## MINI PROJECT

**(2022-23)**

## DESKTOP ASSISTANT USING PYTHON

**Mid-Term Progress Report**



## GLA Institute of Engineering & Technology

## Aman (201500071)

**Ashish Sharma (201500166)**

**Supervised By**

# Mr. Ankit Arora

**(Asst. Professor)**

## Department of Computer Engineering & Applications

#### Department of computer Engineering and Applications GLA University, Mathura

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan, Mathura – 281406**

**Declaration**

I hereby declare that the work which is being presented in the MINI PROJECT “DESKTOP ASSISTANT”,in partial fulfillment of the requirements for MINI PROJECT viva voce, is an authentic record of my own work carried under the supervision of Prof. Mr. Ankit Arora

#### .

Signature of Candidate:

Name of Candidate: Aman, Ashish Sharma

Roll. No. : 201500071,201500166

Course: B. Tech (CSE) Year: 3rd

Semester: 6th

**Abstract**

As we know Python is a suitable language for script writers and developers. Let’s write a script for Voice Assistant using Python. The query for the assistant can be manipulated as per the user’s need.

Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called **Speech Recognition** to allow us to convert audio into text for further processing. In this article, we will look at converting large or long audio files into text using the **Speech Recognition API** in python.

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

* It can send emails for you.
* 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.

# Introduction

## Objective

MAIN OBJECTIVE IS TO MAKE A VIRTUAL DESKTOP ASSISTANT WHICH CAN EASE HUMAN EFFORTS-:

## SOME FEATURE INCLUDE-:

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.

ALL THIS WILL BE DONE USING PYTHON LANGUAGE…

## SOFTWARE AND HARDWARE REQUIRMENTS

**Hardware Requirements**

* + Personal computer with internet connection
  + i3 Processor Based Computer or Higher
  + Memory: 2 GB RAM(Maximum)
  + Hard Drive: 1 GB(Maximum)

## Software Requirements

* + Windows 7 Or Higher
  + Google Chrome Version 40.0.2214 Or Higher
  + Python compiler,SAPI5
  + Visual Studio code/pycharm/anaconda

## Frontend Tools

* + python

# Implementation

## MODULES USED

* **Subprocess:-** This module is used for getting system subprocess details which are used in various commands i.e Shutdown, Sleep, etc. This module comes buit-in with Python.
* **Pyttsx3:-** This module is used for conversion of text to speech in a program it works offline. To install this module type the below command in the terminal.

#### pip install pyttsx3.

* **Wikipedia:-** As we all know Wikipedia is a great source of knowledge just like Geeks for Geeks we have used Wikipedia module to get information from Wikipedia or to perform Wikipedia search. To install this module type the below command in the terminal.
* **Speech Recognition:-** Since we’re building an Application of voice assistant, one of the most important things in this is that your assistant recognizes your voice (means what you want to say/ ask). To install this module type the below command in the terminal.
* **Web browser:-** To perform Web Search. This module comes buit-in with Python.
* **Date-time:-** Date and Time is used to showing Date and Time. This module comes built-int with Python.
* **Requests:** Requests is used for making GET and POST requests. To install this module type the below command in the terminal.

**pip install requests**

## A BRIEF ON PYTHON LANGUAGE:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

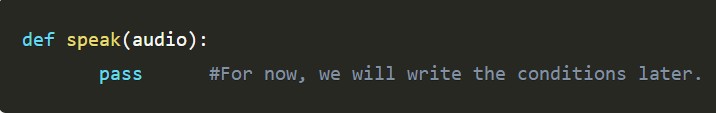
### WHAT IS A DESKTOP ASSISTANT:

Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices.

There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as "what is the weather today," or "play pop music." Those requests are processed and stored in Amazon's cloud.

Detailed Description of code:

## Defining Speak Function

The and 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.

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 which 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 the Python 3

### Installation:

pip install pyttsx3

In case you receive such errors:

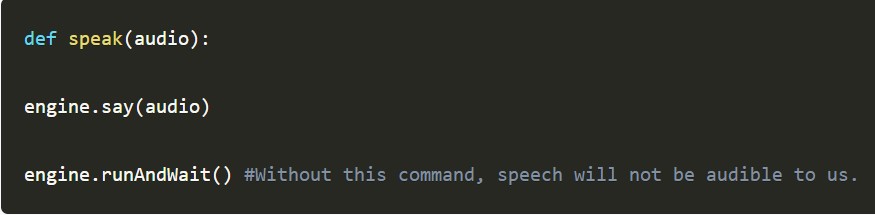
No module named win32com.client No module named win32

No module named win32api

Then, install pypiwin32 by typing the below command in the terminal : pip install pypiwin32.

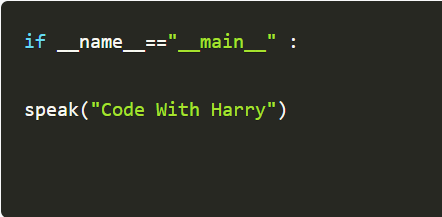
### Writing Our speak() Function :

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



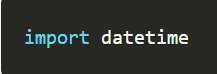
### Creating Our main() function:

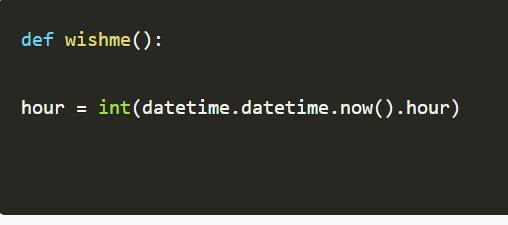
Now, we will create a main() function, and inside this main() Function, we will call our speak function.



### Defining Wish me Function :

Now, we are going to 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:



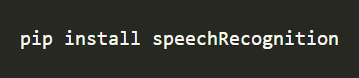


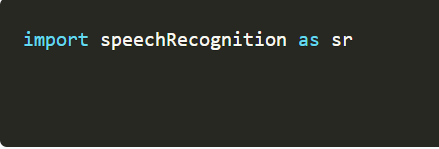
Here, we have stored the integer value of the current hour or time into a variable named hour. Now, we will use this hour value inside an if-else loop.

### Defining Take command Function :

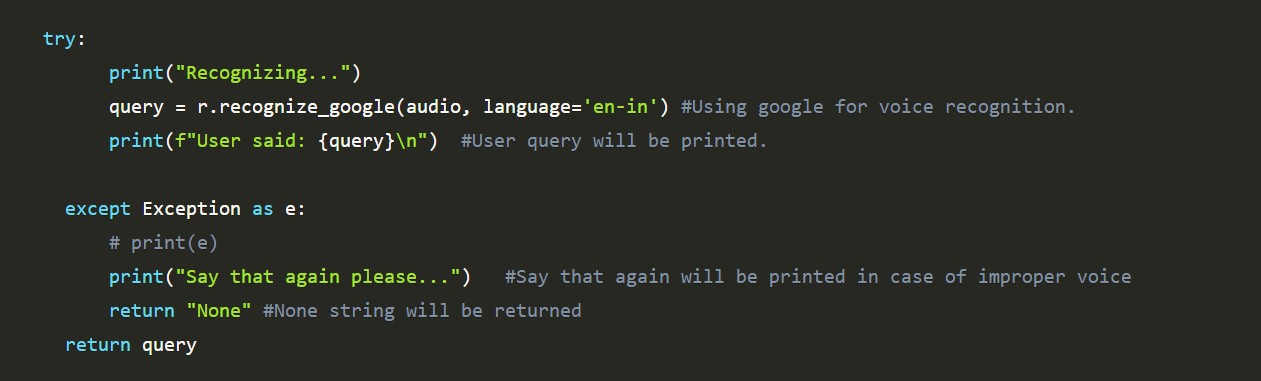
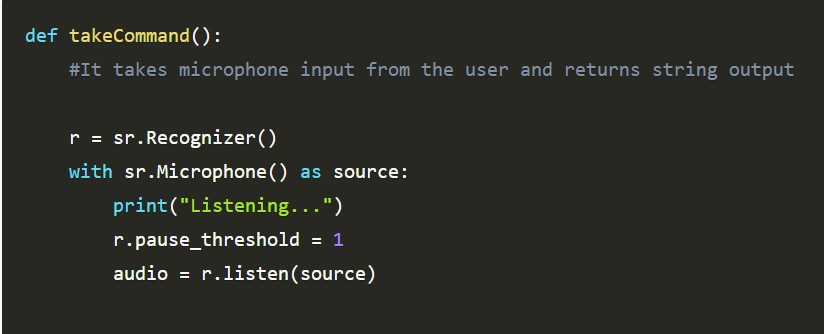
The next most important thing for our A.I. assistant is that it should be able to take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will be able to return a string output by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

