

def average():
    my_list=input("Input a series of numbers seperated with spaces:")
    numeric_list=[float(i) for i in my_list.split(" ")]
    print("The average of the given list is:", sum(numeric_list)/len(numeric_list))
    # help(sum)

if __name__ == "__main__" :
    print("__name__ is set to ", __name__)
    average()
```

```
In [ ]: # This is the basic python script to call average

import average

average.average()
```