Programming and Scripting
Lab Topic 04-Flow control

Introduction.

I would suggest that you create another folder in labs called Topic04-flow, remember to push your code to GitHub
You can save the programs you create in this lab in there.

Strings and Numbers:

1. Write a program (lab04.01-grade.py) that reads in a students percentage mark and prints out corresponding the grade (the program should check that the percentage is valid:
    - Under 40% => Fail
    - Between 40% and 49% => Pass
    - Between 50% and 59% => Merit 2
    - Between 60% and 69% => Merit 1
    - Over 70% => Distinction

```
Enter the percentage: 67
Merit2
```

Answer

```
# This program reads in
# a students percentage
# and prints out the
# corresponding grade

percentage = float(input("Enter the percentage: "))
#print (percentage)

if percentage < 0 or percentage > 100:
    # Later we will show you error handling
    # This should really throw an error
    print ("Please enter a number between 0 and 100")
elif percentage < 40: # we know it is greater than 0
    print ("Fail")
elif percentage < 50: # between 40 and 49
    print ("Pass")
elif percentage < 60: # between 50 and 59
    print ("Merit1")
elif percentage < 70: # between 60 and 69
    print ("Merit2")
else: # the only option left is betwen 70 and 100
    print ("Distinction")
```

2. In practice the percentages are rounded ie 69.5 gets rounded to 70 so is equal to a Distinction.
   How would you modify the program in 1 to deal with this?
   I see two ways.
3. Write a program (lab04.03-average.py) that keeps reading numbers until the user enters a 0.
   The program should then print out each of the numbers entered and the average of them. (Use a list)

```
enter number (0 to quit): 33
enter another (0 to quit): 34
enter another (0 to quit): 0
33
34
The average is 33.5
```

Answer

```
# This program reads in numbers until
# the user enters 0
# it will them print back out again
# and their average

numbers = []

# first number then we check if it is 0 in the while loop
number = int(input("enter number (0 to quit): "))

while number != 0:
    numbers.append(number)

    # read the next number and check if 0
    number = int(input("enter another (0 to quit): "))

for value in numbers:
    print (value)

# I want average to be a float
average = float(sum(numbers)) / len(numbers)
print ("The average is {}".format(average))
```

4. Write a program (lab04.04-student.py) that reads in students until the user enters blank in they students first name. The program should then print out all the students entered in a neat way.

```
enter firstname (blank to quit): joe
enter lastname: burke
enter firstname of next (blank to quit): mary
enter lastname: walsh
enter firstname of next (blank to quit):
here are the students you entered:
joe burke
mary walsh
```

Answer

```
# A Program that reads in students
# until the user enters a blank
# and then prints them all out again

students = []

firstname = input("enter firstname (blank to quit): ").strip()
while firstname != "":
    student = {}
    student["firstname"] = firstname
    lastname = input("enter lastname: ").strip()
    student["lastname"] = lastname
    students.append(student)
    # next student
    firstname = input("enter firstname of next (blank to
quit): ").strip()

print ("here are the students you entered:")
for currentStudent in students:
    print ("{} {}".format(currentStudent["firstname"],
currentStudent["lastname"]))
```

5. Write a program (lab04.05-topthree.py) generates 10 random numbers (0-100) , prints them out then prints out the top three.

10 random numbers	[34, 70, 48, 17, 77, 55, 68, 93, 36, 67]
The top 3 are	[93, 77, 70]

Answer

```
# This program generates 10 random numbers.
# prints them out, then
# prints out the top 3

# I will use a list to store and
# manipulate the numbers

import random
# I programming the general case
howMany    = 10
topHowMany = 3
rangeFrom  = 0
rangeto    = 100

numbers = []

for i in range(0,10):
    numbers.append(random.randint(rangeFrom,rangeto))
print ("{} random numbers\t {}".format(howMany,numbers))

# I am keeping the original list maybe I don't need to
# I got the idea to sort and split the list from stackover flow
# https://stackoverflow.com/q/32296887
topOnes = numbers.copy()
topOnes.sort(reverse = True)
print ("The top {} are \t\t {}
".format(topHowMany,topOnes[0:topHowMany]))
```