

```
import pyttsx3 #pip install pyttsx3

import speech_recognition as sr #pip install speechRecognition

import datetime

import wikipedia #pip install wikipedia

import webbrowser # To take information or to operate browser

import os # To open or search any software in our system

#import smtplib # To send email


# This is used for selecting the voice for using in our program


engine = pyttsx3.init('sapi5')

# SAPI5 (speech application programming interface) is an API developed by microsoft to allow the use of
speech recognition.

voices = engine.getProperty('voices')

# To print the name of the voice

# print(voices[0].id)

engine.setProperty('voice', voices[0].id)


# This function is used by the assistant to speak.

# This function gives the voice to my assistant


def speak(audio):

    engine.say(audio)

    engine.runAndWait()
```

This function is used to tell the assistant about the time so that he can wish me.

```
def wishMe():
```

```
    hour = int(datetime.datetime.now().hour)
```

```
    if hour>=0 and hour<12:
```

```
        speak("Good Morning!,Sir")
```

```
    elif hour>=12 and hour<18:
```

```
        speak("Good Afternoon!,Sir")
```

```
    else:
```

```
        speak("Good Evening!,Sir")
```

```
    speak("I am your Assistant Sir. Please tell me how may I help you")
```

This function take the command by the user to recognise it and work on it.

```
def takeCommand():
```

```
    #It takes microphone input from the user and returns string output
```

```
    r = sr.Recognizer()
```

```
with sr.Microphone() as source:
```

```
    print("Listening...")
```

```
    r.pause_threshold = 1
```

```
    audio = r.listen(source)
```

```
try:
```

```
    print("Recognizing...")
```

```
    query = r.recognize_google(audio, language='en-in')
```

```
    print(f"User said: {query}\n")
```

If the assistant does not recognise the command given by the user then it will run exception.

```
except Exception as e:
```

```
    # print(e)
```

```
    print("Say that again please...")
```

```
    return "None"
```

```
return query
```

It is used for sending the email.

```
"""def sendEmail(to, content):
```

```
    server = smtplib.SMTP('smtp.gmail.com', 587)
```

```
    server.ehlo()
```

```
    server.starttls()
```

```
    server.login('youremail@gmail.com', 'your-password')
```

```
server.sendmail('youremail@gmail.com', to, content)

server.close()"""
```

```
# Here main program starts.
```

```
if __name__ == "__main__":
```

```
    wishMe()
```

```
    while True:
```

```
        # if 1:
```

```
            query = takeCommand().lower()
```

```
            # Logic for executing tasks based on query
```

```
            if 'wikipedia' in query:
```

```
                speak('Searching in Wikipedia...')
```

```
                query = query.replace("wikipedia", "")
```

```
                results = wikipedia.summary(query, sentences=10)
```

```
                speak("According to Wikipedia")
```

```
                print(results)
```

```
                speak(results)
```

```
            elif 'open youtube' in query:
```

```
                webbrowser.open("youtube.com")
```

```
                speak("opening youtube")
```

```
            elif 'open google' in query:
```

```
webbrowser.open("google.com")
```

```
speak("opening Google")
```

```
elif 'open stackoverflow' in query:
```

```
webbrowser.open("stackoverflow.com")
```

```
speak("opening Stackoverflow")
```

```
# To quit our program here we use exit function by giving quit, bye or exit command.
```

```
elif 'quit' in query or 'bye' in query or 'exit' in query:
```

```
speak("Quitting Sir. Thanks for your time")
```

```
exit()
```

```
# By giving open code command we can open our coding platform where i usually code.
```

```
elif "open code" in query:
```

```
codePath= "C:\\Users\\This PC\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
```

```
os.startfile(codePath)
```

```
speak("opening path")
```

```
# By giving time command we can ask about the time by the assistant.
```

```
elif 'the time' in query:
```

```
strTime = datetime.datetime.now().strftime("%H:%M:%S")
```

```
print(strTime)
```

```
speak(f"Sir, the time is {strTime}")
```

elif "open chrome" in query:

```
cpath= "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe"
```

```
os.startfile(cpath)
```

```
speak("opening chrome")
```

elif 'how are you' in query:

```
speak("I am fine, Thank you")
```

```
speak("How are you, Sir")
```

elif 'fine' in query or "good" in query:

```
speak("It's good to know that your fine")
```

elif "what's your name" in query or "What is your name" in query:

```
speak("My friends call me")
```

```
speak("Tushar's assistant")
```

```
print("My friends call me Tushar's assistant")
```

elif "who made you" in query or "who created you" in query:

```
print("I have been created by Tushar.")
```

```
speak("I have been created by Tushar.")
```

elif "who are you" in query:

```
speak("I am your virtual assistant created by Tushar")
```

elif "open powerpoint" in query:

Path= "C:\\Program Files\\Microsoft Office\\root\\Office16\\POWERPNT.EXE"

os.startfile(Path)

speak("opening powerpoint")

elif "open word" in query:

wPath= "C:\\Program Files\\Microsoft Office\\root\\Office16\\WINWORD.EXE"

os.startfile(wPath)

speak("opening Word")

To play music from your internal directeries

```
"""elif 'play music' in query:
```

```
    music_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'
```

```
    songs = os.listdir(music_dir)
```

```
    print(songs)
```

```
    os.startfile(os.path.join(music_dir, songs[0]))
```

```
elif 'send email ' in query:
```

```
    try:
```

```
        speak("What should I say?")
```

```
        content = takeCommand()
```

```
        to = "harryyourEmail@gmail.com"
```

```
        sendEmail(to, content)
```

```
        speak("Email has been sent!")
```

```
    except Exception as e:
```

```
        print(e)
```

```
        speak("Sorry my friend harry bhai. I am not able to send this email")
```

```
"""
```