**PYTHON TUTOR USING VOICE ASSISTANT**

**Problem Statement**

This project will help the students to learn python very easily because in This program student will communicate with the program and This program will give suitable result to students in speech for the students our program is like a game in that way they learn things very faster and very easily.

**Key Features/Benefits**

* VOICE SEARCH
* Easy to use
* Customizable
* Fast loading (speed)

**List of Software Used**

* *PYTHON*
* *PYAUDIO LIBRARY*
* *PYTTSX3 LIBRARY*
* *SPEECH RECOGNISITION LIBRARY*

**Deliverables**

* *Video*
* *ppt*
* *code worksheet*
* *https://cumailin-my.sharepoint.com/personal/bb\_ece13\_cuchd\_in/\_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9jdW1haWxpbi1teS5zaGFyZXBvaW50LmNvbS86ZjovZy9wZXJzb25hbC9iYl9lY2UxM19jdWNoZF9pbi9FcGQ1d1NOVHRNaEJ0T1JSWXQ2WGtsRUJNVU5uS0RPMV9ucjMyTzhsY055ZG53P3J0aW1lPU5XSHRXUW16MkVn&id=%2Fpersonal%2Fbb%5Fece13%5Fcuchd%5Fin%2FDocuments%2FProject%20Submissions%2FBD1A%2FGroup1*

**(*Kindly provide the link where all deliverables are uploaded*)**

**Workflow**

**Step – 1: *Brief title of Step-1***

importing pyttsx3

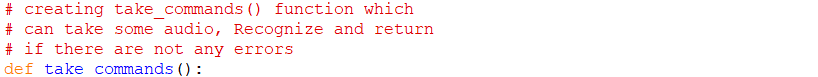
importing speech recognition



**Step – 2: *Brief title of Step-2***

creating take commands () function which can take some audio, Recognize and return

#if there are not any errors

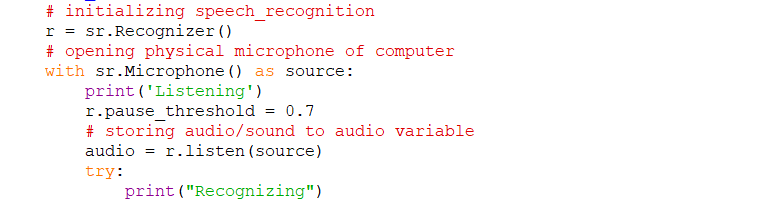


**Step – 3: *Brief title of Step-3***

# initializing speech recognition

# opening physical microphone of computer

# storing audio/sound to audio variable

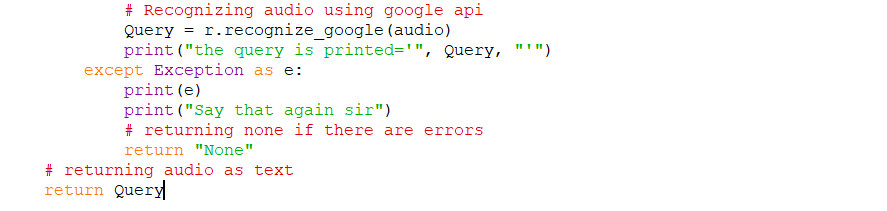


**Step – 4: *Brief title of Step-3***

# Recognizing audio using google api

# returning none if there are errors

# returning audio as text



**Step – 5: *Brief title of Step-3***

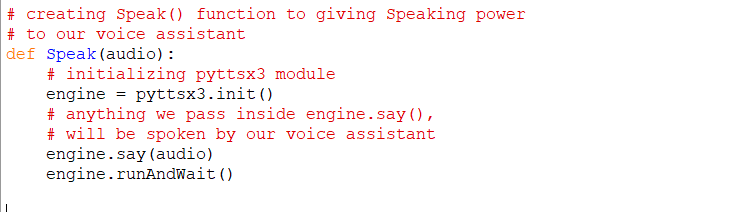
# creating Speak() function to giving Speaking power

