Part I Describe an intriguing learner python exercise

Congratulations on deciding to explore programming! The world desperately needs programmers with your skill sets. Programming is about realizing the art of the possible through voice, image, sounds, colors, mathematics, and science. Computers use Python software to build challenging monster games and take pictures of Mars on NASA's robotic helicopter Ingenuity. Software is amazing and provides a means to transform our world into phantasmagoria.

This exercise will translate something you say vocally into text and speak it back to you. Start by importing software libraries with code containers, called objects, that act and perform work. Objects have embedded instructions, called methods, to perform specific tasks. In this case, methods instruct your computer's microphone to record your voice and translate it into text. What is the best part? Your computer then speaks the decoded text back to you.

Speaking into the computer's microphone hardware is possible because the pyaudio library has a listening object that generates voice data. A Google speech recognition library has a method to translate voice data into printed text. Finally, another method creates an audio file and plays the translated text back to you. Let's get started!

<div align="right"># words: 190/200 max</div>

```python
1  import os
2  os.chdir('C:\\Users\\17574\\Desktop')
3  import speech_recognition as sr
4  import pyaudio
5  with sr.Microphone() as source:
6      print("Ready? Say something quick")
7      myWords = sr.Recognizer().listen(source)
8      print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
9  with open("myAudio.wav", "wb") as file_:
10     file_.write(myWords.get_wav_data())
11 from playsound import playsound
12 playsound('myAudio.wav')
```
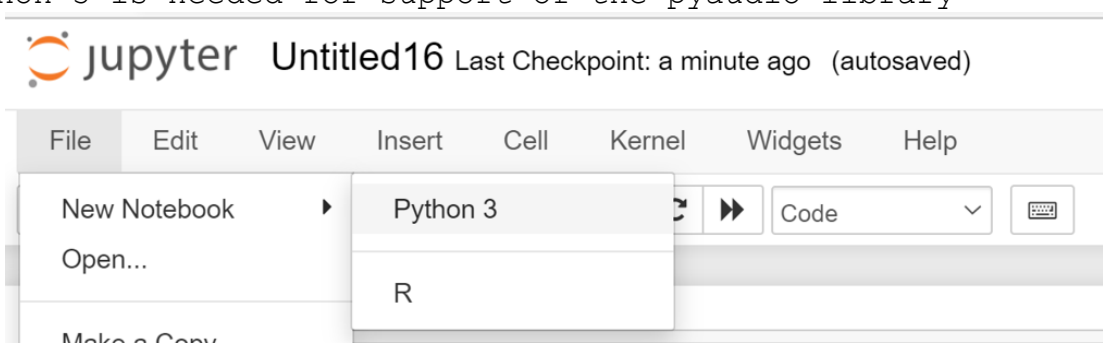
Part II Describe the program and its operating parameters

Objectives:

1. Demonstrate how to create a small Python program, called a **script**, and generate speech to text and text to audio results.
2. Challenge a user to replicate proper syntax, indenting, and other **coding idioms** to ensure programs run as intended.
3. Educate on basic data encoding where **binary** (1 or 0) is used for pictures/voice and **nonbinary** (byte/collations) is for text.
4. Educate on how libraries simplify program feature engineering making the art of the possible a far less daunting task.

