Python JARVIS Text to Speech CLI

Contents

Time required: 60 minutes

This inspiration for this series of tutorials is from:

https://www.freecodecamp.org/news/python-project-how-to-build-your-own-jarvis-using-python/

The JARVIS Project and The Road to OOP

This is a fun tutorial that is a first step to building your own JARVIS (Iron Man) personal assistant. We are also covering how to transform a program from:

- Top to bottom script
- Functional program
- OOP

We want to be able to take any Python script that we find and transform it into an OOP program. This makes it much easier to move into a GUI or Web Based application. OOP makes the core code more modular and portable. The only change we make is the interface.

Requirements Update

Requirements for tutorials and assignments are mean to be minimum guidelines. As long as you meet the requirements, you are free to explore, expand and modify your programs. That is where a large part of higher-level learning takes place. You are beyond the need for step-by-step directions for everything.

Comment your code so I or anything else can understand and carry on with your program.

Take the concepts you have learned; start creating your own unique, creative versions of the assignments.

pyttsx3

Our first step to building a DIY Iron Man suit is to get our computer to speak to us.

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.

The pyttsx3 module supports two voices in Windows. One is female and the second is male which is provided by “sapi5” for windows.

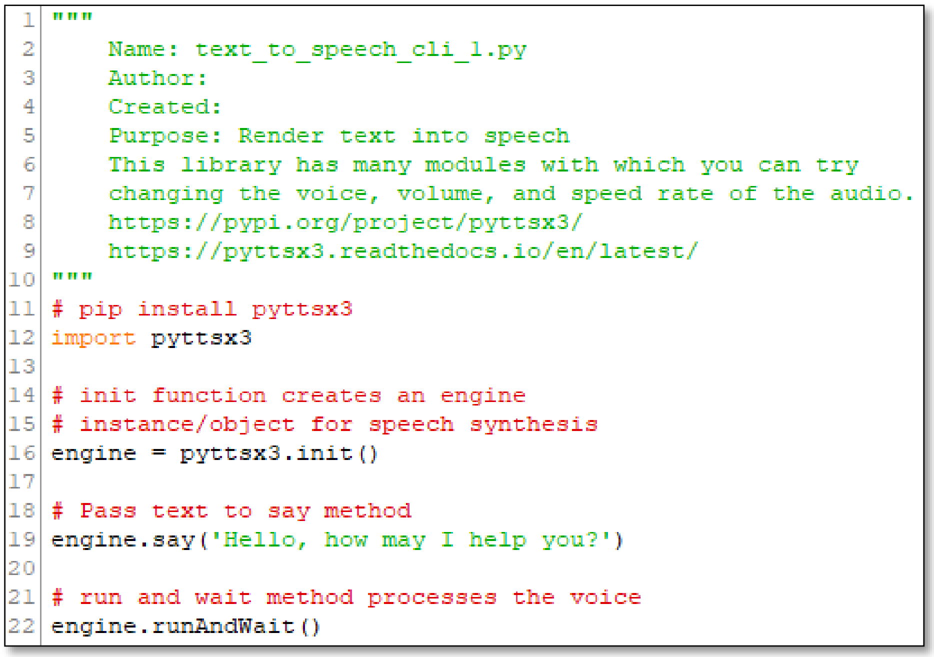
There are a lot of possibilities for creative programs using this library.

Pyttxs3 supports three TTS (Text to Speech) engines:

- sapi5 – SAPI5 on Windows
- nsss – NSSpeechSynthesizer on Mac OS X
- espeak – eSpeak on every other platform