# to our voice assistant

# initializing pyttsx3 module

# anything we pass inside engine.say

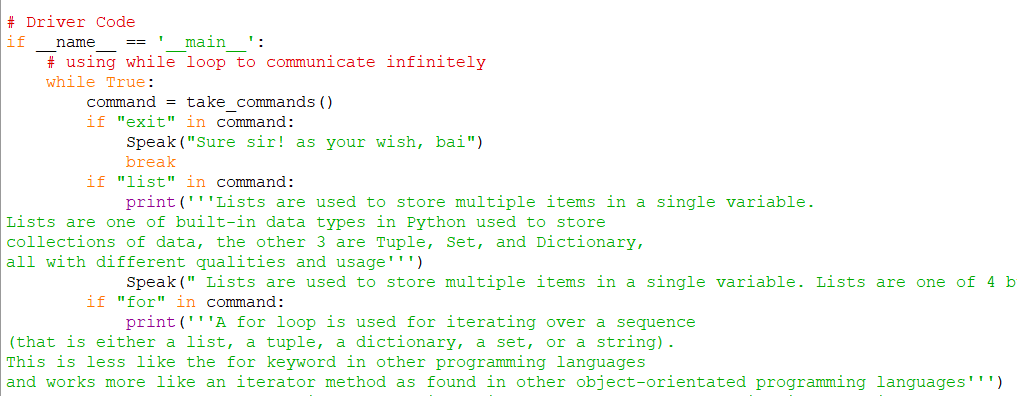
# will be spoken by our voice assistant



**Step – 6: *Brief title of Step-3***

# Driver Code

# using while loop to communicate infinitely



**We use many if commands in while loop to communicate infinitely**

**Code**

***print('''***

***\_ \_ \_ \_***

***| | | | | | | |***

***\_ \_\_ \_ \_| |\_| |\_\_ \_\_\_ \_ \_\_ \_\_ \_\_\_\_\_ \_ \_\_| | \_\_| |***

***| '\_ \| | | | \_\_| '\_ \ / \_ \| '\_ \ \ \ /\ / / \_ \| '\_\_| |/ \_` |***

***| |\_) | |\_| | |\_| | | | (\_) | | | | \ V V / (\_) | | | | (\_| |***

***| .\_\_/ \\_\_, |\\_\_|\_| |\_|\\_\_\_/|\_| |\_| \\_/\\_/ \\_\_\_/|\_| |\_|\\_\_,\_|***

***| | \_\_/ |***

***|\_| |\_\_\_/***

***''')***

***print(" What you Want to learn in python")***

***print('''1. Introduction***

***2.Operatores***

***3.Arithmetic operators***

***4.Relational Operators***

***5.Logical operators***

***6.Bitwise operators***

***7.Assignment operators***

***8.loop***

***9.While Loop***

***10.for Loop***

***11.Nested Loops***

***12.Continue Statement***

***13.Break Statement***

***14.list***

***15.functions***

***16.module***

***17.comments***

***18.Pip***

***19.Exit from the project***

***''')***

***print("Just Speak the name of the topic you get the best result related to the topic")***

***# importing pyttsx3***

***import pyttsx3***

***# importing speech\_recognition***

***import speech\_recognition as sr***

***# creating take\_commands() function which***

***# can take some audio, Recognize and return***

***# if there are not any errors***

***def take\_commands():***

***# initializing speech\_recognition***

***r = sr.Recognizer()***

***# opening physical microphone of computer***

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

***print('Listening')***

***r.pause\_threshold = 0.7***

***# storing audio/sound to audio variable***

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

***try:***

***print("Recognizing")***

***# Recognizing audio using google api***

***Query = r.recognize\_google(audio)***

***print("the query is printed='", Query, "'")***

***except Exception as e:***

***print(e)***

***print("Say that again sir")***

***# returning none if there are errors***

***return "None"***

***# returning audio as text***

***return Query***

***# creating Speak() function to giving Speaking power***

***# to our voice assistant***

***def Speak(audio):***

***# initializing pyttsx3 module***

***engine = pyttsx3.init()***

***# anything we pass inside engine.say(),***

***# will be spoken by our voice assistant***

***engine.say(audio)***

***engine.runAndWait()***

***# Driver Code***

***if \_\_name\_\_ == '\_\_main\_\_':***

***# using while loop to communicate infinitely***

***while True:***

***command = take\_commands()***

***if "exit" in command:***

***Speak("Sure sir! as your wish, bai")***

***break***

***if "list" in command:***

***print('''Lists are used to store multiple items in a single variable.