In [1]:

```
pip install pyttsx3
```

```
Requirement already satisfied: pyttsx3 in c:\users\berto\anaconda3\lib\site-p
ackages (2.90)
Requirement already satisfied: comtypes in c:\users\berto\anaconda3\lib\site-
packages (from pyttsx3) (1.4.1)
Requirement already satisfied: pypiwin32 in c:\users\berto\anaconda3\lib\site
-packages (from pyttsx3) (223)
Requirement already satisfied: pywin32 in c:\users\berto\anaconda3\lib\site-p
ackages (from pyttsx3) (305.1)
Note: you may need to restart the kernel to use updated packages.
```

In [43]:

```
import pyttsx3

tts = pyttsx3.init()

print('Enter the text to speak, or QUIT to quit')

while True:
    text = input('>>> ')
    if text.upper() == 'QUIT':
        print('Bye-bye')
        break
    else:
        tts.say(text)
        tts.runAndWait()
```

```
Enter the text to speak, or QUIT to quit
>>> hi
>>> hi
>>> QUIT
Bye-bye
```

In [45]:

```python
# Step 1: Set Speed Rate
# makes speed faster (set to 155)
import pyttsx3

tts = pyttsx3.init()

print('Enter the text to speak, or QUIT to quit')

while True:
    text = input('>>> ')
    tts.setProperty('rate',155)
    if text.upper() == 'QUIT':
        print('Bye-bye')
        break
    else:
        tts.say(text)
        tts.runAndWait()
```

```
Enter the text to speak, or QUIT to quit
>>> hello
>>> QUIT
Bye-bye
```

In [19]:

```python
# Step 1: Set Volume
# Sets volume very low
import pyttsx3

tts = pyttsx3.init()

print('Enter the text to speak, or QUIT to quit')

while True:
    text = input('>>> ')
    tts.setProperty('volume',0.2)
    if text.upper() == 'QUIT':
        print('Bye-bye')
        break
    else:
        tts.say(text)
        tts.runAndWait()
```

```
Enter the text to speak, or QUIT to quit
>>> why
>>> 0.2
>>> QUIT
Bye-bye
```

In [44]:

```
# Step 1: Set Voices
# Sets voice_id to Zira listed in next script
import pyttsx3

tts = pyttsx3.init()

print('Enter the text to speak, or QUIT to quit')

# Changed voice to Zira
voice_id = "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Speech\\Voices\\Tokens\\T
tts.setProperty('voice', voice_id)

while True:
    text = input('>>> ')
    if text.upper() == 'QUIT':
        print('Bye-bye')
        break
    else:
        tts.say(text)
        tts.runAndWait()
```

```
Enter the text to speak, or QUIT to quit
>>> hi
>>> QUIT
Bye-bye
```

In [37]:

```
# Step 1: Get Info of Voices
import pyttsx3

# Get information on voices
voice = tts.getProperty('voices')

# Prints avaliable voices
print("Voices Avaliable")
for voice in voices:
    print(f"Voice name - {voice.name}")
    print(f"Voice ID = {voice.id}")
```

```
Voices Avaliable
Voice name - Microsoft David Desktop - English (United States)
Voice ID = HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_
EN-US_DAVID_11.0
Voice name - Microsoft Zira Desktop - English (United States)
Voice ID = HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_
EN-US_ZIRA_11.0
```

In [ ]:

```python
# Step 2: Transcribe block
# Defines variable
text_to_spell = "What does tt.runAndWait() do?."

# Uses tts to speak message defined in variable
tts.say(text_to_spell)
# Waits for tts to finish and then moves on
tts.runAndWait()

# Defines the spelling module
def spelling():
    # Speaks each character in the text
    for character in text_to_spell:
        # Uses tts to individually speak each character
        tts.say(character)
        # Waits for tts to finish and then moves on
        tts.runAndWait()

# Calls spelling function to spell each character individually
spelling()

# Uses tts to say the spelling completed to alert user
tts.say("Spelling complete")
# Waits for tts to finish and then moves on
tts.runAndWait()

# tts.runAndWait() Definition -
# A function that pauses the program until text to speech finishes
```

In [71]:

```
# Step 3: Transcribe Block
# Defines function
def question_to_user():
    # Establishes text-to-speech engine
    engine = pyttsx3.init()

    # Asks user to input their first name and assigns it a variable
    first_name = input("Enter your first name: ")
     # Asks user to input their last name and assigns it a variable
    last_name = input("Enter your last name: ")
     # Asks user to input their major and assigns it a variable
    major = input("Enter your intended major: ")

    # Creates new variable named message using variables from inputs above
    message = f"Hello, {first_name} {last_name}. Are you majoring in {major}?"

    # Prints message variable
    engine.say(message)
    # Waits for tts to finish and then moves on
    engine.runAndWait()

# Executes the function established above
question_to_user()
```

```
Enter your first name: Roberto
Enter your last name: Friedlander
Enter your intended major: Cybersecurity
```

In [69]:

```
# Step 4: Song Dictionary
import time

favorite_songs = {
    "Bohemian Rhapsody": "Queen",
    "Stairway to Heaven": "Led Zeppelin",
    "Billionare": "Bruno Mars",
    "Don't Stop Believin": "Journey",
    "Wonderwall": "Oasis",
}

def speak_songs():
    tts = pyttsx3.init()

    tts.say('Those are my five favorite songs!')
    tts.runAndWait()

    for song, artist in favorite_songs.items():
        message = f"My favorite song is {song} by {artist}."
        tts.say(message)
        print(message)
        # Added sleep timer so tts would read each song/artist properly
        time.sleep(1)

speak_songs()
```

```
My favorite song is Bohemian Rhapsody by Queen.
My favorite song is Stairway to Heaven by Led Zeppelin.
My favorite song is Billionare by Bruno Mars.
My favorite song is Don't Stop Believin by Journey.
My favorite song is Wonderwall by Oasis.
```

In [3]:

```
# Step 5: Inches to CM script
import pyttsx3
import time

def speak_height(height):
    cm = height * 2.54

    tts = pyttsx3.init()

    voice_id = "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Speech\\Voices\\Token
    tts.setProperty('voice', voice_id)
    tts.setProperty('rate', 180)
    tts.setProperty('volume', 1.0)

    tts.say(f"Your height in centimeters is {cm}!")
    tts.runAndWait()
    time.sleep(1)

    print(f"Your height in centimeters is {cm}!")

height = float(input("Please enter your height in inches: "))
speak_height(height)
```

```
Please enter your height in inches: 83
Your height in centimeters is 210.82!
```

In [ ]: