Creating your Personal Virtual Assistant

Objective: Our main task is to learn Python and create your very own virtual assistant, such as an alternative to Siri. This project will not only enhance your programming skills but also provide you with a valuable tool that can simplify your daily tasks and interactions with technology.

The reasons why creating your own virtual assistant using Python are as follows:

1. **Practical Application:** Virtual assistants have become an integral part of our lives, helping us with various tasks such as setting reminders, answering questions, playing music, and even controlling smart devices. By building your own virtual assistant, you'll gain hands-on experience in developing a practical application that can be customized to meet your specific needs.

2. **Language Processing and AI:** One of the most exciting aspects of building a virtual assistant is delving into natural language processing and artificial intelligence. You'll explore techniques to parse and understand human language, enabling your assistant to comprehend and respond to user queries accurately. This knowledge will empower you to work on other AI-related projects in the future.

3. **Creativity and Customization:** As you progress with your virtual assistant, you'll have the freedom to add unique features and functionalities. You can tailor the assistant's responses, design a user-friendly interface, integrate it with external services and APIs, and even experiment with voice recognition capabilities. Your creativity will be the driving force behind making your assistant stand out.

4. **Collaboration and Learning:** This project provides an excellent opportunity for teamwork and collaboration. You can work in groups, sharing knowledge and skills, and collectively tackle complex problems. Collaborative learning fosters creativity, enhances communication skills, and exposes you to diverse perspectives, all of which are crucial in an engineering environment.

By embarking on this project, you'll not only expand your programming horizons but also cultivate skills that are highly sought after in the job market. Companies across various industries are increasingly integrating virtual assistants into their products and services, making your expertise in this area highly valuable.

After completing this project, you will enable your "Personal Virtual Assistant" to understand natural language voice commands and completes tasks for you.

Setup your Working Environment

1. Installing Python IDE: PyCharm

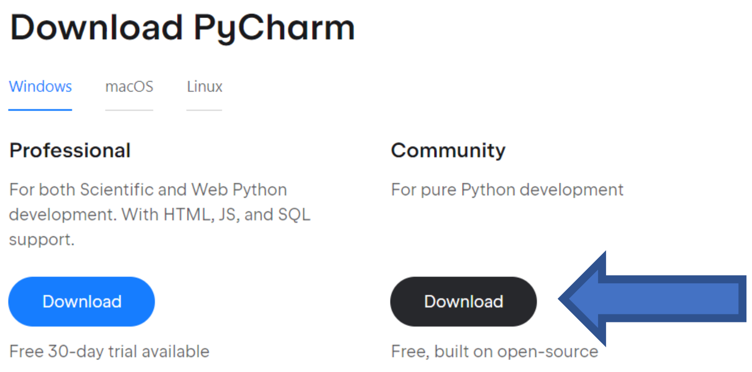
PyCharm is a widely used Python Integrated Development Environment (IDE) providing smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactorings essential tools for Python developers.

Go to PyCharm website using the link below.

https://www.jetbrains.com/pycharm/

Click on DOWNLOAD.

Dowload the community version by clicking on Download on the right.



Download PyCharm

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Free 30-day trial available

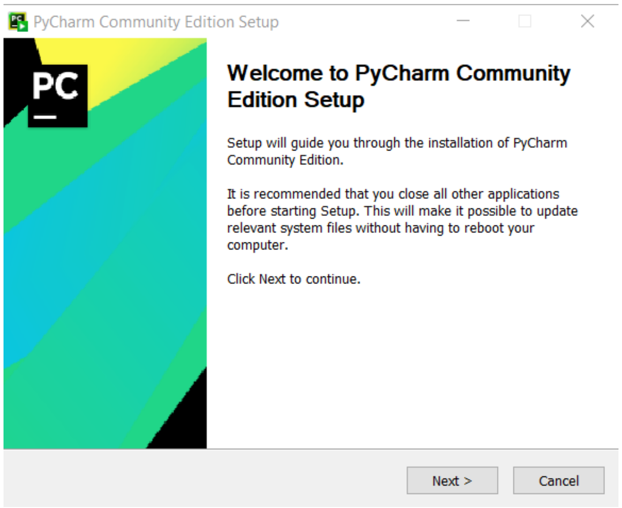
Community

For pure Python development

Free, built on open-source

Open it after downloaded

Click on Next twice

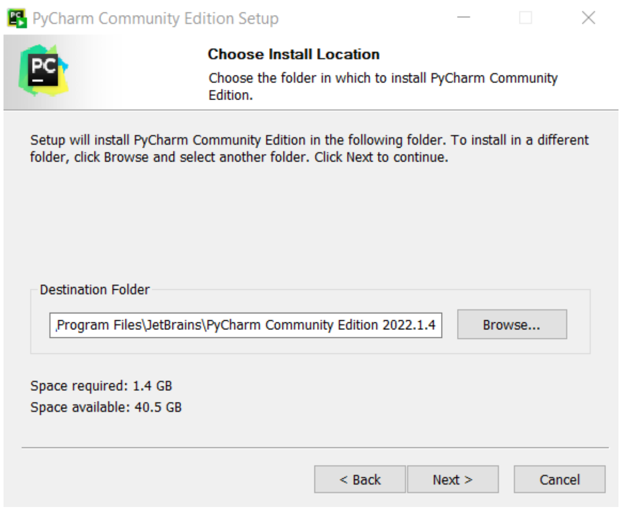

PyCharm Community Edition Setup

Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.


PyCharm Community Edition Setup

Choose Install Location

Choose the folder in which to install PyCharm Community Edition.

Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

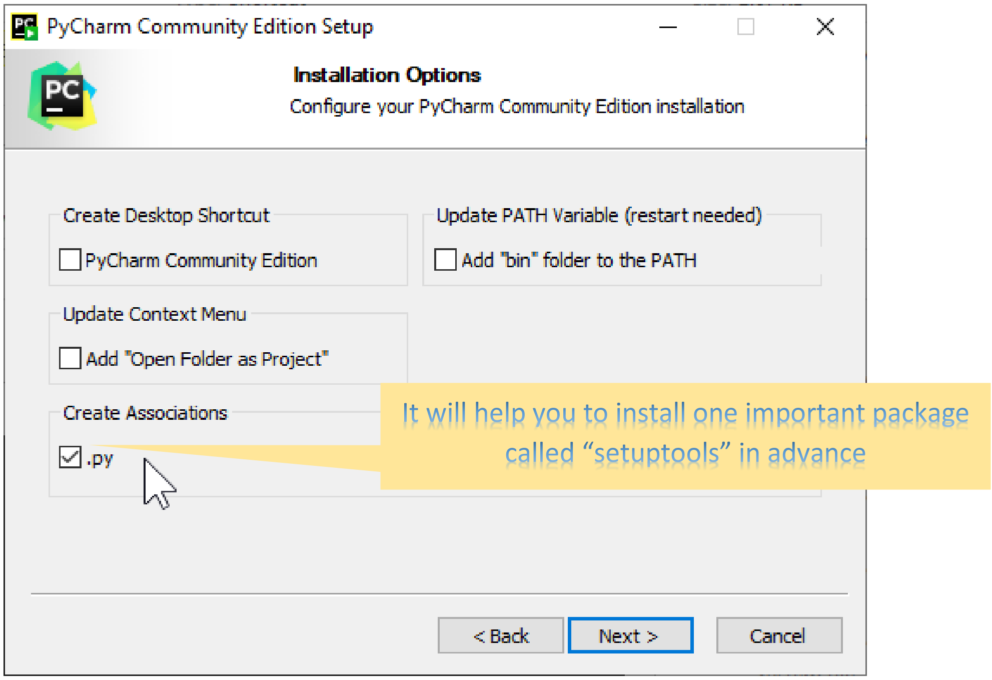
Destination Folder: Program Files\JetBrains\PyCharm Community Edition 2022.1.4

Space required: 1.4 GB

Space available: 40.5 GB

.

Enable the association of .py file with PyCharm


PyCharm Community Edition Setup

Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

□ PyCharm Community Edition

Update PATH Variable (restart needed)

□ Add "bin" folder to the PATH

Update Context Menu

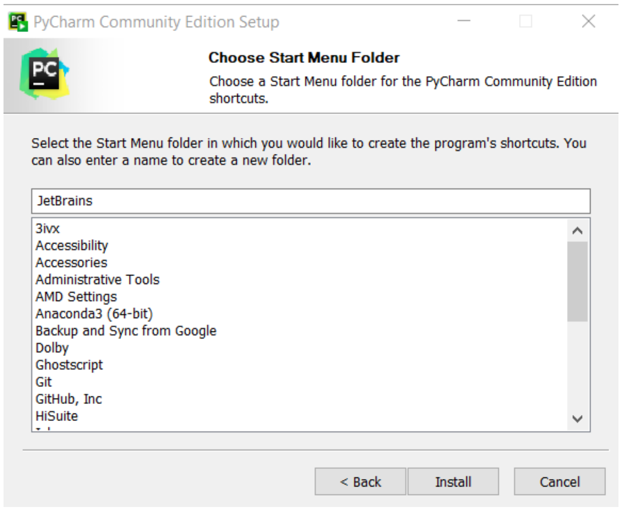
□ Add "Open Folder as Project"

Create Associations

☑ .py

It will help you to install one important package called "setuptools" in advance

Click on Install. It may take some time.



PyCharm Community Edition Setup

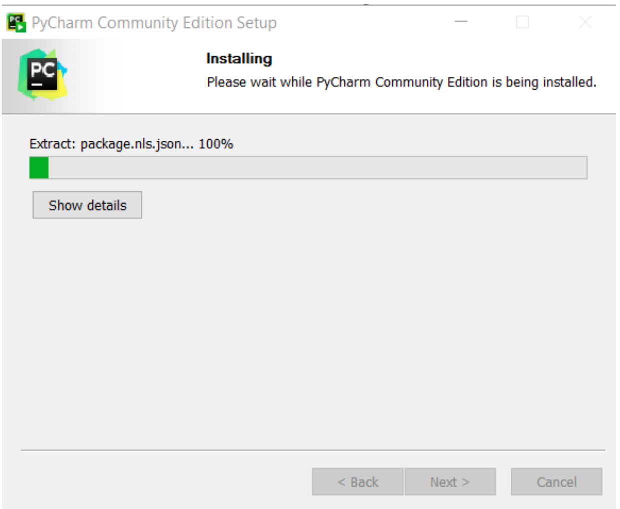
Choose Start Menu Folder

Choose a Start Menu folder for the PyCharm Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

3ivx
Accessibility
Accessories
Administrative Tools
AMD Settings
Anaconda3 (64-bit)
Backup and Sync from Google
Dolby
Ghostscript
Git
GitHub, Inc
HiSuite



PyCharm Community Edition Setup

Installing

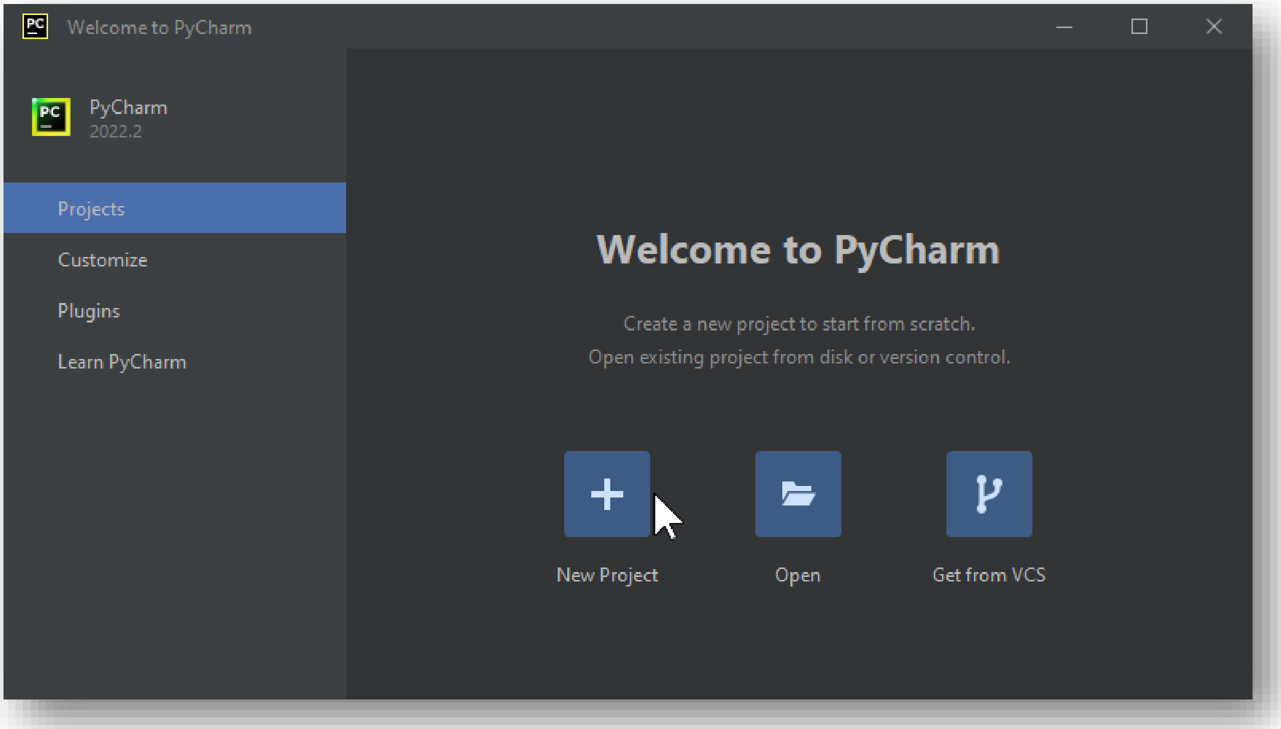
Please wait while PyCharm Community Edition is being installed.

Extract: package.nls.json... 100%

Finally, click on Finish.

When you open Pycharm, it may take some time to load.

Click on New Project.



Welcome to PyCharm

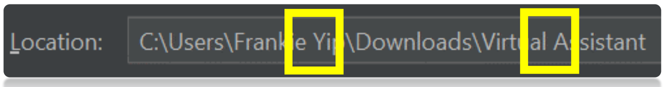
PyCharm
2022.2

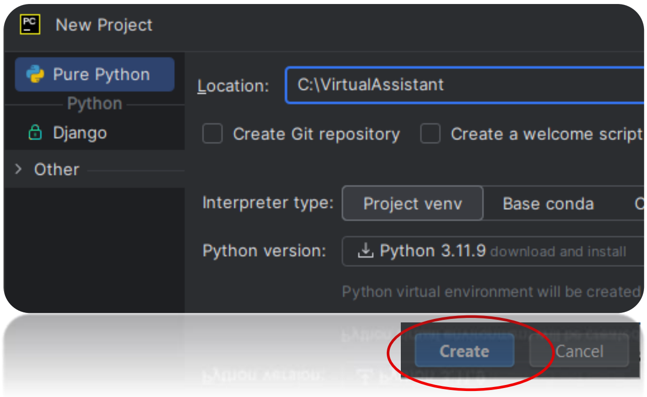
Welcome to PyCharm

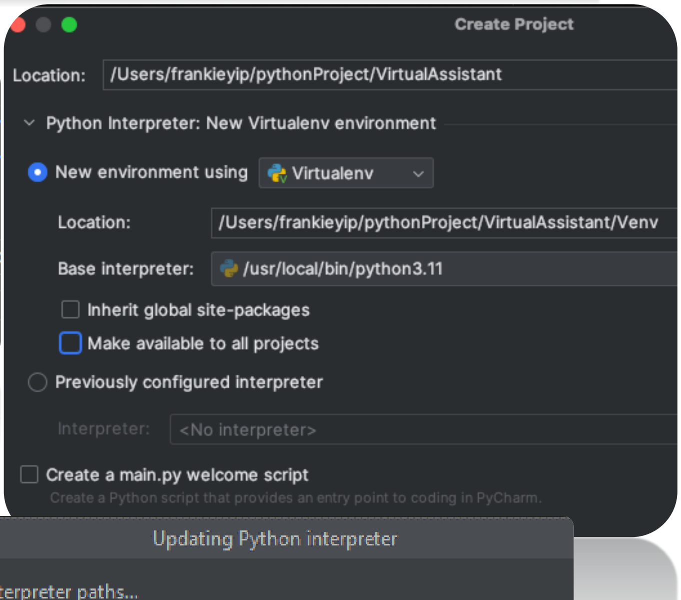
Create a new project to start from scratch.
Open existing project from disk or version control.

You may name your project e.g. "VirtualAssistant". Then click on Create and wait.

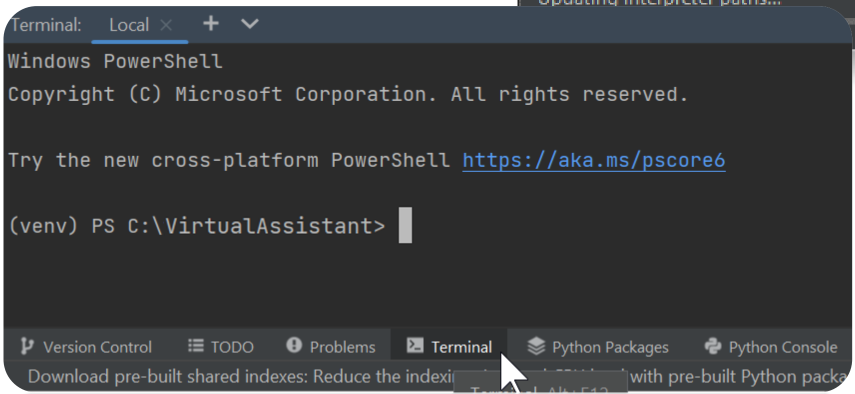
(Path/Filename should not have any space) Wrong Wrong






Create Project
Location:
/Users/frankieyip/pythonProject/VirtualAssistant
Python Interpreter: New Virtualenv environment
● New environment using
Virtualenv
Location:
/Users/frankieyip/pythonProject/VirtualAssistant/Venv
Base interpreter:
/usr/local/bin/python3.11
□ Inherit global site-packages
□ Make available to all projects
○ Previously configured interpreter
Interpreter:
<No interpreter>
□ Create a main.py welcome script
Create a Python script that provides an entry point to coding in PyCharm.
Updating Python interpreter
Updating interpreter paths...

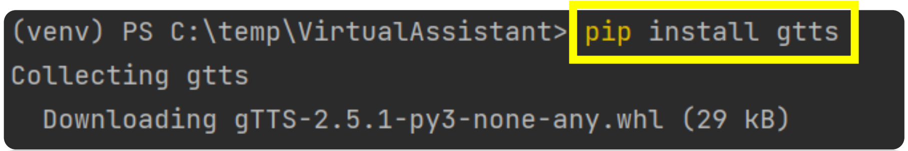
Click the "Open Terminal" button.


Terminal:
Local
Version Control
TODO
Problems
Terminal
Python Packages
Python Console
Download pre-built shared indexes: Reduce the indexi... with pre-built Python packa

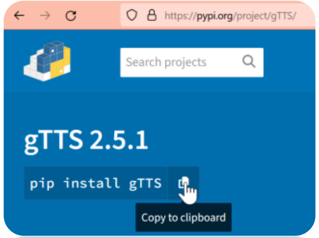
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

(venv) PS C:\VirtualAssistant>
```

Task 1: Installation of the package of Google Text-to-Speech: https://pypi.org/project/gtts/

It is a package that can use python code to convert text to speech and save to mp3 file.
Paste it in the terminal and run the code.

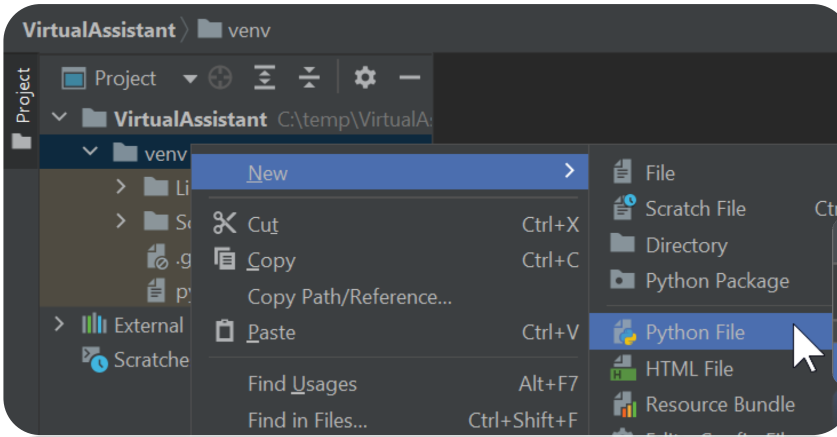


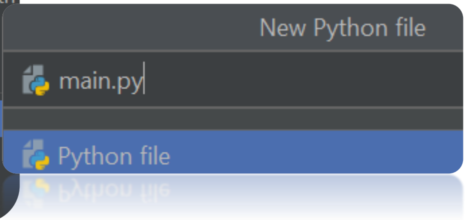
```
(venv) PS C:\temp\VirtualAssistant> pip install gtts
Collecting gtts
  Downloading gTTS-2.5.1-py3-none-any.whl (29 kB)
```



It will take some time for the installment.
When it's done, you should see Successfully installed.

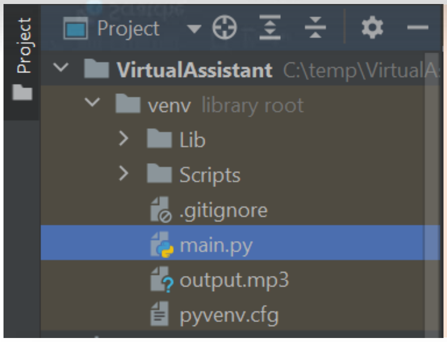
Now, you have downloaded and installed the package successfully and start try python code.



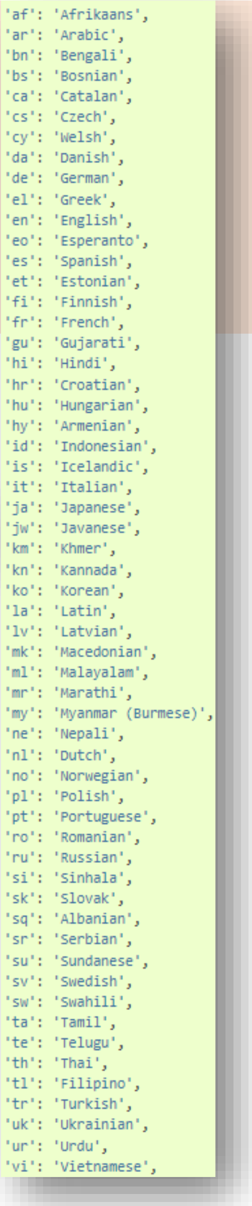


By default, Python interpreter will run “main.py” automatically.

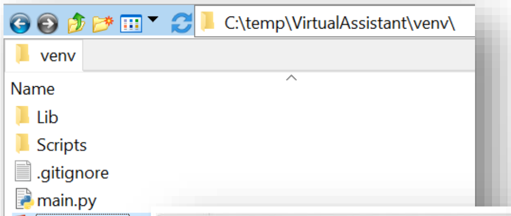
Click the “main.py” file in the side menu and try to input the following code: [Click ▶ to run]



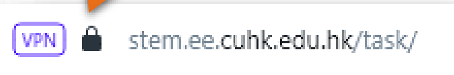
```
from gtts import gTTS

# Convert text to speech
text = "Nice to meet you dummy!"
tts = gTTS(text, lang='en')

# Save the speech as an audio file
tts.save("output.mp3")
```



```
'af': 'Afrikaans',
'ar': 'Arabic',
'bn': 'Bengali',
'bs': 'Bosnian',
'ca': 'Catalan',
'cs': 'Czech',
'cy': 'Welsh',
'da': 'Danish',
'de': 'German',
'el': 'Greek',
'en': 'English',
'eo': 'Esperanto',
'es': 'Spanish',
'et': 'Estonian',
'fi': 'Finnish',
'fr': 'French',
'gu': 'Gujarati',
'hi': 'Hindi',
'hr': 'Croatian',
'hu': 'Hungarian',
'hy': 'Armenian',
'id': 'Indonesian',
'is': 'Icelandic',
'it': 'Italian',
'ja': 'Japanese',
'jw': 'Javanese',
'km': 'Khmer',
'kn': 'Kannada',
'ko': 'Korean',
'la': 'Latin',
'lv': 'Latvian',
'mk': 'Macedonian',
'ml': 'Malayalam',
'mr': 'Marathi',
'my': 'Myanmar (Burmese)',
'ne': 'Nepali',
'nl': 'Dutch',
'no': 'Norwegian',
'pl': 'Polish',
'pt': 'Portuguese',
'ro': 'Romanian',
'ru': 'Russian',
'si': 'Sinhala',
'sk': 'Slovak',
'sq': 'Albanian',
'sr': 'Serbian',
'su': 'Sundanese',
'sv': 'Swedish',
'sw': 'Swahili',
'ta': 'Tamil',
'te': 'Telugu',
'th': 'Thai',
'tl': 'Filipino',
'tr': 'Turkish',
'uk': 'Ukrainian',
'ur': 'Urdu',
'vi': 'Vietnamese',
```

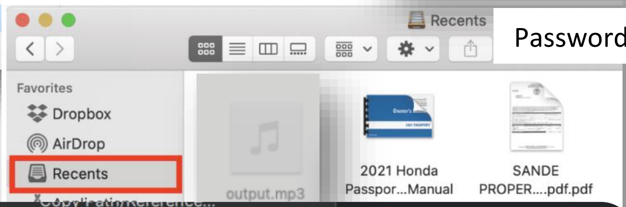


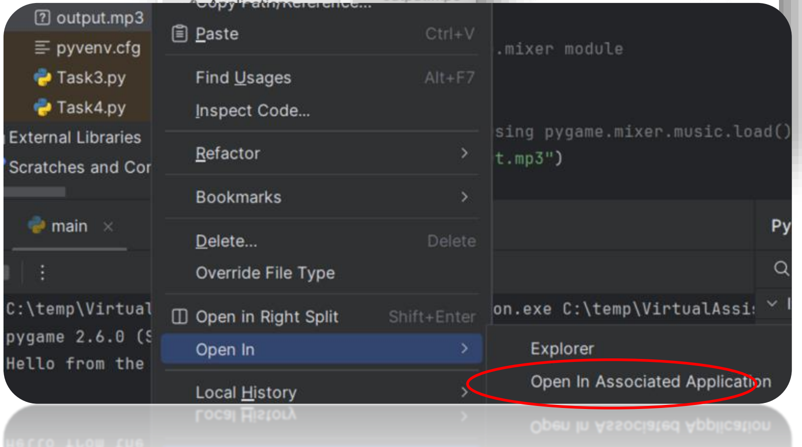
Wanna copy this set of code!!!



stem.ee.cuhk.edu.hk/task/

Password: VPA

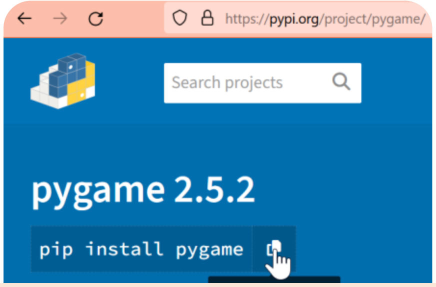




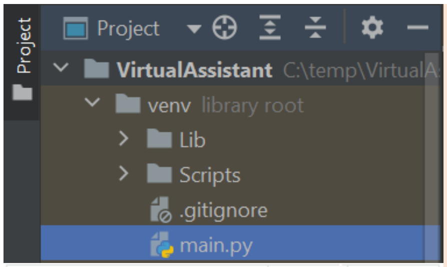
```
.mixer module
sing pygame.mixer.music.load(
t.mp3")
on.exe C:\temp\VirtualAssi
C:\temp\Virtual
pygame 2.6.0 (S
Hello from the
```

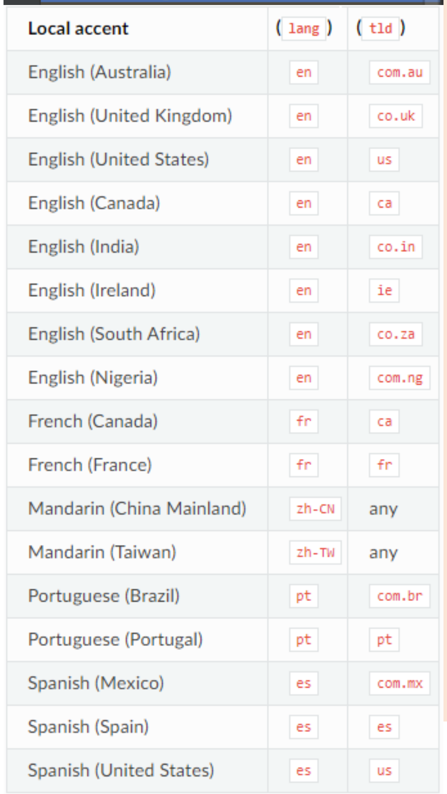
Task 2: Automatically play the sound file without the need to use external media player

"pygame" package is needed to installed to play the speech file generated by gTTS



```
pip install pygame
```

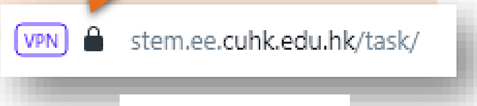




Local accent	(lang)	(tld)
English (Australia)	en	com.au
English (United Kingdom)	en	co.uk
English (United States)	en	us
English (Canada)	en	ca
English (India)	en	co.in
English (Ireland)	en	ie
English (South Africa)	en	co.za
English (Nigeria)	en	com.ng
French (Canada)	fr	ca
French (France)	fr	fr
Mandarin (China Mainland)	zh-CN	any
Mandarin (Taiwan)	zh-TW	any
Portuguese (Brazil)	pt	com.br
Portuguese (Portugal)	pt	pt
Spanish (Mexico)	es	com.mx
Spanish (Spain)	es	es
Spanish (United States)	es	us

```
from gtts import gTTS
from pygame import time, mixer

stop_Speaking = False
# Convert text to speech using gtts and save it as an audio file
command = " Nice to meet you dummy!"
tts = gTTS(command, lang='en', tld='com.au')
tts.save("output.mp3")        # tld means Top Level Domain

# Initialize the pygame.mixer module
mixer.init()

# Load the audio file using pygame.mixer.music.load()
mixer.music.load("output.mp3")

# Play the audio file using pygame.mixer.music.play()
mixer.music.play()

#Give pygame enough time to load and start playing the audio.
time.delay(3000)

# Cleanup the loading of file
mixer.quit()
stop_Speaking = True
```

Wanna copy this set of code!!!



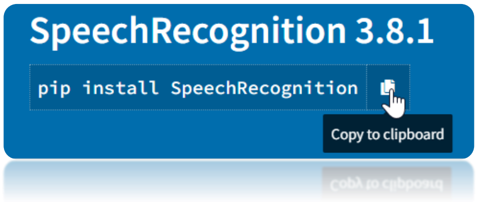
Password: VPA

Is it professional to wait for 3000ms to play the speech file?

```
# Try to check whether the music playing is busy or not by replacing "time.delay()"

    mixer.music.get_busy():
```

Two codes should be added

Task 3: Use Speech Recognition package to read and recognize user speech command



SpeechRecognition 3.8.1

```
pip install SpeechRecognition
```

However, for the "SpeechRecognition" package, it needs to use pyAudio library to execute.

Thus, you need to install "pyAudio" package, too.

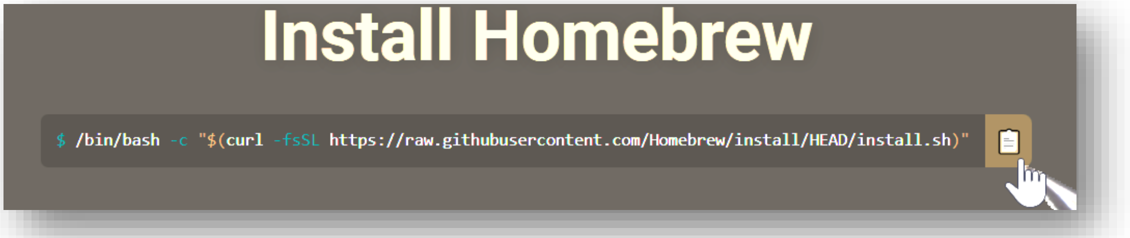
For Windows user,

Simply use "pip install pyaudio" to install

For MacOS user,

In built-in terminal of MacOS,

Firstly, visit https://brew.sh to install Homebrew (Software Manager of MacOS)



Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Once you got a fatal error: Could not resolve HEAD to a revision, type the command:

"git -C $(brew --repository homebrew/core) checkout master" and try to install again

- uninstall the PyAudio & portaudio from you system by the following commands:

  "brew uninstall portaudio" 🡐 Homebrew is a software manager of MacOS

- Secondly, update the Homebrew by using

  "brew update"

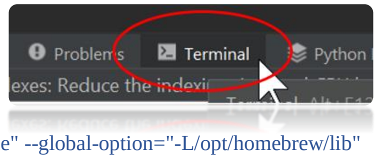
- Thirdly, update both wheel and setuptools by using

  "python3 -m pip install --upgrade pip setuptools wheel"

- Then, install the latest version of portaudio:

  "brew install portaudio --HEAD"

- Finally, go back to Terminal of Pycharm, install PyAudio:



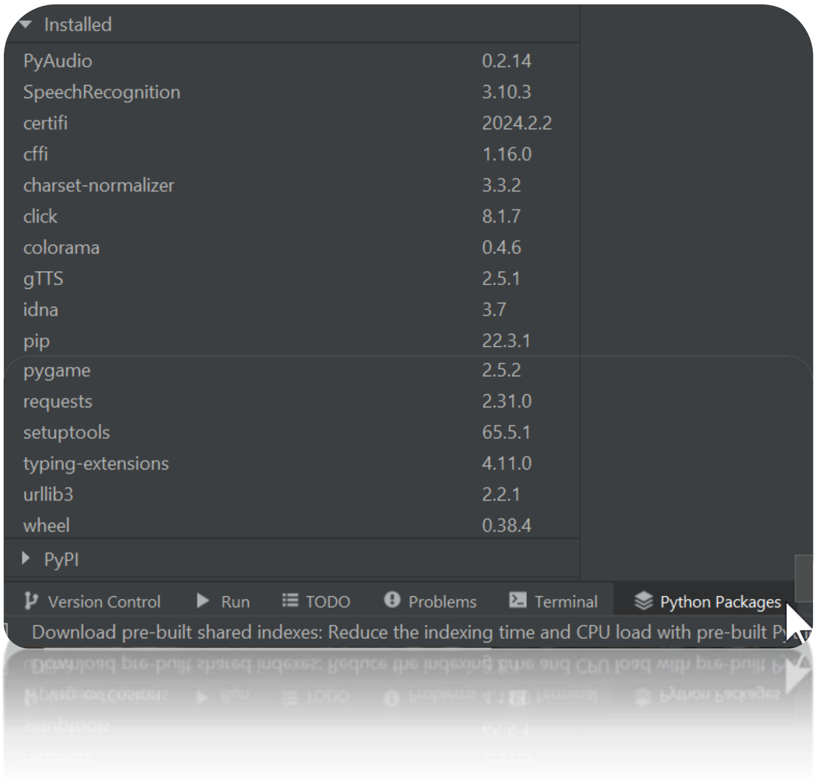
```
pip install pyaudio --global-option="build_ext" --global-option="-I/opt/homebrew/include" --global-option="-L/opt/homebrew/lib"
```

Click the tab of

"Python Packages"

to see what did you install

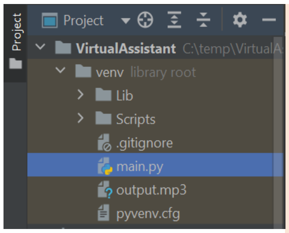
See whether you have installed successfully those four essential packages…



PyAudio	0.2.14
SpeechRecognition	3.10.3
certifi	2024.2.2
cffi	1.16.0
charset-normalizer	3.3.2
click	8.1.7
colorama	0.4.6
gTTS	2.5.1
idna	3.7
pip	22.3.1
pygame	2.5.2
requests	2.31.0
setuptools	65.5.1
typing-extensions	4.11.0
urllib3	2.2.1
wheel	0.38.4

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built P

… Try to use below code to get your voice signal



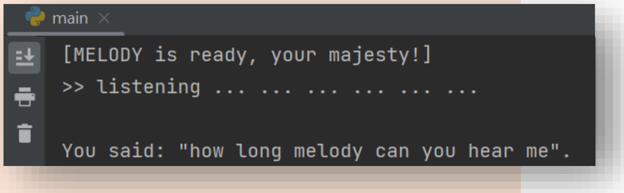
```
Project
VirtualAssistant C:\temp\VirtualA
  venv library root
    Lib
    Scripts
    .gitignore
    main.py
    output.mp3
    pyvenv.cfg
```

```
import speech_recognition as sr
import threading

assistant_name = 'X_x_x_x_x_X'
listener = sr.Recognizer()

# Shared variable to signal when to stop monitoring
stop_Speaking = True
stop_monitoring = False

# Shared variable to signal when to stop monitoring
def monitor_audio(source):
    # Continuously monitor the audio while it's being received
    silent = True
    while not stop_monitoring:
        try:
            with sr.Microphone() as monitor:
                # Listen to a short segment of audio to check if speech is present
                audio = listener.listen(monitor, phrase_time_limit=0.5)
                if audio:
                    if silent:
                        silent = False
                        print(">> listening ...", end="")
                    else:
                        print(" ...", end="")
        except KeyboardInterrupt:
            break

while True:
    if stop_Speaking:
        try:
            with sr.Microphone() as source:
                listener.adjust_for_ambient_noise(source)

                print("\n[" + assistant_name.upper() + ' is ready, your majesty!]')

                # Create a separate thread to monitor the audio
                stop_monitoring = False
                audio_thread = threading.Thread(target=monitor_audio, args=(source,))
                audio_thread.start()

                # Listen for the user's speech
                voice = listener.listen(source)

                # Stop monitoring thread
                stop_monitoring = True

                if voice:
                    print("\n")
                    translated_voice = listener.recognize_google(voice).lower()
                    print('You said: "' + translated_voice + '".')
                else:
                    continue

        except sr.UnknownValueError:
            print('You didn’t say anything or Unable to recognize your speech.')

        except sr.RequestError as e:
            print('Error occurred during speech recognition:', str(e))
```

Wanna copy this set of code!!!

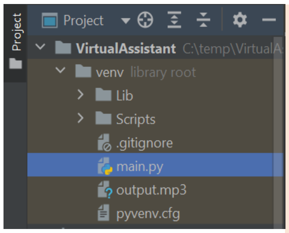
VPN stem.ee.cuhk.edu.hk/task/

Password: VPA

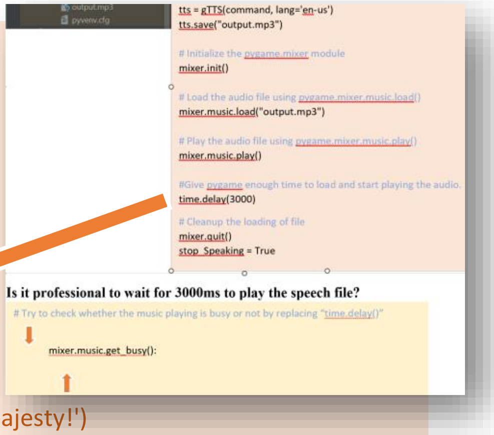


```
[MELODY is ready, your majesty!]
>> listening ... ... ... ... ... ...

