Python NATO ICAO Encoder Tutorial

Contents

Time required: 60 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Requirements

The North America Treaty Organization (NATO) Phonetic Alphabet is the most widely used spelling alphabet. A spelling alphabet (aka radio alphabet, or telephone alphabet) is a set of words used to stand for the letters of an alphabet in oral communication. Each word in the spelling alphabet typically replaces the name of the letter with which it starts. It is used to spell out words when speaking to someone not able to see the speaker, or when the audio channel is not clear.

The International Civil Aviation Organization (ICAO) Alphabet is a series of words which are used to represent each letter of the alphabet. These are used in critical radio communications between airplanes and ground, and between airplanes in flight to avoid misunderstanding.

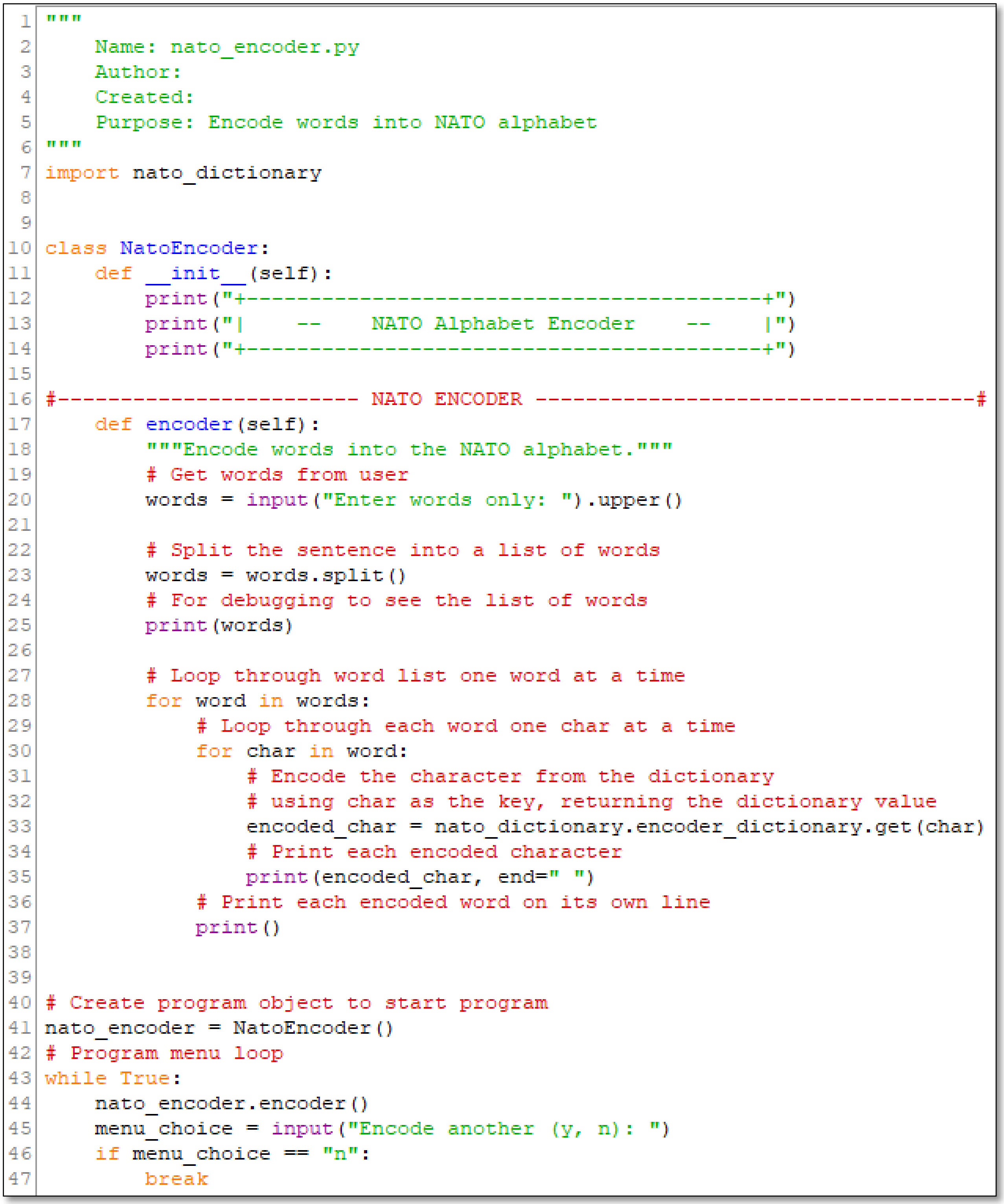
Tutorial 1: NATO Phonetic Alphabet Encoder