***

***Lists are one of built-in data types in Python used to store***

***collections of data, the other 3 are Tuple, Set, and Dictionary,***

***all with different qualities and usage''')***

***Speak(" Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage")***

***if "for" in command:***

***print('''A for loop is used for iterating over a sequence***

***(that is either a list, a tuple, a dictionary, a set, or a string).***

***This is less like the for keyword in other programming languages***

***and works more like an iterator method as found in other object-orientated programming languages''')***

***Speak("A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages")***

***if "functions" in command:***

***print('''A function is a block of code which only runs when it is called.***

***You can pass data, known as parameters, into a function.***

***A function can return data as a result.''')***

***Speak("A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result")***

***if "module" in command:***

***print('''Modules in Python are simply Python files with a .py extension.***

***The name of the module will be the name of the file.***

***A Python module can have a set of functions,***

***classes or variables defined and implemented.''')***

***Speak("modules in Python are simply Python files with a .py extension. The name of the module will be the name of the file. A Python module can have a set of functions, classes or variables defined and implemented")***

***if "logical" in command:***

***Speak("Logical operators perform Logical AND, Logical OR and Logical NOT operations.")***

***print('''***

***Operator Description Syntax***

***and Logical AND: True if both the operands are true x and y***

***or Logical OR: True if either of the operands is true x or y***

***not Logical NOT: True if operand is false not x***

***''')***

***if "operators" in command:***

***Speak("Operators are used to perform operations on variables and values.")***

***print('''Operators are used to perform operations on variables and values.''')***

***if "introduction" in command:***

***Speak("Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.Python is a programming language that lets you work quickly and integrate systems more efficiently.There are two major Python versions: Python 2 and Python 3. Both are quite different")***

***print('''Python is a widely used general-purpose, high level programming language.***

***It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation.***

***It was designed with an emphasis on code readability,***

***and its syntax allows programmers to express their concepts in fewer lines of code.***

***Python is a programming language that lets you work quickly and integrate systems more efficiently.***

***There are two major Python versions: Python 2 and Python 3. Both are quite different''')***

***if "comments" in command:***

***Speak("Comments can be used to explain Python code.Comments can be used to make the code more readable.Comments can be used to make the code more readable.")***

