**Pandas**

1. List to dataframe

a=[[1,2,3,4,5],[6,7,8,9,0]]

a=pd.dataframe(a,columns=['a','b'])

1. Drop duplicate

customers = customers.drop\_duplicates(subset="email", keep="first")

1. Drop none value

students.dropna(subset=['name'])

1. Rename columns

students.rename(columns={

'id': 'student\_id','first':'first\_name','last':'last\_name','age':'age\_in\_years'},

inplace=True)

1. Change type of columns

students["grade"] = students["grade"].astype(int)

**PyPDF2**

1. Open file

with open("198.Extract Text from PDF.pdf","rb") as file:

reader=PyPDF2.PdfReader(file)

1. Read pages

reader.pages

1. Extract text

page.extract\_text()

**File handling**

1. Read file

open("196.Favorite Quote Collector.txt",'r')

1. Read lines

k=f.readlines()

**Random**

1. random.choice(list)

**Pyqt**

import sys

from PyQt6.QtWidgets import QMainWindow,QLabel,QApplication,QTextEdit,QPushButton, QComboBox

1. create a window

app=QApplication(sys.argv)

window=QMainWindow()

window.setWindowTitle("voice typing notepad")

window.setGeometry(500,500,600,700)

window.show()

sys.exit(app.exec())

1. lable

lable=QLabel("voise typing notepad",window)

lable.setGeometry(10,10,400,30)

to change the text "voise typing notepad"

label.setText("no text to save!")

1. toload

QApplication.processEvents()

1. to create text box

textbox=QTextEdit(window)

to append some text

textbox.append(text)

1. button

listenbutton=QPushButton("start listening 🎤",window)

listenbutton.clicked.connect(lambda: hear(lable,textbox))

1. to exit

app.quit

1. combo box(select)

combobox=QComboBox(window)

combobox.addItems(["select your emotion","Happy","Sad","Energetic"])

get text from combobox

text=combobox.currentText()

**Speech recognize**

Each Recognizer instance has seven methods for recognizing speech from an audio source using various APIs. These are:

recognize\_bing(): Microsoft Bing Speech

recognize\_google(): Google Web Speech API

recognize\_google\_cloud(): Google Cloud Speech - requires installation of the google-cloud-speech package

recognize\_houndify(): Houndify by SoundHound

recognize\_ibm(): IBM Speech to Text

recognize\_sphinx(): CMU Sphinx - requires installing PocketSphinx

recognize\_wit(): Wit.ai

Of the seven, only recognize\_sphinx() works offline with the CMU Sphinx engine.

The other six all require an internet connection.

import speech\_recognition as sr

1. r=sr.Recognizer()
2. mic = sr.Microphone()
3. to print micphone avaliable

print(sr.Microphone.list\_microphone\_names())

1. Adjusting for ambient noise

r.adjust\_for\_ambient\_noise(source)

1. Listen

audio = r.listen(source, timeout=10)

text = r.recognize\_google(audio)

**Time**

import time

1. Time.sleep(seconds)

**Datetime**

import datetime

1. today = datetime.date.today()
2. current\_year = today.year
3. string to formate

fest\_date = datetime.datetime.strptime(f"{current\_year}-{date}", "%Y-%m-%d").date()