You said: "how long melody can you hear me".
```

Task 4: Combine all “Text-to-Speech” & “Speech Recognition” together by define functions



```
from gtts import gTTS
from pygame import mixer
import speech_recognition as sr
import threading

assistant_name = "X_x_x_x_x_X"
listener = sr.Recognizer()

stop_Speaking = True
stop_monitoring = False

def speak(command):
```



```
tts = gTTS(command, lang='en-us')
tts.save("output.mp3")

# Initialize the pygame.mixer module
mixer.init()

# Load the audio file using pygame.mixer.music.load()
mixer.music.load("output.mp3")

# Play the audio file using pygame.mixer.music.play()
mixer.music.play()

#Give pygame enough time to load and start playing the audio.
time.delay(3000)

# Cleanup the loading of file
mixer.quit()
stop_Speaking = True
```

Is it professional to wait for 3000ms to play the speech file?

```
# Try to check whether the music playing is busy or not by replacing "time.delay()"

    mixer.music.get_busy():
```

```
speak(assistant_name.upper() + ' is ready, your majesty!')

# Shared variable to signal when to stop monitoring
def monitor_audio(source):
    # Continuously monitor the audio while it's being received
    silent = True
    while not stop_monitoring:
        try:
            with sr.Microphone() as monitor:
                # Listen to a short segment of audio to check if speech is present
                audio = listener.listen(monitor, phrase_time_limit=0.5)
                if audio:
                    if silent:
                        silent = False
                        print(">> listening ...", end="")
                    else:
                        print(" ...", end="")
        except KeyboardInterrupt:
            break

while True:
    if stop_Speaking:
        try:
            with sr.Microphone() as source:
                listener.adjust_for_ambient_noise(source)

                print("\n[" + assistant_name.upper() + ' is ready, your majesty!]')

                # Create a separate thread to monitor the audio
                stop_monitoring = False
                audio_thread = threading.Thread(target=monitor_audio, args=(source,))
                audio_thread.start()

                voice = listener.listen(source)
                stop_monitoring = True

                if voice:
                    print("\n")
                    translated_voice = listener.recognize_google(voice).lower()
                    print('You said: "' + translated_voice + '".')
                    speak('You said: "' + translated_voice + '".')
                else:
                    continue

        except sr.UnknownValueError:
            print('You didn’t say anything or Unable to recognize your speech.')

        except sr.RequestError as e:
            print('Error occurred during speech recognition:', str(e))
```

Task 5: Play specific "YouTube" Videos by using your instruction

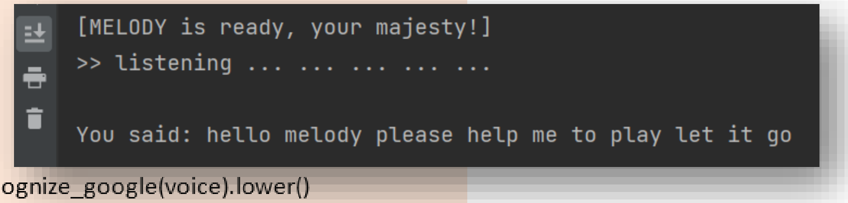
Installation of the package of pywhatkit: https://pypi.org/project/pywhatkit/

pywhatkit 5.4

```
pip install pywhatkit
```

It is a package that we can use python code to send something to others using whatsapp or play YouTube videos.

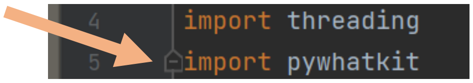
```
while True:
   if stop_Speaking:
      try:
         with sr.Microphone() as source:
            listener.adjust_for_ambient_noise(source)

            print("\n[" + assistant_name.upper() + ' is ready, your majesty!]')

            # Create a separate thread to monitor the audio
            stop_monitoring = False
            audio_thread = threading.Thread(target=monitor_audio, args=(source,))
            audio_thread.start()

            voice = listener.listen(source)
            stop_monitoring = True

            if voice:
               print("\n")
               translated_voice = listener.recognize_google(voice).lower()
               print('You said: "' + translated_voice + '".')
               speak('You said: "' + translated_voice + '".')
            else:
               continue

      except sr.UnknownValueError:
         print('You didn't say anything or U

      except sr.RequestError as e:
         print('Error occurred during speec
```



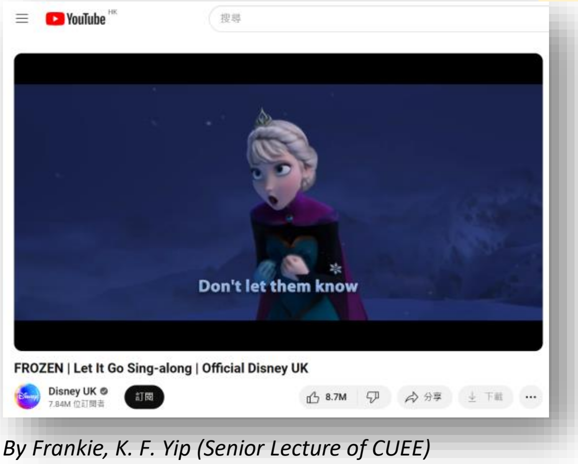
```
[MELODY is ready, your majesty!]
>> listening ... ... ... ... ...

You said: hello melody please help me to play let it go
```

```
if assistant_name.lower() in translated_voice:

   # Find the index of assistant_name in translatedVoice
   index = translated_voice.lower().index(assistant_name.lower())

   # Extract the text after assistant_name
   command = translated_voice[index + len(assistant_name):].strip()

   if 'play' in command:

      song = command[command.index('play'):].strip()

      speak("Got it, I'll help you to " + song + ', your majesty!')

      pywhatkit.playonyt(song)
```

Remember to import pywhatkit



```
import threading
import pywhatkit
```

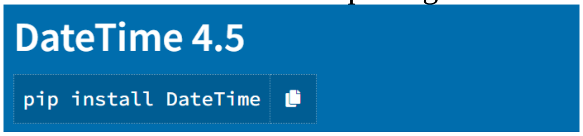


Try on your own:

Asking: "Assistant Name you set,
play any song you like "

Other Packages for you to try:

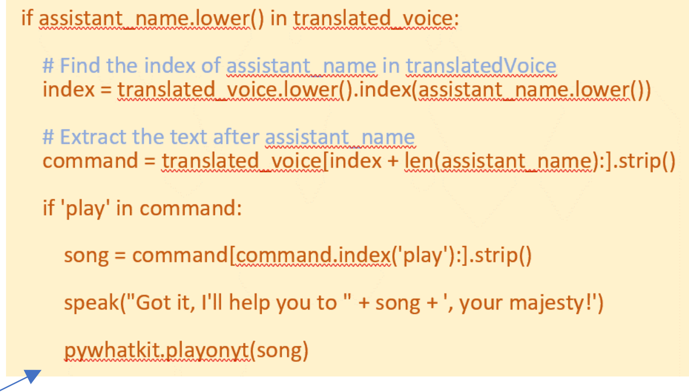
Task 6: Installation of the package of DateTime: https:// pypi.org/project/DateTime/



DateTime 4.5

```
pip install DateTime
```

It is a package that we can use python code to get the real time data.



```
if assistant_name.lower() in translated_voice:

    # Find the index of assistant_name in translatedVoice
    index = translated_voice.lower().index(assistant_name.lower())

    # Extract the text after assistant_name
    command = translated_voice[index + len(assistant_name):].strip()

    if 'play' in command:

        song = command[command.index('play'):].strip()

        speak("Got it, I'll help you to " + song + ', your majesty!')

        pywhatkit.playonyt(song)
```

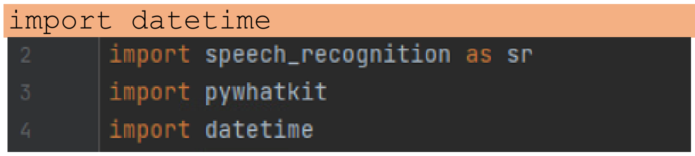
This function will tell you the time,

Paste below to the code after

```
    elif 'time' in command:
        time = datetime.datetime.now().strftime('%I:%M %p')
        speak('Current time is ' + time)
```

Remember to add the following code on the top:

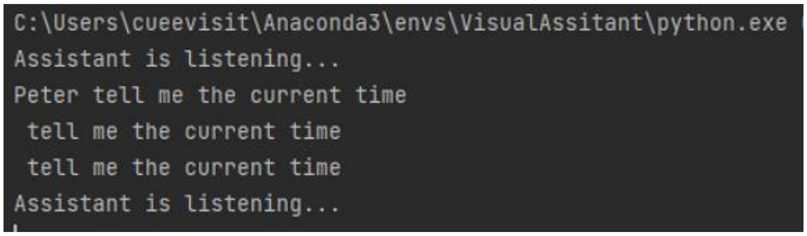
```
import datetime
```



```
import speech_recognition as sr
import pywhatkit
import datetime
```

Try on your own:

Asking: "Assistant Name you set, what is the current time "



```
C:\Users\cueevisit\Anaconda3\envs\VisualAssitant\python.exe
Assistant is listening...
Peter tell me the current time
 tell me the current time
 tell me the current time
Assistant is listening...
```

Your voice assistant will tell you the time by voice

Type some funny code

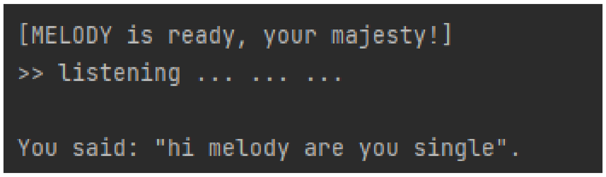
Task 7: Answer the question ‘are you single’?

Paste below to the code:

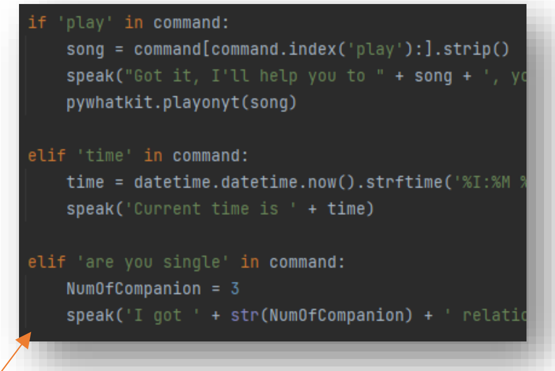
```
        elif 'are you single' in command:
            NumOfCompanion = 3
            speak('I’ve got ' + str(NumOfCompanion) + ' relationship before.')
```

Try on your own:

Asking: “Assistant Name you set, are you single”



```
[MELODY is ready, your majesty!]
>> listening ... ... ...