ID	Purpose	Code
| 1<br>2<br>3<br>4<br>5<br>6<br>7<br>8<br>9<br>10<br>11<br>12 | (Library) computer features<br>• **instruction where to store a file**<br>(Library) speech generation<br>(Library) audio generation<br><br>>method to activate microphone<br>>ask a user to speak<br>>translate what user speaks<br>>display what user speaks<br><br>>>create an audio file<br><br>(Library) audio generation<br>• **play audio file back to user** | ```
1 import os
2 os.chdir('C:\\Users\\17574\\Desktop')
3 import speech_recognition as sr
4 import pyaudio
5 with sr.Microphone() as source:
6     print("Ready? Say something quick")
7     myWords = sr.Recognizer().listen(source)
8     print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
9 with open("myAudio.wav", "wb") as file_:
0     file_.write(myWords.get_wav_data())
1 from playsound import playsound
12 playsound('myAudio.wav')
``` |

Scenario 1: Generate a working program in a Python integrated development environment (IDE) such as Anaconda. The following example uses the Jupyter notebook program as part of the Anaconda Install.

Step	Task
1.	• Open Jupyter notebook and select File\New Notebook \ Python 3 • Python 3 is needed for support of the pyaudio library ![Jupyter Untitled16 Last Checkpoint: a minute ago (autosaved). Menu bar: File Edit View Insert Cell Kernel Widgets Help. New Notebook ▶ Python 3 / R, Open..., Make a Copy]

- Use the "+" symbol to add separate instruction statements
- Use this to make sure each code section runs
- Once completed will run like a pro in the last section for a seamless interaction
- Desired outcome: Speak vocally at computer => see text on screen => hear audio from machine

2.
```python
""" Part 1: Set Computer File Directory
    os=operating system"""
import os
os.chdir('C:\\Users\\17574\\Desktop')
```

3.
```python
""" Part 2: Set Google Speech Recognition
            and Microphone Library Functions """
import speech_recognition as sr
import pyaudio
```

4.
```python
""" Part 3: Ask user to same something
    use Google speech to parse words"""
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
```

5.
```python
"""Part 4: Encode words into audio file
    audio data is binary so add 'wb'
    for 'write binary data (1 or 0)"""
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
```

6.
```python
"""Part 5: Import a generic microphone module """
from playsound import playsound
playsound('myAudio.wav')
```

| **Run like a Pro** | ```python
""" Run like a Pro """
import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr
import pyaudio
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
from playsound import playsound
playsound('myAudio.wav')
```<br><br>Ready? Say something quick<br>You Said...: I like cake |

Review

Scenario 2: Expand code requiring 2 audio requests but deliver a single audio outcome file

Hint: The trick of this scenario is to create 2 separate myWords variables.

- In Python variables are either implicitly or explicitly declared.
- Code line 7 "my Words" is an implicit declaration as its type is not declared, such a character (char) or number
- Add a "_1" to the variable and then duplicate code lines 5-8 with a second variable myWords_2
- Finally, combine the myWords_1 with myWords_2 into myWords to deliver the audio output

Step	Task
1.	Duplicate code rows 5 to 8
2.	Paste after row 9 so new line is one row 9
3.	Change names of myWords to: myWords_1 and myWords_2
4.	Add a new line of code on line 13 • myWords = myWords_1 + myWords_2

| **Result** | ```
1 import os
2 os.chdir('C:\\Users\\17574\\Desktop')
3 import speech_recognition as sr
4 import pyaudio
5 with sr.Microphone() as source:
6     print("Ready? Say something quick")
7     myWords_1 = sr.Recognizer().listen(source)
8     print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
9 with sr.Microphone() as source:
10     print("Ready? Say something quick")
11     myWords_2 = sr.Recognizer().listen(source)
12     print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
13 myWords = myWords_1 + myWords_2
14 with open("myAudio.wav", "wb") as file_:
15     file_.write(myWords.get_wav_data())
16 from playsound import playsound
17 playsound('myAudio.wav')
``` |
| **Jupyter Notebook** | ```
import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr
import pyaudio
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_1 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_2 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
myWords = myWords_1 + myWords_2
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
from playsound import playsound
playsound('myAudio.wav')
```

```
Ready? Say something quick
You Said...: Nacho
Ready? Say something quick
You Said...: Nacho
``` |

Part III Provide a learner quiz and answer exemplar

Q1) What does "wb" stand for and why was it used in creation of the audio file?

Answer: wb = write binary. While text data is coded in nonbinary constructs of collations or characters, audio and image data are encoded in binary format (1 or 0). (29 words)

Q2) What is the main difference between the use of double quotations (" ") and single quotations (' ')?
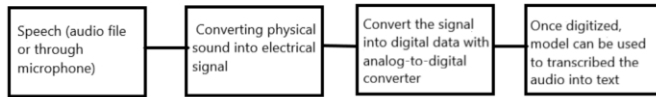
Answer: in python double quotations denotes text strings and code methods or instructions require string values. Single quotes is reserved for inserting paths or file names into instruction command consistent with underlying operating system language. (35 words)

Q3) [extra points] what is meant by phrase 'coding idioms' in the 2ⁿᵈ objective statement at start of Part 2?

