

Project 1: Iron Man Jarvis AI Desktop Voice Assistant | Python Tutorials For Absolute Beginners #120

Have you ever wondered how cool it would be to have your own A.I. assistant? Imagine how easier it would be to send emails without typing a single word, doing Wikipedia searches without opening web browsers, and performing many other daily tasks like playing music with the help of a single voice command. In this tutorial, I will teach you how you can make your personal A.I. assistant using Python.

What can this A.I. assistant do for you?

- It can send emails on your behalf.
- It can play music for you.
- It can do Wikipedia searches for you.
- It is capable of opening websites like Google, Youtube, etc., in a web browser.
- It is capable of opening your code editor or IDE with a single voice command.

Enough talks! Let's start building our own J.A.R.V.I.S.

15:25 - Defining Wish me Function :

Now, we will make a **wishme()** function, that will make our J.A.R.V.I.S. wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program, by:

```
import datetime
```

Now, let's start defining the **wishme()** function:

```
def wishme():  
  
    hour = int(datetime.datetime.now().hour)
```

Here, we have stored the current hour or time integer value into a variable named hour. Now, we will use this hour value inside an if-else loop.

18:27 – Defining Take command Function :

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a **takeCommand()** function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

Before defining the takeCommand() function, we need to install a module called speechRecognition. Install this +

☐ 1. Python Tutorials Teaser

Free YouTube Video

☐ 2. What is Programming and Why Python?

Free YouTube Video

☐ 3. Downloading Python and Pycharm Installation

Free YouTube Video

☐ 4. Using Modules & Pip In Python

Free YouTube Video

☐ 5. Writing Our First Python Program

Free YouTube Video

Calculator

Free YouTube Video

☐ 8. Variables, Datatypes and Typecasting

Free YouTube Video

☐ 9. String Slicing And Other Functions in Python

Free YouTube Video

```
elif 'the time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"Sir, the time is {strTime}")
```

In the above, code with using datetime() function and storing the current or live system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into speech.

[38:45](#) – Defining Task 6: To open the VS Code Program

```
elif 'open code' in query:
    codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
    os.startfile(codePath)
```

To open the VS Code or any other application, we need the code path of the application.

[10:08](#) – Starting VS Code

I am going to use the VS Code IDE in this video. Feel free to use any other IDE you are comfortable d with. Start a new project and make a file called jarvis.py.

[10:54](#) – Defining Speak Function

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our J.A.R.V.I.S. talk, we will make a function called **speak()**. This function will take audio as an argument, and then it will pronounce it.

```
def speak(audio):
    pass    #For now, we will write the conditions later.
```

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak() function we made. We are going to install a module called **pyttsx3**.

What is pyttsx3?

- A python library that will help us to convert text to speech. In short, it is a text-to-speech library.
- It works offline, and it is compatible with Python 2 as well as Python 3.

Installation:

```
pip install pyttsx3
```

In case you receive such errors:

- No module named win32com.client
- No module named win32
- No module named win32api

Then install pywin32 by typing the below command in the terminal :

What is sapi5?

- Microsoft developed speech API.
- Helps in synthesis and recognition of voice.

What Is Voiceld?

- Voice id helps us to select different voices.
- voice[0].id = Male voice
- voice[1].id = Female voice

Writing Our speak() Function :

We made a function called speak() at the starting of this tutorial. Now, we will write our speak() function to convert our text to speech.

```
def speak(audio):  
  
    engine.say(audio)  
  
    engine.runAndWait() #Without this command, speech will not be audible to us.
```

Creating Our main() function:

We will create a main() function, and inside this main() Function, we will call our speak function.

Code:

```
if __name__=="__main__" :  
  
    speak("Code With Harry")
```

Whatever you will write inside this speak() function will be converted into speech. Congratulations! With this, our J.A.R.V.I.S. has its own voice, and it is ready to speak.

Let's start coding the takeCommand() function :

```
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') #Using google for voice recognition.
    print(f"User said: {query}\n") #User query will be printed.

except Exception as e:
    # print(e)
    print("Say that again please...") #Say that again will be printed in case of improper voice
    return "None" #None string will be returned
return query

```

[27:30](#) – Coding logic of Jarvis

Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc.

[28:04](#) – Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module :

```
pip install wikipedia
```

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

```

if __name__ == "__main__":
    wishMe()
    while True:
        # if 1:
        query = takeCommand().lower() #Converting user query into lower case

        # Logic for executing tasks based on query
        if 'wikipedia' in query: #if wikipedia found in the query then this block will be executed
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

```

In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

[31:24](#) – Defining Task 2: To open YouTube site in a web-browser

To open any website, we need to import a module called webbrowser. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statement.

[33:37](#) – Defining Task 4: To play music

To play music, we need to import a module called `os`. Import this module directly with an import statement.

```
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]))
```

In the above code, we first opened our music directory and then listed all the songs present in the directory with the `os` module's help. With the help of `os.startfile`, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, J.A.R.V.I.S. will play any random song from the song directory.

Steps to get the code path of the application:

Step 1: Open the file location.

Step 2: Right-click on the application and click on properties.

Step 3: Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, I am saving the target into a variable called `codePath`, and then we are using the `os` module to open the application.

[41:05](#) – Defining Task 7: To send Email

To send an email, we need to import a module called `smtplib`.

What is `smtplib`?

- Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and to route emails between mail servers. An instance method called **`sendmail`** is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:
- **The sender:** Email address of the sender.
- **The receiver:** T Email of the receiver.
- **The message:** A string message which needs to be sent to one or more than one recipient.

[44:03](#) – Defining Send email function :

We will create a **`sendEmail()`** function, which will help us send emails to one or more than one recipient.

```
def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('youreemail@gmail.com', 'your-password')
```

```

elif 'email to harry' 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")

```

We are using the try and except block to handle any possible error while sending emails.

51:26 – Recapitulate

1. First of all, we have created a **wishme()** function that gives the greeting functionality according to our A.I system time.
2. After wishme() function, we have created a **takeCommand()** function, which helps our A.I to take command from the user. This function is also responsible for returning the user's query in a string format.
3. We developed the code logic for opening different websites like google, youtube, and stack overflow.
4. Developed code logic for opening VS Code or any other application.
5. At last, we added functionality to send emails.

56:13 – Is it an A.I.?

Many people will argue that the virtual assistant that we have created is not an A.I, but it is the output of a bunch of the statement. But, if we look at the fundamental level, the sole purpose of A.I develop machines that can perform human tasks with the same effectiveness or even more effectively than humans.

It is a fact that our virtual assistant is not a very good example of A.I., but it is an A.I. !

58:30 – The E.N.D.

With this, you have successfully made your very first virtual assistant. Explore and try to add other functionalities to J.A.R.V.I.S. I hope you all have liked this tutorial. Feel free to ask your queries in the QnA section.

Code as described/written in the video

```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[0].id)

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

def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")

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

[< Previous](#)[Next >](#)