Tutorial 1: Text to Speech CLI

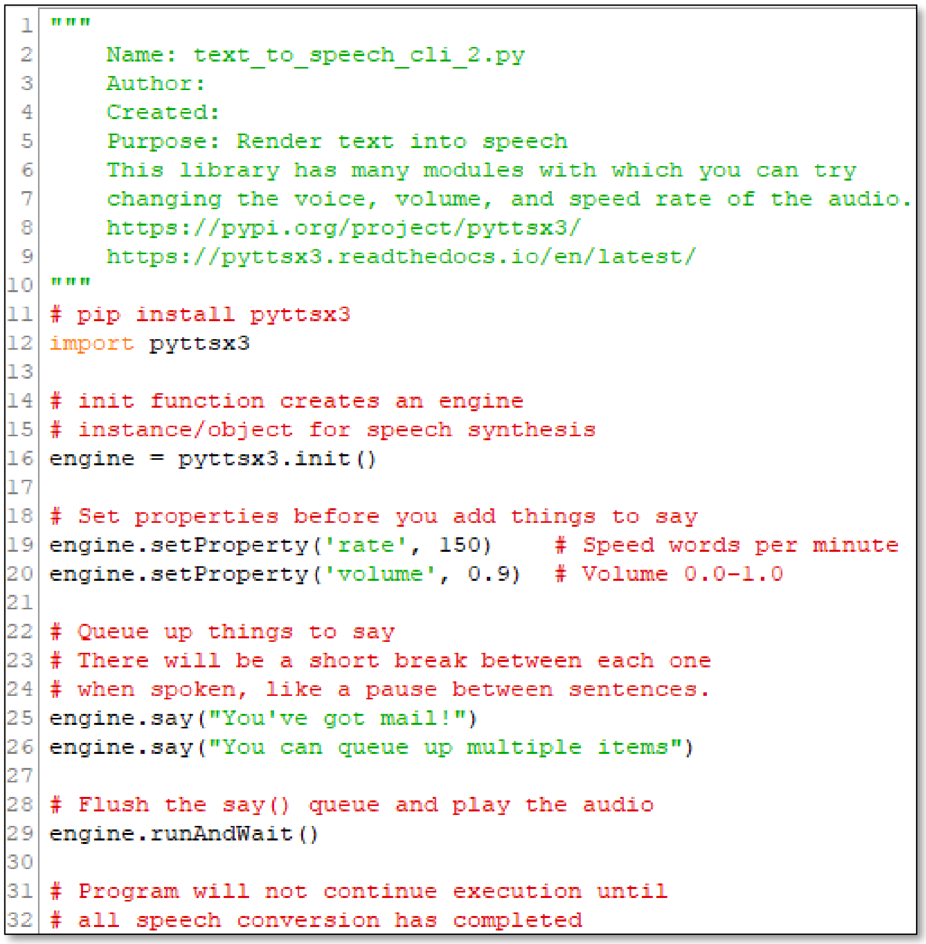
1. Install pyttsx: **pip install pyttsx3**



```python
"""
    Name: text_to_speech_cli_1.py
    Author:
    Created:
    Purpose: Render text into speech
    This library has many modules with which you can try
    changing the voice, volume, and speed rate of the audio.
    https://pypi.org/project/pyttsx3/
    https://pyttsx3.readthedocs.io/en/latest/
"""
# pip install pyttsx3
import pyttsx3

# init function creates an engine
# instance/object for speech synthesis
engine = pyttsx3.init()

# Pass text to say method
engine.say('Hello, how may I help you?')

# run and wait method processes the voice
engine.runAndWait()
```