***print('''Comments can be used to explain Python code.***

***Comments can be used to make the code more readable.***

***Comments can be used to prevent execution when testing code.''')***

***if "arithmetic" in command:***

***Speak("Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication and division")***

***print('''Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication and division.***

***Operator Description Syntax***

***+ Addition: adds two operands x + y***

***- Subtraction: subtracts two operands x - y***

***\* Multiplication: multiplies two operands x \* y***

***/ Division (float): divides the first operand by the second x / y***

***// Division (floor): divides the first operand by the second x // y***

***% Modulus: returns the remainder when first operand is divided by the second x % y***

***\*\* Power : Returns first raised to power second x \*\* y***

***''')***

***if "for" in command:***

***Speak("A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages")***

***print('''A for loop is used for iterating over a sequence***

***(that is either a list, a tuple, a dictionary, a set, or a string).***

***This is less like the for keyword in other programming languages,***

***and works more like an iterator method as found in other object-orientated programming languages***

***Syntax:***

***for iterator\_var in sequence:***

***statements(s)''')***

***if "relational" in command:***

***Speak("Relational operators compares the values. It either returns True or False according to the condition")***

***printT('''Relational operators compares the values. It either returns True or False according to the condition***

***Operator Description Syntax***

***> Greater than: True if left operand is greater than the right x > y***

***< Less than: True if left operand is less than the right x < y***

***== Equal to: True if both operands are equal x == y***

***!= Not equal to - True if operands are not equal x != y***

***>= Greater than or equal to: True if left operand is greater than or equal to the right x >= y***

***<= Less than or equal to: True if left operand is less than or equal to the right x <= y***

***''')***

***if "pip" in command:***

***Speak("PIP is a package manager for Python packages, or modules if you like.")***

***print('''PIP is a package manager for Python packages, or modules if you like.''')***

***if "bitwise" in command:***

***Speak("Bitwise operators acts on bits and performs bit by bit operation.")***

***print('''Bitwise operators acts on bits and performs bit by bit operation.***

***Operator Description Syntax***

***& Bitwise AND x & y***

***| Bitwise OR x | y***

***~ Bitwise NOT ~x***

***^ Bitwise XOR x ^ y***

***>> Bitwise right shift x>>***

***<< Bitwise left shift x<<''')***

***if "assignment" in command:***

***Speak("Assignment operators are used to assign values to the variables.")***

***print('''Assignment operators are used to assign values to the variables.***

***Operator Description Syntax***

***= Assign value of right side of expression to left side operand x = y + z***

***+= Add AND: Add right side operand with left side operand and then assign to left operand a+=b a=a+b***

***-= Subtract AND: Subtract right operand from left operand and then assign to left operand a-=b a=a-b***

***\*= Multiply AND: Multiply right operand with left operand and then assign to left operand a\*=b a=a\*b***

***/= Divide AND: Divide left operand with right operand and then assign to left operand a/=b a=a/b***

***%= Modulus AND: Takes modulus using left and right operands and assign result to left operand a%=b a=a%b***

***//= Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand a//=b a=a//b***

***\*\*= Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand a\*\*=b a=a\*\*b***

***&= Performs Bitwise AND on operands and assign value to left operand a&=b a=a&b***

***|= Performs Bitwise OR on operands and assign value to left operand a|=b a=a|b***

***^= Performs Bitwise xOR on operands and assign value to left operand a^=b a=a^b***

***>>= Performs Bitwise right shift on operands and assign value to left operand a>>=b a=a>>b***

***<<= Performs Bitwise left shift on operands and assign value to left operand a <<= b a= a << b***

***''')***

***if "while" in command:***

***Speak("In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied. And when the condition becomes false, the line immediately after the loop in program is executed")***

***print('''In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied.***

***And when the condition becomes false, the line immediately after the loop in program is executed***

***Syntax :***

***while expression:***

***statement(s)***

***''')***

***if "nested" in command:***

***Speak("Python programming language allows to use one loop inside another loop. Following section shows few examples to illustrate the concept. ")***

***print('''Python programming language allows to use one loop inside another loop. Following section shows few examples to illustrate the concept.***

***syntex:***

***for iterator\_var in sequence:***

***for iterator\_var in sequence:***

***statements(s)***

***statements(s)''')***

***if "continue" in command:***

***Speak("It returns the control to the beginning of the loop")***

***print('''It returns the control to the beginning of the loop***

***syntex:***

***for letter in 'geeksforgeeks':***

***if letter == 'e' or letter == 's':***

***continue***

***print 'Current Letter :', letter***

***var = 10''')***

***if "break" in command:***

***Speak("it brings control out of the loop")***

***print(''' it brings control out of the loop***

***syntex:***

***for letter in 'geeksforgeeks':***

***# break the loop as soon it sees 'e'***

***# or 's'***

***if letter == 'e' or letter == 's':***

***break***

***print 'Current Letter :', letter''')***