The code words are given in the Python dictionary below. You can copy and paste this dictionary into your program.

Create a Python file named: **nato_dictionary.py**

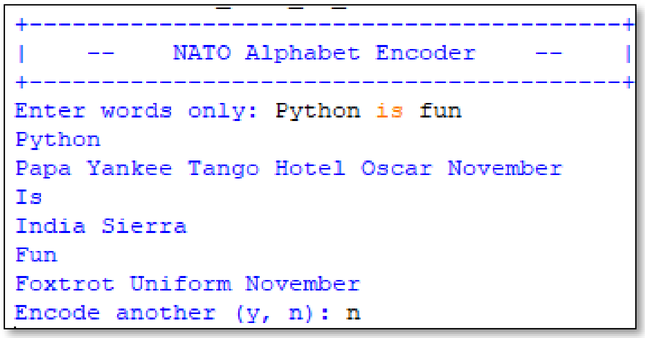
```
"""
    filename: nato_dictionary.py
    NATO phonetic alphabet
"""
encoder_dictionary = {
    'A': 'Alpha',  'B': 'Bravo',   'C': 'Charlie',
    'D': 'Delta',  'E': 'Echo',    'F': 'Foxtrot',
    'G': 'Golf',   'H': 'Hotel',   'I': 'India',
    'J': 'Juliett','K': 'Kilo',    'L': 'Lima',
    'M': 'Mike',   'N': 'November','O': 'Oscar',
    'P': 'Papa',   'Q': 'Quebec',  'R': 'Romeo',
    'S': 'Sierra', 'T': 'Tango',   'U': 'Uniform',
    'V': 'Victor', 'W': 'Whiskey', 'X': 'X-ray',
    'Y': 'Yankee', 'Z': 'Zulu'
}
```

Create a Python file named: nato_encoder.py



```python
"""
    Name: nato_encoder.py
    Author:
    Created:
    Purpose: Encode words into NATO alphabet
"""
import nato_dictionary


class NatoEncoder:
    def __init__(self):
        print("+-----------------------------------------+")
        print("|    --    NATO Alphabet Encoder    --    |")
        print("+-----------------------------------------+")

#------------------------ NATO ENCODER -----------------------------------#
    def encoder(self):
        """Encode words into the NATO alphabet."""
        # Get words from user
        words = input("Enter words only: ").upper()

        # Split the sentence into a list of words
        words = words.split()
        # For debugging to see the list of words
        print(words)

        # Loop through word list one word at a time
        for word in words:
            # Loop through each word one char at a time
            for char in word:
                # Encode the character from the dictionary
                # using char as the key, returning the dictionary value
                encoded_char = nato_dictionary.encoder_dictionary.get(char)
                # Print each encoded character
                print(encoded_char, end=" ")
            # Print each encoded word on its own line
            print()


# Create program object to start program
nato_encoder = NatoEncoder()
# Program menu loop
while True:
    nato_encoder.encoder()
    menu_choice = input("Encode another (y, n): ")
    if menu_choice == "n":
        break
```

Example run:



```
+-----------------------------------------+
|    --    NATO Alphabet Encoder    --    |
+-----------------------------------------+
Enter words only: Python is fun
Python
Papa Yankee Tango Hotel Oscar November
Is
India Sierra
Fun
Foxtrot Uniform November
Encode another (y, n): n
```

pyttsx3

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.

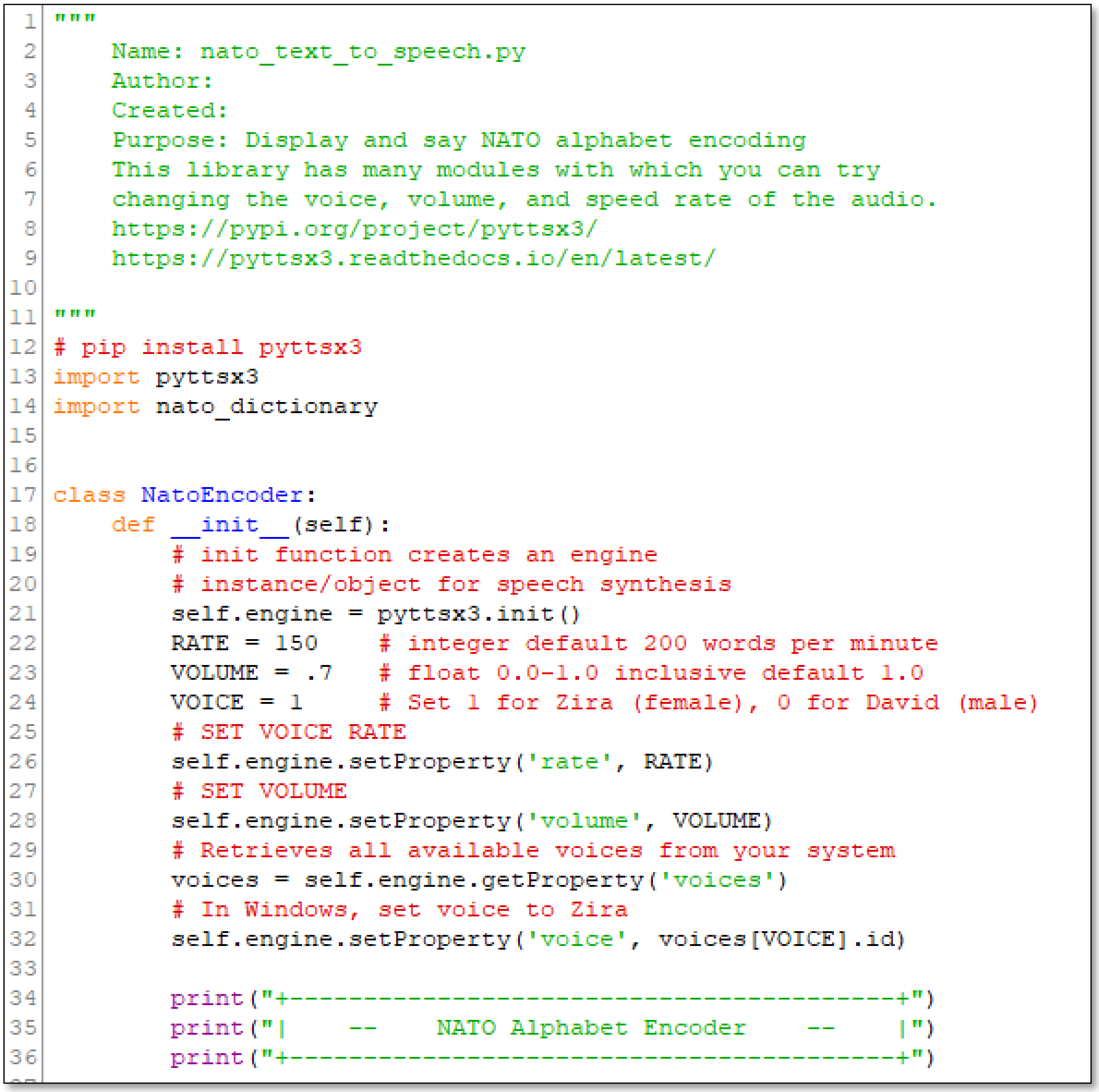
Pyttxs3 supports three TTS (Text to Speech) engines:

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

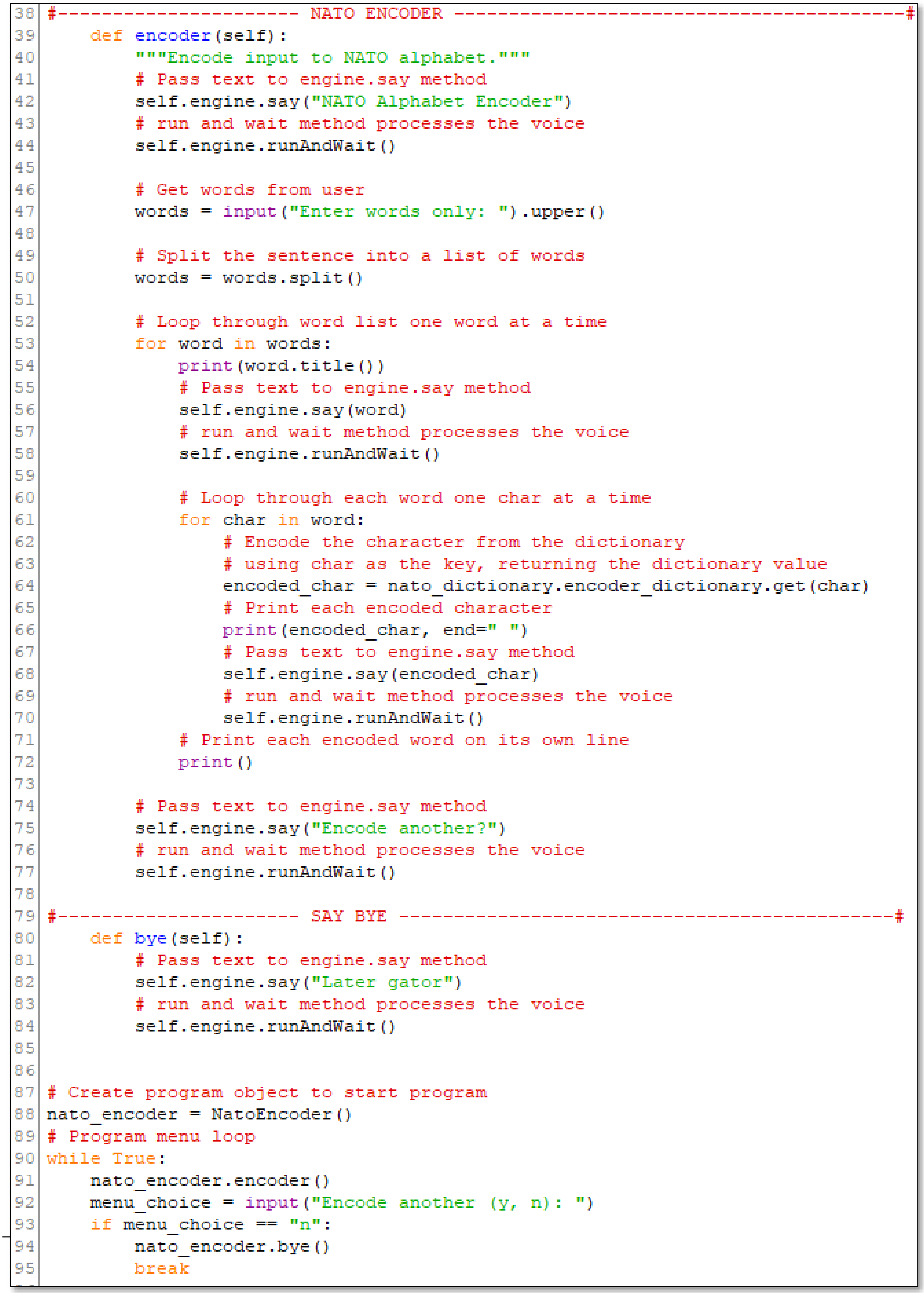
Tutorial 2: NATO Encoder Text to Speech

1. Install the **pyttsx** module: **pip install pyttsx3**
2. Copy **nato_encoder.py** to **nato_text_to_speech.py**

The following code modifies the NATO Converter to use text to speech.

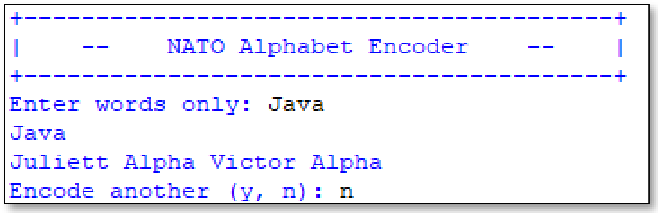


```python
"""
    Name: nato_text_to_speech.py
    Author:
    Created:
    Purpose: Display and say NATO alphabet encoding
    This library has many modules with which you can try
    changing the voice, volume, and speed rate of the audio.
    https://pypi.org/project/pyttsx3/
    https://pyttsx3.readthedocs.io/en/latest/

"""
# pip install pyttsx3
import pyttsx3
import nato_dictionary


class NatoEncoder:
    def __init__(self):
        # init function creates an engine
        # instance/object for speech synthesis
        self.engine = pyttsx3.init()
        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
        self.engine.setProperty('rate', RATE)
        # SET VOLUME
        self.engine.setProperty('volume', VOLUME)
        # Retrieves all available voices from your system
        voices = self.engine.getProperty('voices')
        # In Windows, set voice to Zira
        self.engine.setProperty('voice', voices[VOICE].id)

        print("+-----------------------------------------+")
        print("|    --    NATO Alphabet Encoder    --    |")
        print("+-----------------------------------------+")
```



```python
#---------------------- NATO ENCODER -----------------------------------------#
    def encoder(self):
        """Encode input to NATO alphabet."""
        # Pass text to engine.say method
        self.engine.say("NATO Alphabet Encoder")
        # run and wait method processes the voice
        self.engine.runAndWait()

        # Get words from user
        words = input("Enter words only: ").upper()

        # Split the sentence into a list of words
        words = words.split()

        # Loop through word list one word at a time
        for word in words:
            print(word.title())
            # Pass text to engine.say method
            self.engine.say(word)
            # run and wait method processes the voice
            self.engine.runAndWait()

            # Loop through each word one char at a time
            for char in word:
                # Encode the character from the dictionary
                # using char as the key, returning the dictionary value
                encoded_char = nato_dictionary.encoder_dictionary.get(char)
                # Print each encoded character
                print(encoded_char, end=" ")
                # Pass text to engine.say method
                self.engine.say(encoded_char)
                # run and wait method processes the voice
                self.engine.runAndWait()
            # Print each encoded word on its own line
            print()

        # Pass text to engine.say method
        self.engine.say("Encode another?")
        # run and wait method processes the voice
        self.engine.runAndWait()

#---------------------- SAY BYE ---------------------------------------------#
    def bye(self):
        # Pass text to engine.say method
        self.engine.say("Later gator")
        # run and wait method processes the voice
        self.engine.runAndWait()


# Create program object to start program
nato_encoder = NatoEncoder()
# Program menu loop
while True:
    nato_encoder.encoder()
    menu_choice = input("Encode another (y, n): ")
    if menu_choice == "n":
        nato_encoder.bye()
        break
```

Example run:



```
+------------------------------------------+
|    --    NATO Alphabet Encoder    --     |
+------------------------------------------+
Enter words only: Java
Java
Juliett Alpha Victor Alpha
Encode another (y, n): n
```

Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.