Answer: coding or programming idioms is a semantic structure of a programming language. A semantic structure represents a combination of indentation, parenthesis, brackets, single/double quotes, and anaphora combinations. (28 words)

How does Speech recognition work?

| Speech (audio file or through microphone) | → | Converting physical sound into electrical signal | → | Convert the signal into digital data with analog-to-digital converter | → | Once digitized, model can be used to transcribed the audio into text |

Speech Recognition process

Hidden Markov Model (HMM), deep neural network models are used to convert the audio into text. A full detailed process is beyond the scope of this blog. In this blog, I am demonstrating how to convert speech to text using Python. This can be done with the help of the "*Speech Recognition*" API and "*PyAudio*" library.

```
Anaconda Prompt (Anaconda3)                                    —   □   ×

(base) C:\Users\17574>!pip install SpeechRecognition
'!pip' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Users\17574>pip install SpeechRecognition
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.8.1-py2.py3-none-any.whl (32.8 MB)
     |████████████████████████████████| 32.8 MB 6.4 MB/s
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.8.1
WARNING: You are using pip version 20.0.2; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\users\17574\anaconda3\python.exe -m pip install --upgrade pip' command.

(base) C:\Users\17574>
```

```
Anaconda Prompt (Anaconda3)

(base) C:\>
(base) C:\>cd Users\17574\Anaconda3

(base) C:\Users\17574\Anaconda3>
```

[python - Error While Installing Pyaudio in windows 10 - Stack Overflow](https://stackoverflow.com/questions/61461086/error-while-installing-pyaudio-in-windows-10)
https://stackoverflow.com/questions/61461086/error-while-installing-pyaudio-in-windows-10

```
pip install SpeechRecognition
pip install playsound
pip install pipwin
pipwin install pyaudio
```

(having issue so trying this one: ====➡>> conda install -c anaconda pyaudio

```
-------------------- The following packages will be downloaded:

    package                    |               build
    ---------------------------|-----------------
    conda-4.9.0                |               py37_0            3.1 MB
anaconda
    portaudio-19.6.0           |           he774522_4            240 KB
anaconda
    pyaudio-0.2.11             |      py37he774522_2             189 KB
anaconda
    ------------------------------------------------------------
                                        Total:            3.5 MB
The following NEW packages will be INSTALLED:
  portaudio          anaconda/win-64::portaudio-19.6.0-he774522_4
  pyaudio            anaconda/win-64::pyaudio-0.2.11-py37he774522_2
The following packages will be UPDATED:
  conda                               4.8.3-py37_0 --> 4.9.0-
py37_0
------------------------------------------------
```

```
Package `pyaudio` found in cache
Downloading package . . .
https://download.lfd.uci.edu/pythonlibs/x6hvwk7i/PyAudio-0.2.11-cp37-cp37m-win_amd64.whl
PyAudio-0.2.11-cp37-cp37m-win_amd64.whl
 [*] 111 kB / 111 kB @ 103 kB/s [#################] [100%, 0s left]
Processing c:\users\17574\pipwin\pyaudio-0.2.11-cp37-cp37m-win_amd64.whl
Installing collected packages: PyAudio
Successfully installed PyAudio-0.2.11

(base) C:\Users\17574\Anaconda3>
```

Gtts

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\17574\Anaconda3>pip install pyglet
Collecting pyglet
  Downloading pyglet-1.5.22-py3-none-any.whl (1.1 MB)
     ---------------------------------------- 1.1/1.1 MB 7.2 MB/s eta 0:00:00
Installing collected packages: pyglet
Successfully installed pyglet-1.5.22

(base) C:\Users\17574\Anaconda3>
```

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\17574\Anaconda3>pip install playsound
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7033 sha256=6d83a5188b82e8d462a19a35e16c9a
69cd80e1873ded2463f41ab3360c496cb3
  Stored in directory: c:\users\17574\appdata\local\pip\cache\wheels\ba\f8\bb\ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e
22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0

(base) C:\Users\17574\Anaconda3>
```