You said: "hi melody are you single".
```



```
if 'play' in command:
    song = command[command.index('play'):].strip()
    speak("Got it, I'll help you to " + song + ', yo
    pywhatkit.playonyt(song)

elif 'time' in command:
    time = datetime.datetime.now().strftime('%I:%M %
    speak('Current time is ' + time)

elif 'are you single' in command:
    NumOfCompanion = 3
    speak('I got ' + str(NumOfCompanion) + ' relati
```

It’ll tell you “I’ve got 3 relationship before” by voice.

Task 8a: Response to unclear command

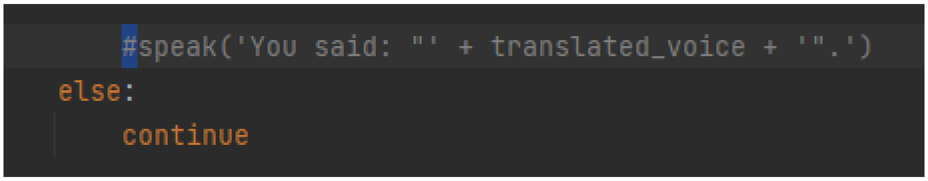
If the virtual assistant heard that you called his/her name, but the command is unclear,

Paste below to the code after:

```
        else:
            speak('I heard you your majesty! But I am unclear about
                                   your instruction. Could you please repeat it')
```

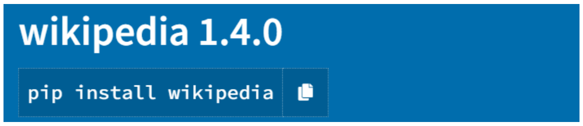
Your assistant will say
“I heard you your majesty!
But I am unclear about your instruction.
Could you please repeat it”
by voice

Task 8b: Comment out annoying speaking what the virtual assistant heard by adding “#”



```
        #speak('You said: "' + translated_voice + '".')
    else:
        continue
```

Task 9 (Optional): Installation of the package of Wikipedia (search Wikipedia function):
https://pypi.org/project/wikipedia/

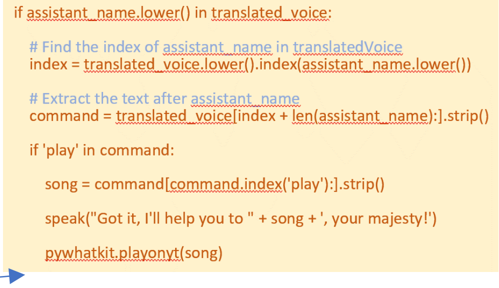


wikipedia 1.4.0

```
pip install wikipedia
```

It is a package that we can use python code to search Wikipedia.
This function will tell you the information from Wikipedia

by asking“who is”



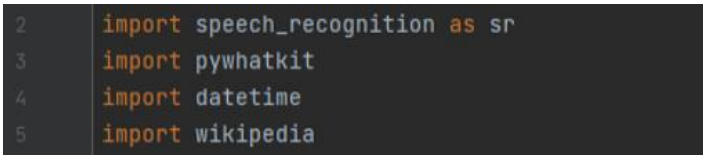
```
if assistant_name.lower() in translated_voice:

    # Find the index of assistant_name in translatedVoice
    index = translated_voice.lower().index(assistant_name.lower())

    # Extract the text after assistant_name
    command = translated_voice[index + len(assistant_name):].strip()

    if 'play' in command:

        song = command[command.index('play'):].strip()

        speak("Got it, I'll help you to " + song + ', your majesty!')

        pywhatkit.playonyt(song)
```

Paste below to the code after

```
  elif 'who is' in command:
        person = command.replace('who is', '')
        info = wikipedia.summary(person, 1)
        print(info)
        speak(info)
```

Type the following code on the top

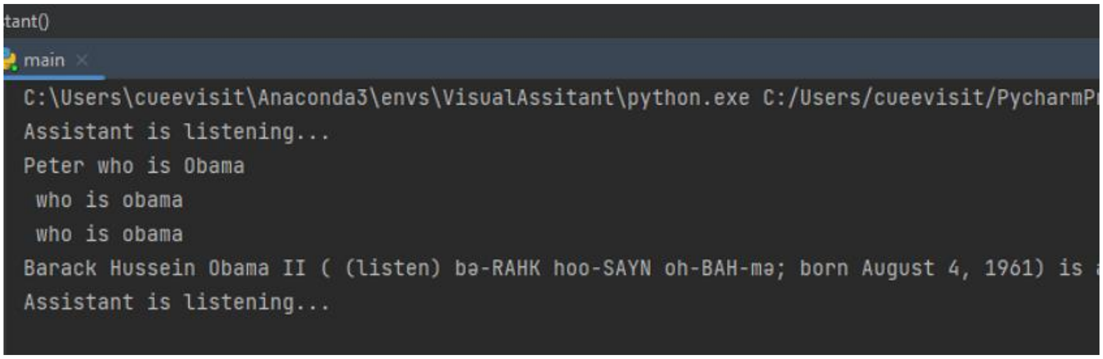
```
import wikipedia
```



```
import speech_recognition as sr
import pywhatkit
import datetime
import wikipedia
```

Try on your own:

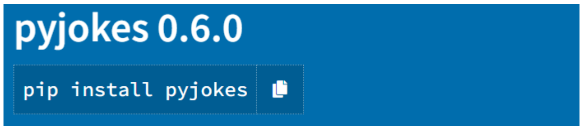
Asking: “Assistant Name you set, who is anymone”



```
C:\Users\cueevisit\Anaconda3\envs\VisualAssitant\python.exe C:/Users/cueevisit/PycharmP
Assistant is listening...
Peter who is Obama
 who is obama
 who is obama
Barack Hussein Obama II ( (listen) bə-RAHK hoo-SAYN oh-BAH-mə; born August 4, 1961) is
Assistant is listening...
```

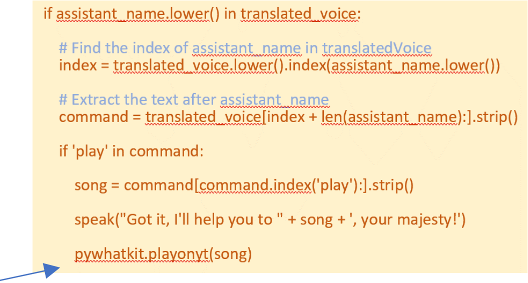
The assistant will tell you the information by voice and text

Task 10 (Optional): Installation of the package of pyJokes (search Jokes function): https://pypi.org/project/pyjokes/



pyjokes 0.6.0

```
pip install pyjokes
```



```
if assistant_name.lower() in translated_voice:

    # Find the index of assistant_name in translatedVoice
    index = translated_voice.lower().index(assistant_name.lower())

    # Extract the text after assistant_name
    command = translated_voice[index + len(assistant_name):].strip()

    if 'play' in command:

        song = command[command.index('play'):].strip()

        speak("Got it, I'll help you to " + song + ', your majesty!')

        pywhatkit.playonyt(song)
```

It is a package that we can use python code to have the program to say some jokes.

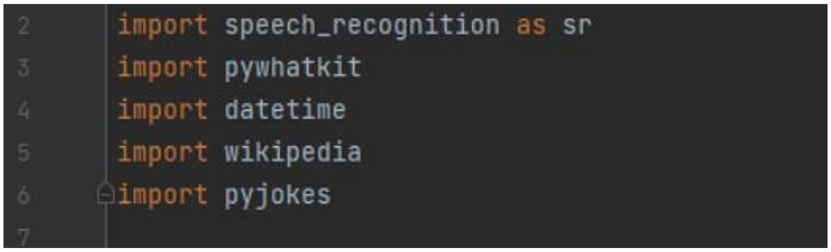
This function will tell you a joke

Paste below to the code after

```
       elif 'joke' in command:
              speak(pyjokes.get_joke())
```

Paste the following code on he top :

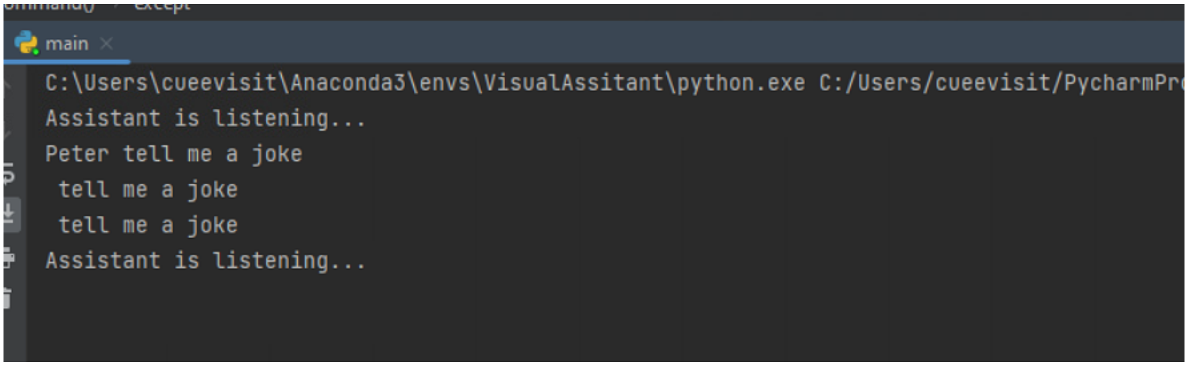
```
import pyjokes
```



```
import speech_recognition as sr
import pywhatkit
import datetime
import wikipedia
import pyjokes
```

Try on your own:

Asking: "Assistant Name you set, tell me a joke"



