Documentation for Speech Recognition and Text-to-Speech Program

Overview:

This Python program uses two libraries: **SpeechRecognition** and **pyttsx3** to capture audio input from the user through a microphone, convert it into text, and then convert that text into speech. The program listens continuously to the microphone and processes the speech until manually interrupted.

Libraries Used:

1. **SpeechRecognition** (`speech_recognition`):
    - This library is used to recognize speech from audio and convert it into text.
    - The `recognize_google()` function is used to send audio input to Google's speech recognition service and return the recognized text.
2. **pyttsx3**:
    - This library is used to convert text into speech.
    - It initializes a speech engine to speak the recognized text.

Code Explanation:
Import Libraries:

```
 import speech_recognition as sr
import pyttsx3
```

1.
    - `speech_recognition` is imported as `sr` to handle speech recognition.
    - `pyttsx3` is imported for converting text to speech.

Initialize Recognizer:

```
 r = sr.Recognizer()
```

2.
    - An instance of the recognizer object is created, which will be used to process the audio.

Text-to-Speech Function (SpeakText):

```
def SpeakText(command):
  engine = pyttsx3.init()
  engine.say(command)
  engine.runAndWait()
```

3.
    - This function takes the recognized text (in the form of `command`) as input.
    - It initializes the `pyttsx3` engine, uses it to say the `command`, and waits until the speech has finished.

Continuous Speech Recognition Loop:

```
while(1):
  try:
    with sr.Microphone() as source2:
      r.adjust_for_ambient_noise(source2, duration=0.2)
      audio2 = r.listen(source2)
      MyText = r.recognize_google(audio2)
      MyText = MyText.lower()
      print("Did you say "+MyText)
      SpeakText(MyText)
  except sr.RequestError as e:
    print("Could not request results; {0}".format(e))
  except sr.UnknownValueError:
    print("unknown error occured")
```

4.
    - The program runs in an infinite loop (`while(1)`) to keep listening for speech input.
    - **Microphone as Source**:
        - The program listens to the microphone input using the `Microphone()` context manager.
        - `r.adjust_for_ambient_noise(source2, duration=0.2)` adjusts for ambient noise before capturing the audio.
        - `audio2 = r.listen(source2)` listens to the input and stores the audio data in `audio2`.
    - **Speech Recognition**:

        - `MyText = r.recognize_google(audio2)` sends the audio data to Google's speech recognition API to convert the speech into text.
        - The text is converted to lowercase using `MyText = MyText.lower()`.
    - **Output and Text-to-Speech**:
        - It prints the recognized text with `print("Did you say "+MyText)`.
        - The recognized text is passed to `SpeakText(MyText)` to be converted into speech.

5. **Error Handling:**
    - **RequestError**: If the speech recognition service fails to respond, the program catches the error and prints a message: `Could not request results`.
    - **UnknownValueError**: If the speech recognition system cannot understand the input, it catches the error and prints: `unknown error occurred`.

Example Usage:

- The program will continuously listen for the user's voice input, recognize what they say, and read it out loud. It will keep running until manually interrupted (e.g., by pressing `Ctrl+C`).