Tutorial 2: Text to Speech

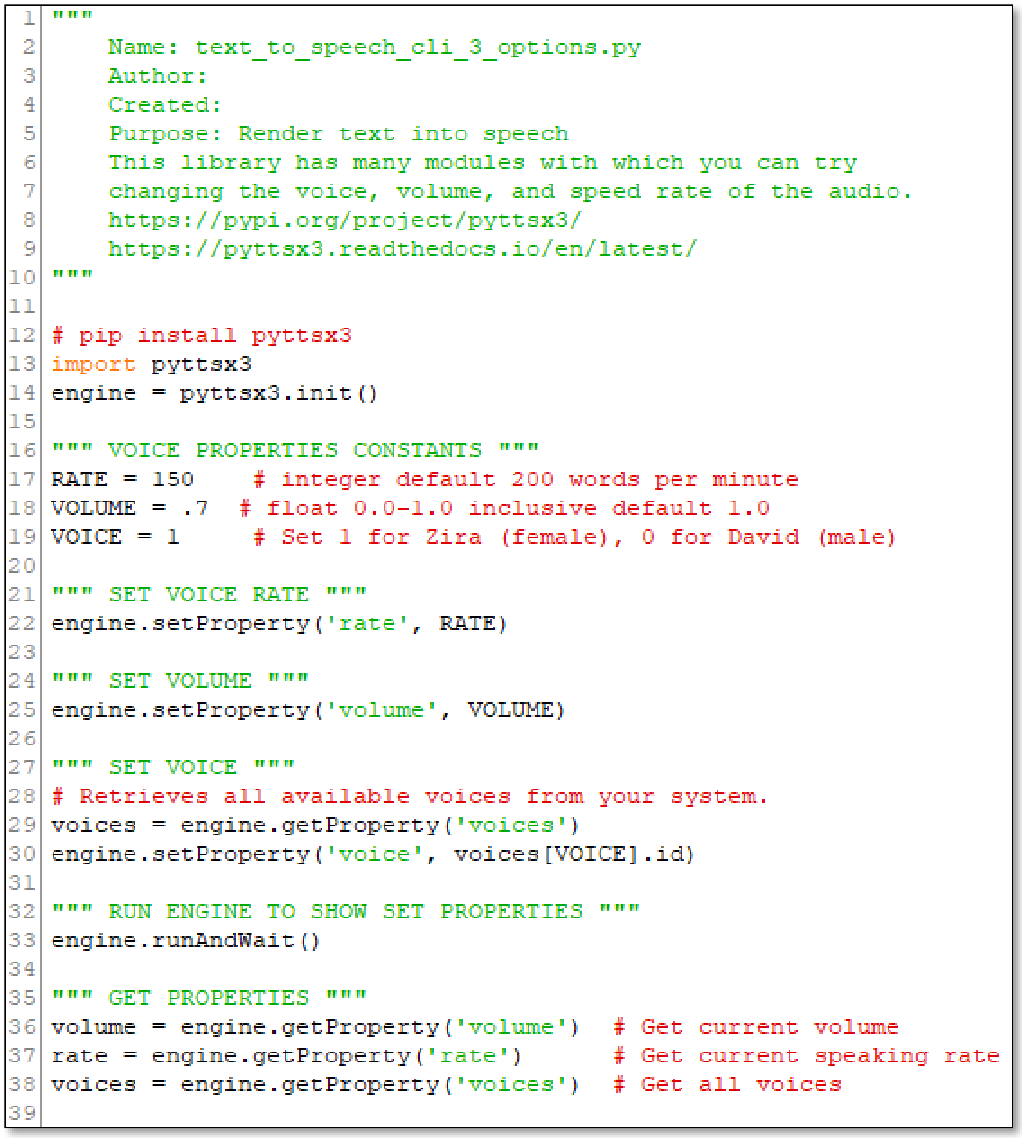
Let’s add some options.



```python
"""
    Name: text_to_speech_cli_2.py
    Author:
    Created:
    Purpose: Render text into speech
    This library has many modules with which you can try
    changing the voice, volume, and speed rate of the audio.
    https://pypi.org/project/pyttsx3/
    https://pyttsx3.readthedocs.io/en/latest/
"""
# pip install pyttsx3
import pyttsx3

# init function creates an engine
# instance/object for speech synthesis
engine = pyttsx3.init()

# Set properties before you add things to say
engine.setProperty('rate', 150)     # Speed words per minute
engine.setProperty('volume', 0.9)  # Volume 0.0-1.0

# Queue up things to say
# There will be a short break between each one
# when spoken, like a pause between sentences.
engine.say("You've got mail!")
engine.say("You can queue up multiple items")

# Flush the say() queue and play the audio
engine.runAndWait()

# Program will not continue execution until
# all speech conversion has completed
```

Tutorial 3: Text to Speech Options and Save to MP3

Don’t forget the troubleshooting technique of commenting out lines if you are having trouble.

