CS623 Lab-1
Introduction to Python for Multimedia

Introduction

Python is a hybrid language that has the advantages of both a general purpose programming language (e.g. C/C++) and scripting language (e.g. shell). In a scripting language, each statement can be executed independently, unlike C/C++ where you need to compile the whole code and generate a machine executable. Hence, it is easier to write programs in Python. On the other hand, general purpose languages produce more compact and faster code. However, due to increased computing power, a large number of software/algorithms are written in Python. Consequently, a large amount of code is available for re-use in terms of packages. Note that Python is also basically written in the C programming language (like Matlab).

---

2. Installation

Python comes pre-installed on Mac and Linux machines. To check the version, just open the terminal and type python -V and enter. It will take you into the python environment. Now you can start writing individual statements on the command prompt and hit enter to run the statements.

In practise, instead of writing individual statements, we write full programs in text files. The file extension used for python in py, i.e., file.py. You can put all your commands in this file and run in one shot with the following command:
$python file.py

You can write the Python programs in vi or any editor of your choice. Currently a number of popular development environments are provided by Anaconda Python distribution. If you are installing Anaconda, you can skip the rest of this section.

Installing python packages

The reusable code of python is available in the form of packages (which is a collection of modules). You have to import a package/module in order to use the functions of that package. You can make a Google search to find the package you need to import. For example, to use functions for matrix/array operations we need to import numpy package.

Before you can import a package in your program, you need to install the package using pip, which is a package manager for python. If you do not have pip installed, install pip with the following commands:

- sudo apt install python-pip [for python 2]
- sudo apt install python3-pip [for python 3]

For installing packages, run the following commands:

- pip install numpy or pip install --user numpy
- pip install matplotlib or pip install --user matplotlib
- pip install opencv-python or pip install --user opencv-python
- pip install sounddevice or pip install --user sounddevice
- pip install SoundFile or pip install --user SoundFile

For importing packages/modules, use the following syntax:

- import numpy
- import numpy as np
- import matplotlib.pyplot as plt

---

3. jupyter lab - web-based programming interface

Like all other languages, there are a number of GUI based platform for writing and debugging Python programs. Below are the instructions for one such popular platform called jupyter.

To install jupyter, run the following command:
$pip install jupyterlab or pip install --user jupyterlab [for python 2 users]
$pip3 install jupyterlab or pip3 install --user jupyterlab [for python 3 users]

Note: In Lab-2 computers pip3 is already installed, so you need to use the second command [python 3 user] for installing jupyterlab i.e. $pip3 install jupyterlab or pip3 install --user jupyterlab

To run the interface, go to the working directory using the terminal and run the following command:
$jupyter lab

Its a web-based interface, so it will open a web-browser. You can get more information about jupyter here: https://jupyterlab.readthedocs.io/en/stable/

---

4. Python Variables/Expressions

- Value assignment to variables
  - counter = 100 # An integer assignment
  - miles = 1000.0 # A floating point
  - name= "John" # A string
- Multiple Assignment
  - a = b = c = 1
  - a,b,c = 1,2,"john"
- Standard Data Types
  - Numbers: var1 = 10
  - String: str = 'Hello World!'
  - List: list = ['abcd', 786 , 2.23, 'john', 70.2]
  - Tuple(similar to list but it is read only, no updation): tuple = ('abcd', 786 , 2.23, 'john', 70.2)
  - Dictionary:

```
dict['one'] = "This is one"
dict[2]     = "This is two"
tinydict = {'name': 'john','code':6734, 'dept': 'sales'}
print dict['one']       # Prints value for 'one' key
print dict[2]           # Prints value for 2 key
print tinydict          # Prints complete dictionary
print tinydict.keys()   # Prints all the keys
print tinydict.values() # Prints all the values
```

A nice tutorial of other Python features is available here:
https://www.cse.unsw.edu.au/~en1811/python-docs/python-3.6.4-docs-pdf/tutorial.pdf

5. Basic File I/O

Numerical data is generally given in excel or csv files. There are numerous libraries that contain functions to read from excel files. One such popular library is pandas. In order to use the functions from this library, you first need to import it in your program. Below is a simple example to read height and weight data from an excel file and calculate. Extend this program to find anomalies using Z-value test. Try with different thresholds. You may also modify the data to create anomaly and see if your method detects it.

```
#Import the package that contains functions to read from excel files
import pandas as pd

#import Numpy package for numerical operations
import numpy as np

#data is a DataFrame type of variable. It is a data structure defined in pandas to store two dimensional tabular data. The top row is used to assign labels to each column. The rows are labelled 0 to n-1
data = pd.read_excel(r'height-weight.xlsx')

#convert the data into a Numpy array
data_array=data.to_numpy()

#Convert height from inches to meters
height_meters=data_array[:,1]/39.37

#Calculate BMI (kg per meter square)
bmi=data_array[:,2]/height_meters

#Print bmi values
print(bmi)
```

6. Basic Audio I/O

[Ref: https://realpython.com/playing-and-recording-sound-python/]

There are many packages that can be used to read an audio file in python. One example is given below. **Kindly note that you need to install each package before you import it.**

```
#Import sounddevice module (a module is a part/function of a package) and name it sd
import sounddevice as sd
```

```
#Import soundfile and name the module as sf
import soundfile as sf

filename = 'myfile.wav'
# Extract data and sampling rate from file using sound file package
data, fs = sf.read(filename, dtype='float32')
#play the audio using sound device
sd.play(data, fs)
status = sd.wait()  # Wait until file is done playing
```

---

7. Basic Text Read/Write

To open a text file, run the following command:

File_object = open("File_Name","Access_Mode")

E.g:

```
file1 = open("MyFile.txt","a")
file2 = open("D:\Text\MyFile2.txt","w+")
```

A simple word count program

```
file=open("sample.txt","r+")
wordcount={}
for word in file.read().split():
   if word not in wordcount:
      wordcount[word] = 1
   else:
      wordcount[word] += 1

for k,v in wordcount.items():
   print k,v
file.close();
```

---

8. Basic Image I/O - Read/Display image

For basic image I/O applications, you need the following packages: OpenCV, Matplotlib, PIL

```
#Import modules from Matplotlib
import matplotlib.image as mpimg
import matplotlib.pyplot as plt

# Read Image k1.jpg
img = mpimg.imread('k1.jpg')

#Display Image
plt.imshow(img)

#You can also read an image using opencv. To import opencv computer vision library, you must install opencv package
import cv2

# Read RGB image
img = cv2.imread('k1.jpg')

# Output img with window name as 'image'
cv2.imshow('image', img)

#To maintain output window until
# user presses a key
cv2.waitKey(0)

# To destroying present windows on screen
cv2.destroyAllWindows()
```

Drawing a circle and storing it back.

```
import numpy as np
import cv2

img = np.zeros((512,512,3), np.uint8)

#cv2.circle(img,(x-coordinate,y-coordinate), radius, (color code), thickness)

cv2.circle(img,(100, 100), 25, (0,255,0),5)

cv2.imshow('Test image',img)

cv2.imwrite('Test_gray.jpg', img)
```

Further tasks

(1) Read any four images of cats and create image collage. Then save it on the disk.
(2) Read an audio and then play it. Now we have to manipulate the audio. We will discard each alternate second of the audio. Play the manipulated audio and save it.
(3) Read a text file (you may create on your own) and count the number of word occurences. Then take top 10 most occuring words and plot histogram. Display and save the histogram.