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



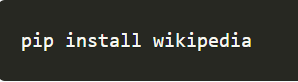


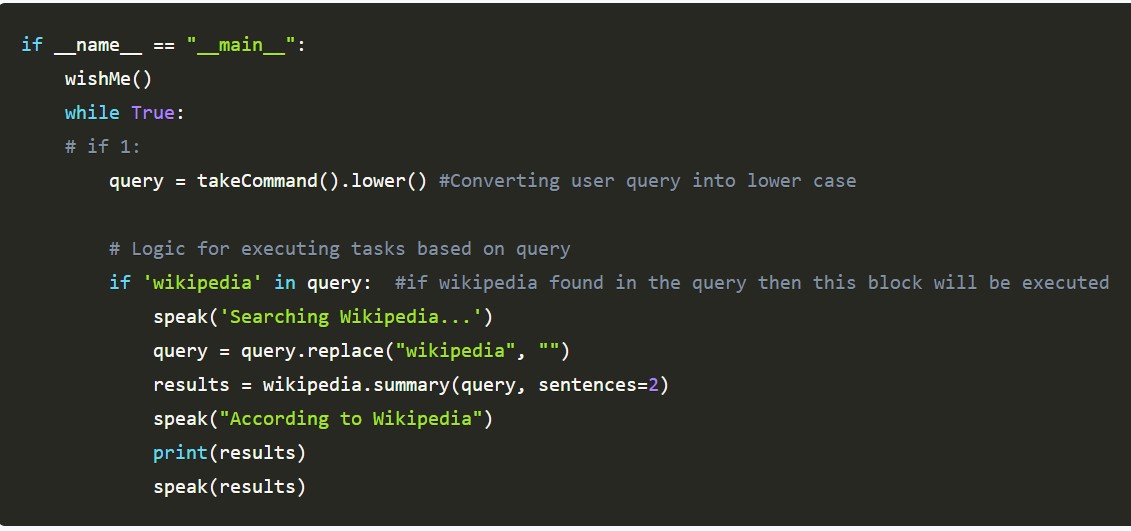
### Let's start coding the takeCommand() function :



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 :

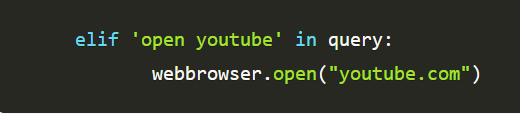




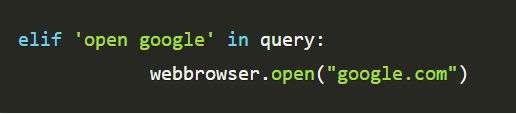
In the above code, we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the help of speak function.

### 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 pip statement, we can directly import it into our program by writing an import statement.

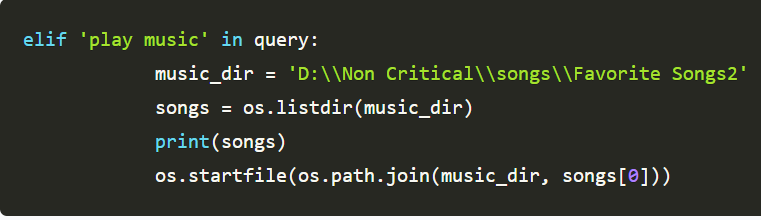


### Defining Task 3: To open Google site in a web-browser



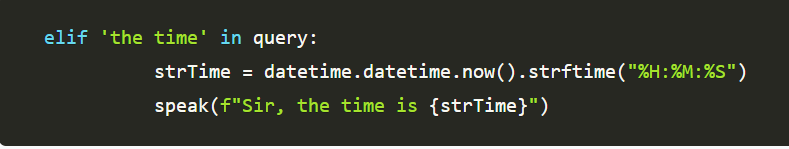
Defining Task 4: To play music

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



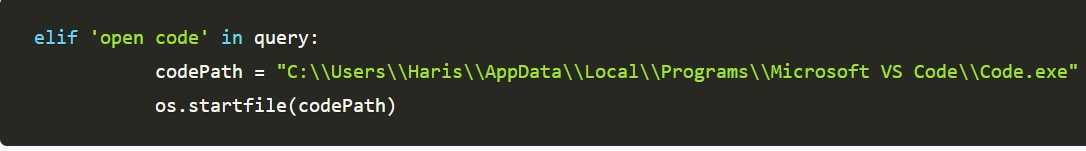
In the above code, we first opened our music directory and then listed all the songs present in the directory with the help of the os module. With the help of os.starfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with 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.