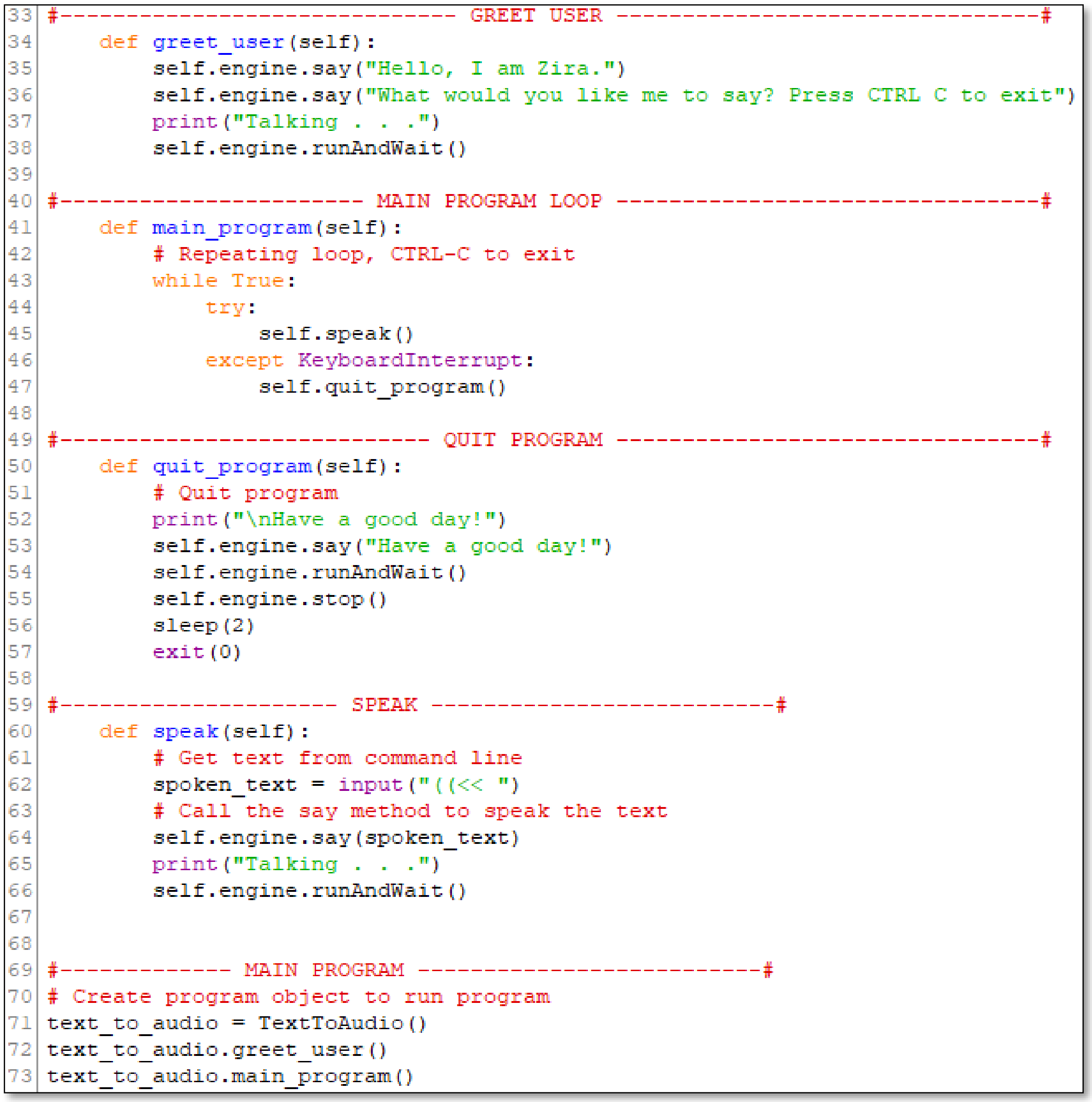


```python
"""
    Name: text_to_speech_cli_3_options.py
    Author:
    Created:
    Purpose: Render text into speech
    This library has many modules with which you can try
    changing the voice, volume, and speed rate of the audio.
    https://pypi.org/project/pyttsx3/
    https://pyttsx3.readthedocs.io/en/latest/
"""

# pip install pyttsx3
import pyttsx3
engine = pyttsx3.init()

""" VOICE PROPERTIES CONSTANTS """
RATE = 150     # integer default 200 words per minute
VOLUME = .7  # float 0.0-1.0 inclusive default 1.0
VOICE = 1      # Set 1 for Zira (female), 0 for David (male)

""" SET VOICE RATE """
engine.setProperty('rate', RATE)

""" SET VOLUME """
engine.setProperty('volume', VOLUME)

""" SET VOICE """
# Retrieves all available voices from your system.
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[VOICE].id)

""" RUN ENGINE TO SHOW SET PROPERTIES """
engine.runAndWait()

""" GET PROPERTIES """
volume = engine.getProperty('volume')  # Get current volume
rate = engine.getProperty('rate')      # Get current speaking rate
voices = engine.getProperty('voices')  # Get all voices
```

```python
""" Display PROPERTIES """
# Display all voices
for voice in voices:
    print(voice.name)
print(rate)
print(volume)

engine.say("Hello World!")
engine.say("My name is Zira.")
engine.say(f"My current speaking rate is {rate}")
engine.say(f"My current volume is {volume}")
engine.runAndWait()

""" SAVE VOICE TO MP3 FILE """
# On linux make sure that 'espeak' and 'ffmpeg' are installed
engine.save_to_file('Hello World', 'test.mp3')
engine.runAndWait()
# Stop the speech synthesis engine
engine.stop()
```

