The `speech_recognition` Package

The `speech_recognition` package provides a high-level interface to record and process audio inputs in Python.

Reference:

- https://github.com/Uberi/speech_recognition
- http://people.csail.mit.edu/hubert/pyaudio/#downloads
- https://github.com/Uberi/speech_recognition/blob/master/reference/library-reference.rst
- https://github.com/Uberi/speech_recognition/blob/master/examples/write_audio.py
- https://github.com/Uberi/speech_recognition/blob/master/examples
- https://github.com/Uberi/speech_recognition/blob/master/speech_recognition/**main**.py
- https://github.com/s2t2/learning-new-sounds

Prerequisites

The `SpeechRecognition` Python package depends on another Python package called `pyaudio`, which itself depends on a lower-level library caled `portaudio` (not a Python package).

On Mac OS, use homebrew to install `portaudio`:

```
# Mac Terminal:
brew install portaudio
```

On Windows, install these Microsoft Visual C++ Build Tools, then afterwards, run the following Anaconda command to install `portaudio`:

```
# Windows Git Bash:
conda install -c conda-forge portaudio
```

Installation

Note: this package may not yet work in Python 3.7, so when creating your Anaconda virtual environment, specify version 3.6 instead:

```
conda create -n sounds-env python=3.6
conda activate sounds-env
```

Use Pip to install the package and its `pyaudio` dependency:

```
pip install pyaudio # depends on the lower-level "portaudio" library

pip install SpeechRecognition # depends on the "pyaudio" Python package
```

Usage

Recording Audio

Record audio using your computer's built-in microphone, and save that to a file:

```
import speech_recognition as sr

client = sr.Recognizer()

with sr.Microphone() as mic:
    print("Say something!")
    audio = client.listen(mic)

with open("my-recording.flac", "wb") as f:
    f.write(audio.get_flac_data())
```

Recognizing Speech

Record audio using your computer's built-in microphone, and recognize the spoken words:

```
import speech_recognition as sr

client = sr.Recognizer()

with sr.Microphone() as mic:
    print("Say something!")
    audio = client.listen(mic)

# returns the transcript with the highest confidence:
transcript = client.recognize_google(audio)
#> 'how old is the Brooklyn Bridge'

# returns all transcripts:
response = client.recognize_google(audio, show_all=True)
#> {
#>   'alternative': [
#>     {
#>       'transcript': 'how old is the Brooklyn Bridge',
#>       'confidence': 0.987629
```

```
#>     }
#>   ],
#>   'final': True
#> }
```