```
C:\Users\cueevisit\Anaconda3\envs\VisualAssitant\python.exe C:/Users/cueevisit/PycharmPr
Assistant is listening...
Peter tell me a joke
 tell me a joke
 tell me a joke
Assistant is listening...
```

The assistant will tell you the joke by voice

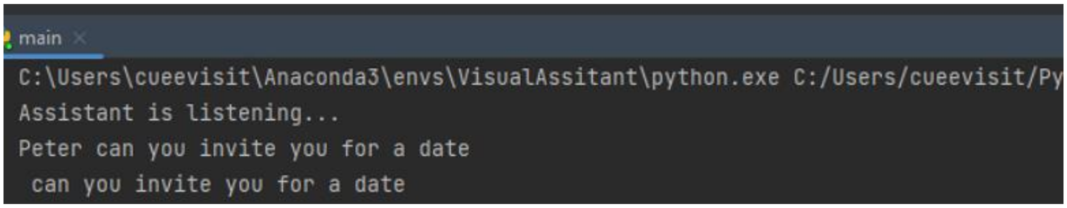
Task 11 (Optional): To refuse your "dating" invitation

Paste below to the code:

```
    elif 'date' in command:
        speak('sorry, I have a headache')
```

Try on your own:

Asking: "Assistant Name you set, can I invite you for a date"



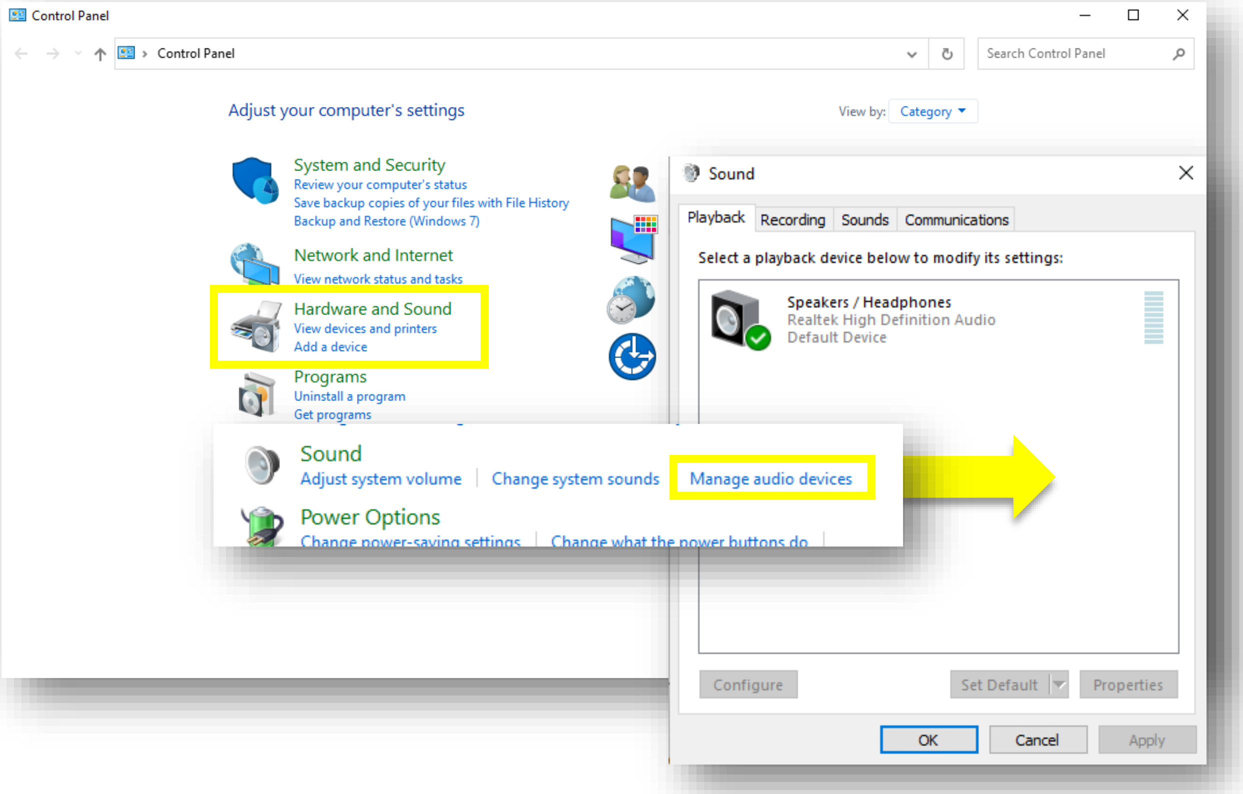
```
C:\Users\cueevisit\Anaconda3\envs\VisualAssitant\python.exe C:/Users/cueevisit/Py
Assistant is listening...
Peter can you invite you for a date
 can you invite you for a date
```

The assistant will tell you: 'sorry, I have a headache'

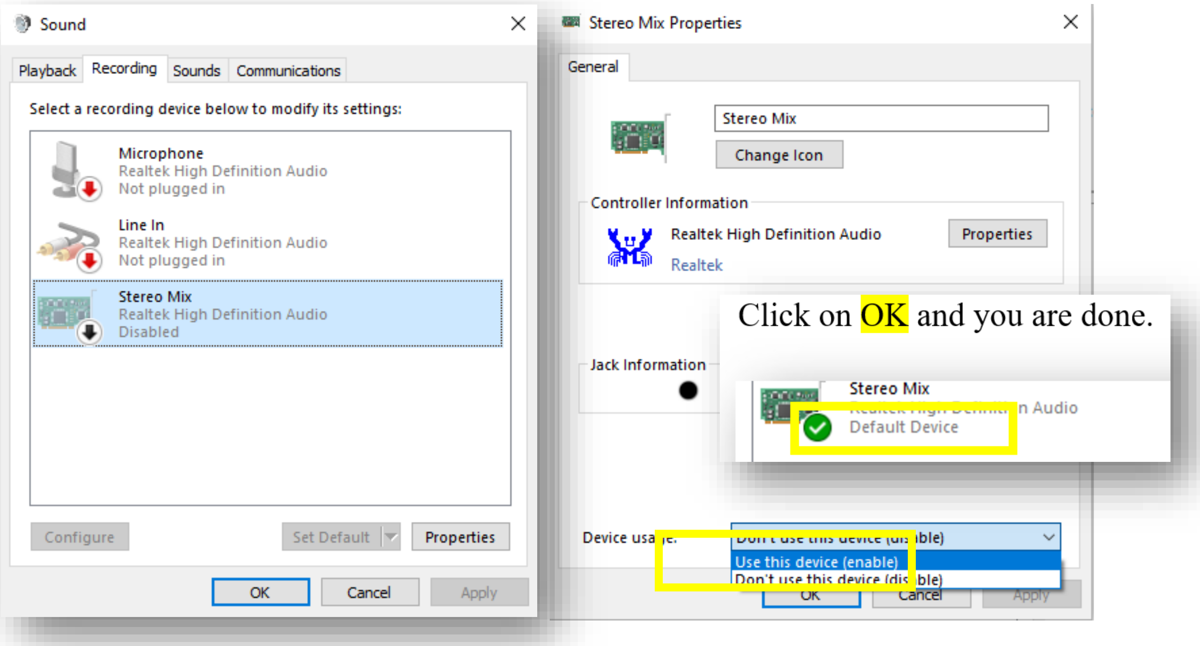
Appendix

How to enable Speech Recognition of Python Code detect system sound in Windows?

1. Open the control panel and click the Hardware and Sound, then Manage audio devices.



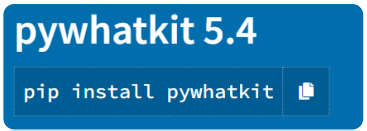
2. Click on Recording and select Stereo Mix. Change to Use this device(enable).



Click on OK and you are done.

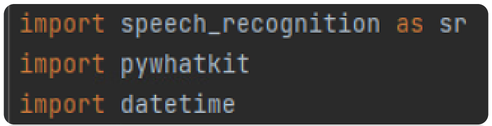
Task 12: Send whatsapp message to specific person immediately

Installation of the package of pywhatkit: https://pypi.org/project/pywhatkit/



pywhatkit 5.4

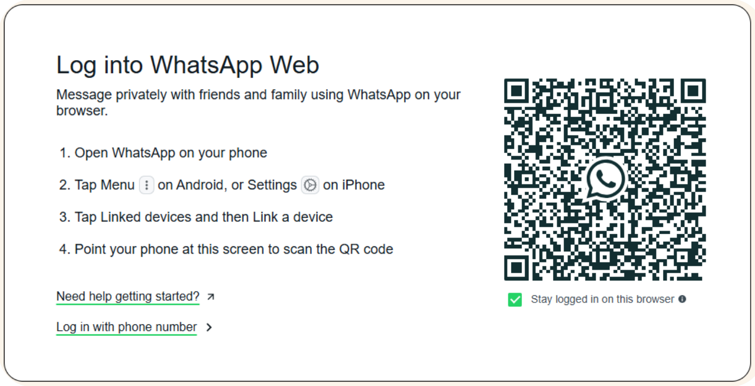
```
pip install pywhatkit
```



```
import speech_recognition as sr
import pywhatkit
import datetime
```

It is a package that we can use python code to send something to others using whatsapp

Open your default browser to visit https://web.whatsapp.com/ and scan QRcode to login



Log into WhatsApp Web

Message privately with friends and family using WhatsApp on your browser.

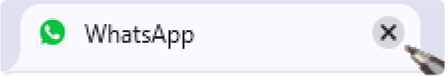
1. Open WhatsApp on your phone
2. Tap Menu ⋮ on Android, or Settings ⚙ on iPhone
3. Tap Linked devices and then Link a device
4. Point your phone at this screen to scan the QR code

Need help getting started? ↗

Log in with phone number ›

☑ Stay logged in on this browser

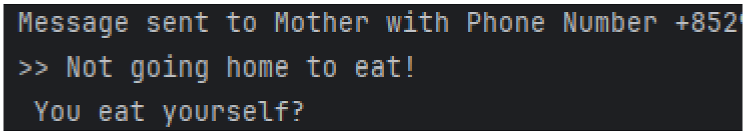
After logged in, you close the tab of this web chatroom interface