### Defining Task 5: To know the current time



In the above, code with are using datetime() function and storing the current or live of the 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 the speech.

### Defining Task 6: To open the VS Code Program



To open the VS Code or any other application, we need the code path of the application. 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 code Path, and then we are using the os module to open the application.

### Defining Task 7: To send Email

To send an email, we need to import a module called smtplib. What is smtplib?

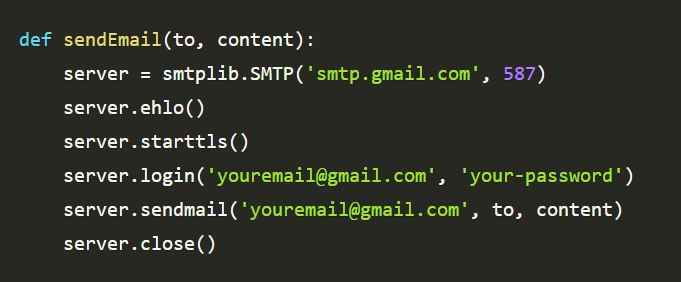
Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and to route emails between mail servers. An instance method called sendmail is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:

The sender: Email address of the sender. The receiver:T Email of the receiver.

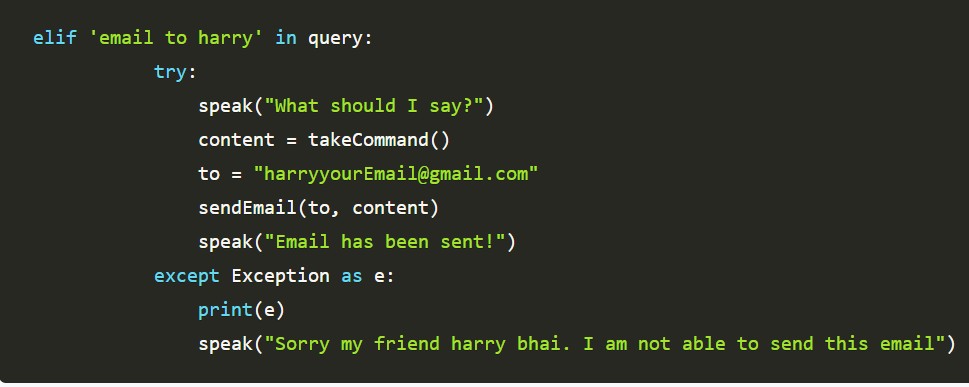
The message: A string message which needs to be sent to one or more than one recipient.

### Defining Send email function :

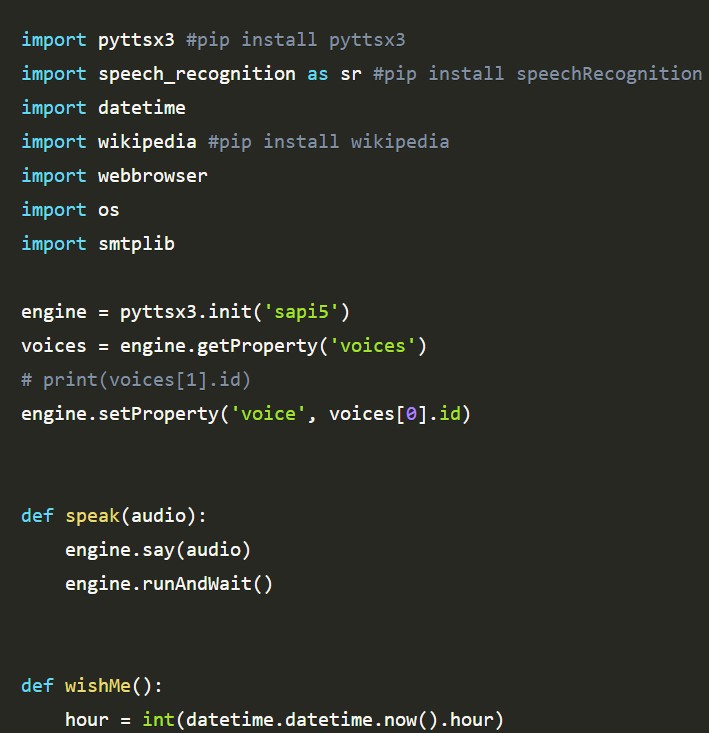
Now, we will create a sendEmail() function, which will help us to send emails to one or more than one recipients.

