How To Import or Use Image inside the Tkinter window

pip install pillow

Code:

```
from tkinter import *
from PIL import Image,ImageTk
root= Tk()
root.geometry("200x300")
img = Image.open("imagescenry.jpg")
img = img.resize((200, 300))
photo = ImageTk.PhotoImage(img) #converting the image so that it can be used in tkinter window
label = Label(root, image=photo)
label.place(x=0,y=0)
btn1=Button(root,text="exit",bd=9,command=root.destroy)
btn1.pack()
mainloop()
```

Voice-Assistance Project:

Step 1: Install Required Libraries

Open your terminal (CMD, PowerShell, or terminal in VS Code) and install the following:

pip install pyttsx3

pip install SpeechRecognition

pip install wikipedia

pip install pyautogui

pip install pyjokes

pip install pyaudio

Note: If pyaudio fails to install, run:

pip install pipwin

pipwin install pyaudio

pip install pyttsx3 SpeechRecognition wikipedia pyautogui pyjokes pyaudio

Libraries Explanation:

1. **import pyttsx3**

- **Purpose:** Converts text to speech (TTS).
- **Use in your code:** Makes your assistant speak using the speak() function.
- **Example:**

```
pyttsx3.init().say("Hello")
```

---

2. **import datetime**

- **Purpose:** Access current date and time.
- **Use in your code:** Used in the wishme() function to greet based on time.
- **Example:**

```
datetime.datetime.now().hour  # Gets the current hour
```

---

3. **import speech_recognition as sr**

- **Purpose:** Captures and converts **spoken audio to text** using Google Speech Recognition.
- **Use in your code:** Used in the takecommand() function to understand user input.
- **Example:**

```
sr.Recognizer().listen(source)
```

- **import webbrowser**

- **Purpose:** Opens URLs in the default web browser.
- **Use in your code:** To open Google, YouTube, or specific URLs.
- **Example:**

webbrowser.open("https://www.google.com")

---

4. **import wikipedia**

- **Purpose:** Allows searching and summarizing topics from Wikipedia.
- **Use in your code:** Used to fetch and speak Wikipedia summaries.
- **Example:**

wikipedia.summary("Python programming")

---

5. **import os**

- **Purpose:** Provides access to operating system features like opening files or applications.
- **Use in your code:** To open Notepad or other software.
- **Example:**

os.startfile("C:\\Windows\\notepad.exe")

---

6. **import time**

- **Purpose:** Used for time formatting, delays, or current time.
- **Use in your code:** To speak the current time.
- **Example:**

time.strftime("%H:%M:%S")

---

7. **import pyautogui**

- **Purpose:** Automates keyboard/mouse control.
- **Use in your code:** For volume up/down/mute key presses.
- **Example:**

pyautogui.press('volumeup')

---

8. **import pyjokes**

- **Purpose:** Generates random programming-related jokes.
- **Use in your code:** To tell jokes using speak().
- **Example:**

pyjokes.get_joke()

---

9. **import random**

- **Purpose:** Generates random values.
- **Use in your code:** To simulate a coin toss.
- **Example:**

random.choice(['head', 'tail'])

1. Text-to-Speech Engine Setup

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id)

Explanation:

- pyttsx3.init('sapi5'): Initializes the speech engine. 'sapi5' is used for Windows voice (Microsoft Speech API).
- getProperty('voices'): Gets a list of available voices on your system.
- setProperty('voice', voices[1].id): Sets the assistant's voice. voices[0] is usually male, voices[1] is female.

2. Speak Function

```
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

Explanation:

- say(audio): Queues the text to be spoken.
- runAndWait(): Speaks the text and waits until it's done.

3. Greet Function Based on Time

```
def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        speak("Good morning")
    elif hour >= 12 and hour < 16:
        speak("Good afternoon")
    else:
        speak("Good evening")
```

Explanation:

- Checks the current hour and gives an appropriate greeting.

4. Take Command from Microphone

```
def takecommand():
    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 command is {query}")
    except Exception as e:
        print("say again..")
        return 'none'
    return query
```

- Recognizer(): Creates a recognizer instance.
- Microphone(): Opens the system's default microphone to capture voice.

- pause_threshold = 1: Waits 1 second of silence before considering input complete.
- r.listen(source): Listens to the source and captures audio.
- recognize_google(): Uses Google API to convert voice to text.
- If recognition fails, it returns "none".

5. Main Function Block

if __name__ == '__main__':

This ensures that the script only runs when it's executed directly (not imported as a module).

6. Main Program Execution

wishme()

speak("hello everyone")

speak("Welcome")

What happens here?

- The assistant greets based on time.
- Speaks a welcome message when the program starts.

7. Command Loop

while True:

q = takecommand().lower()

- Starts an infinite loop to keep listening for commands.
- Converts the recognized voice text to lowercase for easier comparison.

Inside the Loop – Handling Commands:

```
if 'open google' in q:
    webbrowser.open("https://www.google.com/") Opens Google.

elif 'open youtube' in q:
    webbrowser.open("https://www.youtube.com/") Opens YouTube.

elif 'play song' in q:
    webbrowser.open("https://youtu.be/BLlTFapgvOo?si=GbdVXNOeDVZtzRdj") Opens a
specific YouTube song.

elif 'wikipedia' in q:
    query = q.replace('wikipedia', '')
    result = wikipedia.summary(query, sentences=4)
    print(result)
    speak(result)
```

Searches Wikipedia for what comes after the word "wikipedia" and speaks a 4-sentence summary.

```
    elif 'open notepad' in q:
        speak("opening notepad")
        p = "C:\\Windows\\notepad.exe" 
```
Opens Notepad.
```
        os.startfile(p)

    elif 'current time' in q:
        speak(time.strftime("%H %M %S %p"))
```
Tells the current time in hours, minutes, seconds, and AM/PM.

Volume Controls with pyautogui:
```
    elif 'volume up' in q:
        pyautogui.press('volumeup')

    elif 'volume down' in q:
        pyautogui.press('volumedown')

    elif 'mute' in q:
        pyautogui.press('volumemute')

    elif 'unmute' in q:
        pyautogui.press('volumeunmute')
```

Telling a Joke

```
elif 'jokes' in q:
    speak(pyjokes.get_joke())
```
Fetches and speaks a random programming joke.

Tossing a Coin

```
elif 'toss a coin' in q:
    a = ['head', 'tail']
    result = random.choice(a)
    print(f"the computer choose {result}")
    speak(f"the computer choose {result}")
```
Simulates a coin toss by randomly choosing "head" or "tail" and announcing it.

Code:

```
import pyttsx3

import datetime

import speech_recognition as sr

import webbrowser

import wikipedia

import os

import time

import pyautogui

import pyjokes
```

```
import random

# Initialize the speech engine
engine = pyttsx3.init('sapi5')  # Use Microsoft Speech API
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)  # Female voice

# Speak function
def speak(audio):
    engine.say(audio)
    engine.runAndWait()

# Wish the user
def wishme():
    hour = int(datetime.datetime.now().hour)
    if 0 <= hour < 12:
        speak("Good morning")
    elif 12 <= hour < 16:
        speak("Good afternoon")
    else:
        speak("Good evening")

# Listen for user commands
def takecommand():
    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 command: {query}")
    except Exception:
        print("Say that again...")
        return "none"
    return query

# Main program
if __name__ == '__main__':
    wishme()
    speak("Hello everyone")
    speak("Welcome")

    while True:
        q = takecommand().lower()

        if 'open google' in q:
            webbrowser.open("https://www.google.com/")

        elif 'open youtube' in q:
            webbrowser.open("https://www.youtube.com/")

        elif 'play song' in q:
            webbrowser.open("https://youtu.be/BLlTFapgvOo?si=GbdVXNOeDVZtzRdj")
```

```
elif 'wikipedia' in q:
    query = q.replace('wikipedia', '')
    result = wikipedia.summary(query, sentences=4)
    print(result)
    speak(result)

elif 'open notepad' in q:
    speak("Opening Notepad")
    os.startfile("C:\\Windows\\notepad.exe")

elif 'current time' in q:
    current_time = time.strftime("%H:%M:%S %p")
    speak(f"The time is {current_time}")

elif 'volume up' in q:
    speak("Increasing volume")
    pyautogui.press('volumeup')

elif 'volume down' in q:
    speak("Decreasing volume")
    pyautogui.press('volumedown')

elif 'mute' in q:
    speak("Muting volume")
    pyautogui.press('volumemute')

elif 'unmute' in q:
    speak("Unmuting volume")
```

```
        pyautogui.press('volumeup')

    elif 'jokes' in q:
        joke = pyjokes.get_joke()
        print(joke)
        speak(joke)

    elif 'toss a coin' in q:
        result = random.choice(['head', 'tail'])
        print(f"The computer chose {result}")
        speak(f"The computer chose {result}")

    elif 'exit' in q or 'quit' in q or 'stop' in q:
        speak("Goodbye! Have a great day.")
        break
```

Github link: https://github.com/Aditya2129939/Python-Training-BFGI