```
    elif 'whatsapp my mother' in command:
                Target = 'Mother'
                PhoneNumber = '+852xxxxxxxx'
                Msg = 'Not going home to eat!\nYou eat yourself?'
                pywhatkit.sendwhatmsg_instantly(PhoneNumber, Msg, 10, True, 3)
                print(f"Message sent to {Target} with Phone Number {PhoneNumber}:\n>> {Msg}")
                speak(f"Message sent to {Target} with Phone Number {PhoneNumber}: {Msg}")
```

Try on your own:

Asking: "Assistant Name you set, whatsapp my mother not going home to eat"

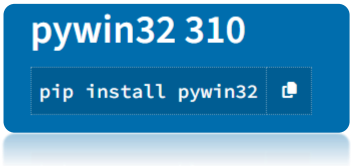


```
Message sent to Mother with Phone Number +852
>> Not going home to eat!
 You eat yourself?
```

The assistant will tell you:

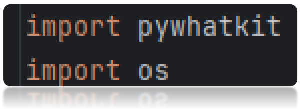
Message sent to Mother with Phone Number +852xxxxxxxx:
Not going home to eat! You eat yourself?'

Task 13: Send whatsapp image with caption to specific person immediately


pywin32 310

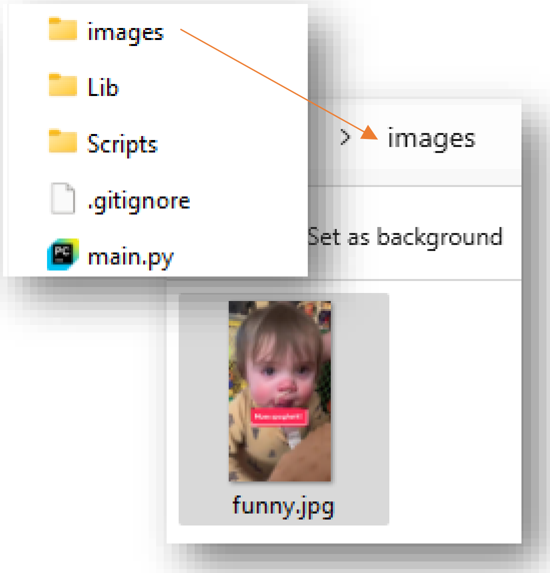
```
pip install pywin32
```

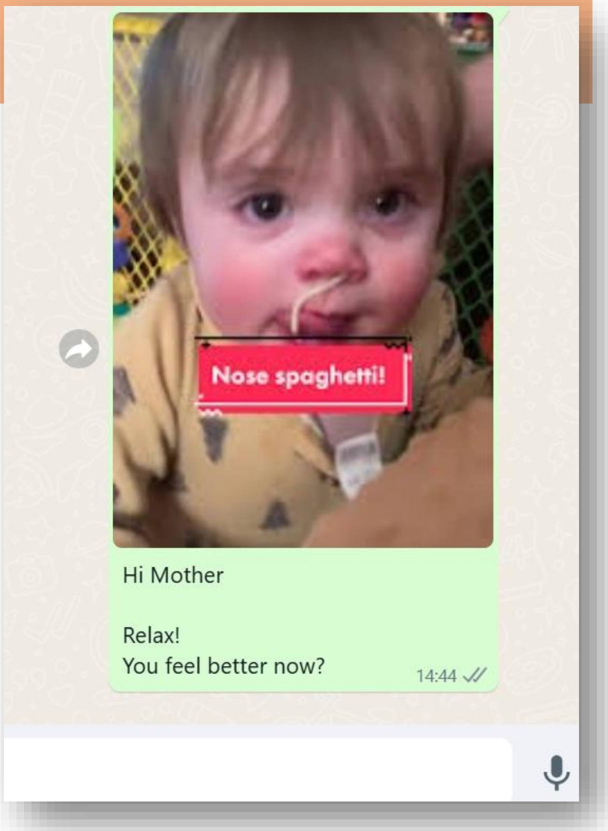
Need to install one more package called "pywin32"



```
import pywhatkit
import os
```

```
elif 'whatsapp image' in command:
    Target = 'Mother'
    PhoneNumber = "+852xxxxxxxx"

    # Specify the relative path to the image
    image_path = "images/funny.jpg"  # Ensure this path is correct
    caption = "Hi " + Target + "\n\n" + "Relax!\nYou feel better now?"

    # Check if the image file exists before sending
    if os.path.isfile(image_path):
        try:
            # Send the image with the caption
            pywhatkit.sendwhats_image(PhoneNumber, image_path, caption, 8, True, 3)
            print(f"Message sent to {Target} with Phone Number {PhoneNumber}:\n>> {caption}")
            speak(f"Message sent to {Target} with Phone Number {PhoneNumber}:\n>> {caption}")
        except Exception as e:
            print(f"An error occurred: {e}")
    else:
        print(f"Image not found: {image_path}")
```





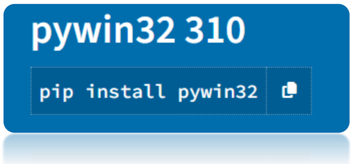
Hi Mother

Relax!
You feel better now? 14:44

Try on your own:

Asking: "Assistant Name you set, whatsapp image"

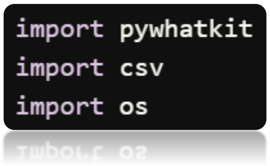
Task 14: Send whatsapp messages to multiple people in batch



pywin32 310

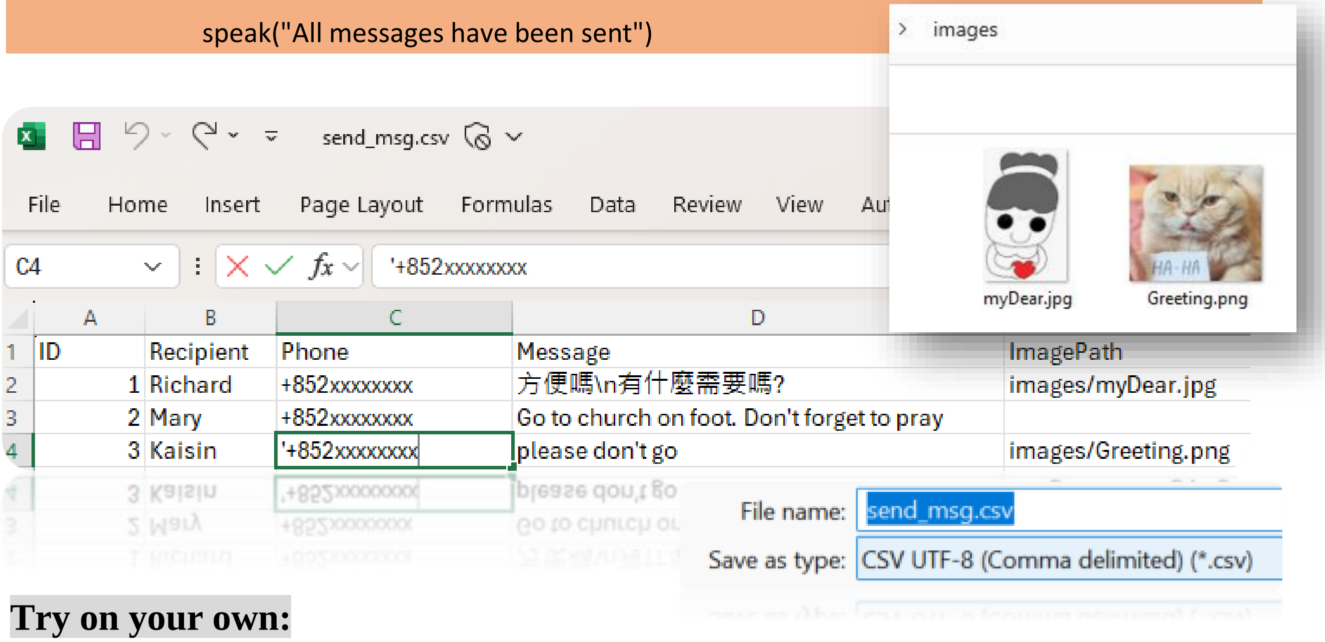
```
pip install pywin32
```

Need to install one more package called "pywin32"



```
import pywhatkit
import csv
import os
```

```
    elif 'whatsapp my schedule' in command:
        with open("send_msg.csv", "r", encoding='utf-8') as file:
            reader = csv.DictReader(file)  # Use tab as delimiter
            for row in reader:
            # Create the full message including the greeting
            message = 'Hi ' + row['Recipient'] + ',\n\n' + row['Message'].replace('\\n', '\n')
            phone = row['Phone']
            image_path = row['ImagePath']

            print(row)  # Print the row for debugging

            # Check if ImagePath is not empty
            if image_path:
                # Ensure the image file exists before sending
                if os.path.isfile(image_path):
                    pywhatkit.sendwhats_image(phone, image_path, message, 8, True, 3)
                else:
                    print(f"Image not found: {image_path}")
            else:
                pywhatkit.sendwhatmsg_instantly(phone, message, 8, True, 3)

            speak("All messages have been sent")
```



C4 '+852xxxxxxxx

	A	B	C	D	E
1	ID	Recipient	Phone	Message	ImagePath
2	1	Richard	+852xxxxxxxx	方便嗎\n有什麼需要嗎?	images/myDear.jpg
3	2	Mary	+852xxxxxxxx	Go to church on foot. Don't forget to pray	
4	3	Kaisin	'+852xxxxxxxx	please don't go	images/Greeting.png

File name: send_msg.csv

Save as type: CSV UTF-8 (Comma delimited) (*.csv)

Try on your own:

Asking: "Assistant Name you set, whatsapp my schedule now"

Practicum Coursework:

Your Own Virtual Assistant Creation (Due: 4:59pm of Today's Session)

- Build your own virtual assistant with specific name that you choose
- The virtual assistant should make use of at least THREE packages.
  (At most choose ONE OPTIONAL package from what I have mentioned in this set of notes)
- You can search https://pypi.org for other TWO packages to try.
- Demonstrate your virtual assistant to TA to get marks
- Upload your source code – Your Student ID.py (e.g. 1155131649.py) to blackboard

(Beware of Plagiarism, a permanent drawback will be issued by HKIE)

hbc-random-number 0.2.0

hkobservatory 1.0.6

fortunetelleracs1029 1.0.0