Tutorial 4: Text to Speech Interactive OOP

```python
"""
    Name: text_to_speech_cli_4.py
    Author:
    Created:
    Purpose: Render text into speech
    This library has many modules with which you can try
    changing the voice, volume, and speed rate of the audio.
    https://pypi.org/project/pyttsx3/
    https://pyttsx3.readthedocs.io/en/latest/
"""
from sys import exit
from time import sleep
# pip install pyttsx3
import pyttsx3


class TextToAudio:
    def __init__(self):
        # Change these constants to experiment with the speech engine
        RATE = 150     # integer default 200 words per minute
        VOLUME = 0.9   # float 0.0-1.0 inclusive default 1.0
        VOICE = 1      # Set 1 for Zira (female), 0 for David (male)
        # Initialize the pyttxs3 voice engine
        self.engine = pyttsx3.init()
        self.engine.setProperty('rate', RATE)
        self.engine.setProperty('volume', VOLUME)
        # Retrieves all available voices from your system.
        voices = self.engine.getProperty('voices')
        self.engine.setProperty('voice', voices[VOICE].id)
        # Run engine to set properties
        self.engine.runAndWait()

```



```
#------------------------------- GREET USER --------------------------------#
    def greet_user(self):
        self.engine.say("Hello, I am Zira.")
        self.engine.say("What would you like me to say? Press CTRL C to exit")
        print("Talking . . .")
        self.engine.runAndWait()

#------------------------ MAIN PROGRAM LOOP --------------------------------#
    def main_program(self):
        # Repeating loop, CTRL-C to exit
        while True:
            try:
                self.speak()
            except KeyboardInterrupt:
                self.quit_program()

#----------------------------- QUIT PROGRAM --------------------------------#
    def quit_program(self):
        # Quit program
        print("\nHave a good day!")
        self.engine.say("Have a good day!")
        self.engine.runAndWait()
        self.engine.stop()
        sleep(2)
        exit(0)

#---------------------- SPEAK --------------------------#
    def speak(self):
        # Get text from command line
        spoken_text = input("((<< ")
        # Call the say method to speak the text
        self.engine.say(spoken_text)
        print("Talking . . .")
        self.engine.runAndWait()


#------------- MAIN PROGRAM -------------------------#
# Create program object to run program
text_to_audio = TextToAudio()
text_to_audio.greet_user()
text_to_audio.main_program()
```

GitHub Presence

One of our unofficial projects this semester is to work on building everyone's personal GitHub repository.

GitHub is a presence and resume for programmers and their skills. Look at my GitHub. Everything in there is either my creation or a fork of something from GitHub.

www.github.com/itinstructor

Give credit where credit is due. Look at the copyright. Don't publish someone's else's work as your own.

I use Creative Commons copyright for any projects I publish on GitHub.

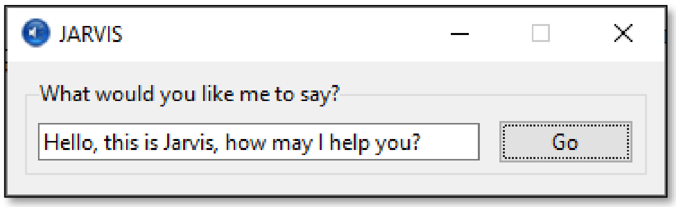
https://creativecommons.org/licenses/by/3.0/us/

Assignment: JARVIS Tkinter

Put your Tkinter skills to the test. Use what you learned and create a Tkinter version of JARVIS as we go through the JARVIS project.

We are building this project in OOP, which makes it easy to convert to a Tkinter GUI. The code is the same, the interface is all that has changed.

Here is one possible design from a student.


JARVIS
What would you like me to say?
Hello, this is Jarvis, how may I help you?
Go

Assignment Submission

Attach all program files to the assignment in BlackBoard.