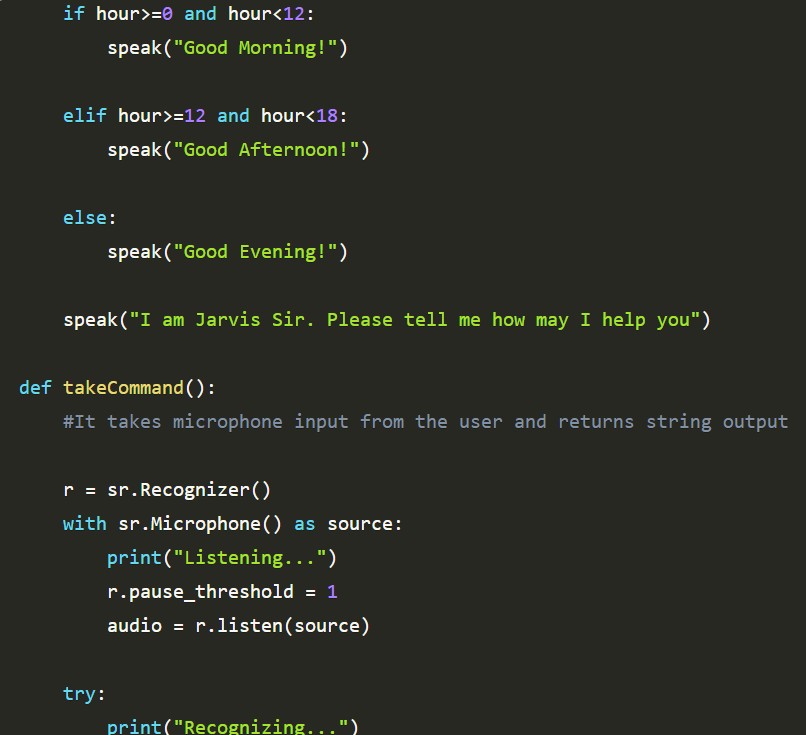


### Calling sendEmail() function inside the main() function:



WHOLE CODE:



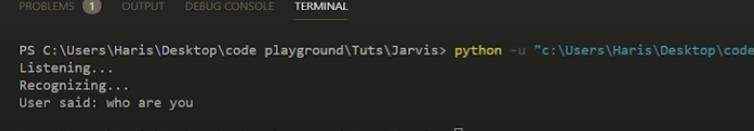








### SAMPLE OUTPUTS:





SUMMARY:

First of all, we have created a wishme() function that gives the functionality of greeting according to the system time to our A.I.

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.

We developed the code logic for opening different websites like google, youtube, and stack overflow.

Developed code logic for opening VS Code or any other application. At last, we added functionality to send emails.

With this, you have successfully made your very first virtual assistant.

## BIBLIOGRAPHY:

1. [**https://www.udemy.com/course/learn-python-from-scratch-**](https://www.udemy.com/course/learn-python-from-scratch-basic-to-advance/learn/lecture/21289804#overview)[**basic-to-advance/learn/lecture/21289804#overview**](https://www.udemy.com/course/learn-python-from-scratch-basic-to-advance/learn/lecture/21289804#overview)

## [www.wikipedia.com](http://www.wikipedia.com/)

1. [**https://medium.com/voice-tech-podcast/desktop-assistant-in-**](https://medium.com/voice-tech-podcast/desktop-assistant-in-python-3-6-e11ab7739f70)[**python-3-6-e11ab7739f70**](https://medium.com/voice-tech-podcast/desktop-assistant-in-python-3-6-e11ab7739f70)
2. [**https://en.wikipedia.org/wiki/Python\_(programming\_language)**](https://en.wikipedia.org/wiki/Python_(programming_language))