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\17574\Anaconda3>pip install gtts
Collecting gtts
  Downloading gTTS-2.2.3-py3-none-any.whl (25 kB)
Requirement already satisfied: click in c:\users\17574\anaconda3\lib\site-packages (from gtts) (7.0)
Requirement already satisfied: requests in c:\users\17574\anaconda3\lib\site-packages (from gtts) (2.22.0)
Requirement already satisfied: six in c:\users\17574\anaconda3\lib\site-packages (from gtts) (1.12.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\17574\anaconda3\lib\site-packages (from requests->gtts) (2
019.9.11)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\17574\anaconda3\lib\site-packages (from requests->gtts)
 (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\17574\anaconda3\lib\site-packages (fr
om requests->gtts) (1.24.2)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\17574\anaconda3\lib\site-packages (from requests->gtts) (2.8)
Installing collected packages: gtts
Successfully installed gtts-2.2.3

(base) C:\Users\17574\Anaconda3>
```

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\17574\Anaconda3>pip install pyglet
Collecting pyglet
  Downloading pyglet-1.5.22-py3-none-any.whl (1.1 MB)
     -------------------------------------- 1.1/1.1 MB 7.2 MB/s eta 0:00:00
Installing collected packages: pyglet
Successfully installed pyglet-1.5.22

(base) C:\Users\17574\Anaconda3>
```

1. Import Speech recognition library
2. Initializing recognizer class in order to recognize the speech. We are using google speech recognition.
3. Audio file supports by speech recognition: *wav, AIFF, AIFF-C, FLAC*. I used 'wav' file in this example
4. I have used 'taken' movie audio clip which says "*I don't know who you are I don't know what you want if you're looking for ransom I can tell you I don't have money*"

By default, google recognizer reads English. It supports different languages, for more details please check this [documentation](#).

Appendix – Script for running with a Command Prompt

Use the embedded file to run on a terminal or command prompt with Python 3 installed. Ensure to install Python libraries using pip automation on a command prompt. For example:


```
Anaconda Prompt (Anaconda3)

(base) C:\Users\17574\Anaconda3>pip install pyglet
Collecting pyglet
  Downloading pyglet-1.5.22-py3-none-any.whl (1.1 MB)
     ------------------------------------ 1.1/1.1 MB 7.2 MB/s
Installing collected packages: pyglet
Successfully installed pyglet-1.5.22
```

Words: - (29+35+28) <answer words>=

For more information see, https://realpython.com/what-is-pip/ .

Command Instruction	File	Pip Install Files & Sequence
py C:\Users\17574\Desktop\my_program.py	my_program.py	pip install SpeechRecognition pip install playsound pip install pipwin pipwin install pyaudio

```
# Command Prompt instructions
# py C:\Users\17574\Desktop\my_program.py

""" Part 1: Import Library Facilitators"""
import os
import pyaudio
import speech_recognition
from playsound import playsound

""" Part 2: Set library features """
import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr  #google speec recogn.
r = sr.Recognizer() # Initialize speech recognizing

""" Part 3: Talk and Record into Computer Microphone """
#import pyaudio
with sr.Microphone() as source:
    print("Say something quick !")
    audio_text = r.listen(source)
```

```python
    print("Thank you, just a second")
    try:
        # using google speech recognition
        print("Text: "+r.recognize_google(audio_text))
    except:
         print("Sorry, I did not get that")


""" Part 4: Have Computer Play Back Message"""
with open("mySound.wav", "wb") as f_ile:
    f_ile.write(audio_text.get_wav_data())
#from playsound import playsound
playsound('C:\\Users\\17574\\Desktop\\mySound.wav')

#---------------Additional Information---------------------#
"""
Google search => what does 'wb' mean in python?
while binary mode must be used when writing non-text files like
images.
The wb indicates that the file is opened for writing in _
binary mode.
When writing in binary mode, Python makes no _
changes to data as it is written to the file. In text mode _
 (when the b is excluded as in just w or when you specify text_
 mode with wt), however, Python will encode the text based _
on the default text encoding.
Additionally, Python will _
convert line endings (\n) to whatever the platform-specific_
line ending is, which would corrupt a binary file like an _
exe or png file. Text mode should therefore be used when _
writing text files (whether using plain text or a text-based_
format like CSV),while binary mode must be used when writing _
non-text files like images.
References:
https://stackoverflow.com/questions/2665866/what-does-wb-mean-in-
this-code-using-python
https://docs.python.org/3/tutorial/inputoutput.html#reading-and-
writing-files
https://docs.python.org/3/library/functions.html#open
edited Oct 1, 2019
"""
```