

Модуль 3

Урок 3. Стандартные и сторонние библиотеки Python

Библиотеки Python, использование для разных задач. Библиотека распознавания и воспроизведения речи, голосовой помощник.

На Python можно строить как простые программы, так и сложные ИИ системы. В данном занятии мы покажем, как реализовать распознавание голоса и выполнение различных команд. Подобные занятия можно проводить как отдельные мастер-классы как для новых учеников с достаточным уровнем подготовки, так и проводить в формате занятий для стандартного курса Python.

Гибкость и простота использования пакета SpeechRecognition делают его отличным выбором для любого проекта Python. Тем не менее, поддержка каждой функции каждого API, который он включает, не гарантируется. Вам нужно будет потратить некоторое время на изучение доступных вариантов, чтобы выяснить, будет ли SpeechRecognition работать в вашем конкретном случае.

Перед стартом работы, вам стоит убедиться в нескольких вещах:

- у вас установлен Python на вашем компьютере;
- у вас установлен текстовый редактор, к примеру PyCharm;
- у вас установлен пакетный менеджер Pip.

Установка библиотек

На Python можно строить как простые программы, так и сложные ИИ системы. В данной статье мы покажем как реализовать распознавание голоса и выполнение различных команд.

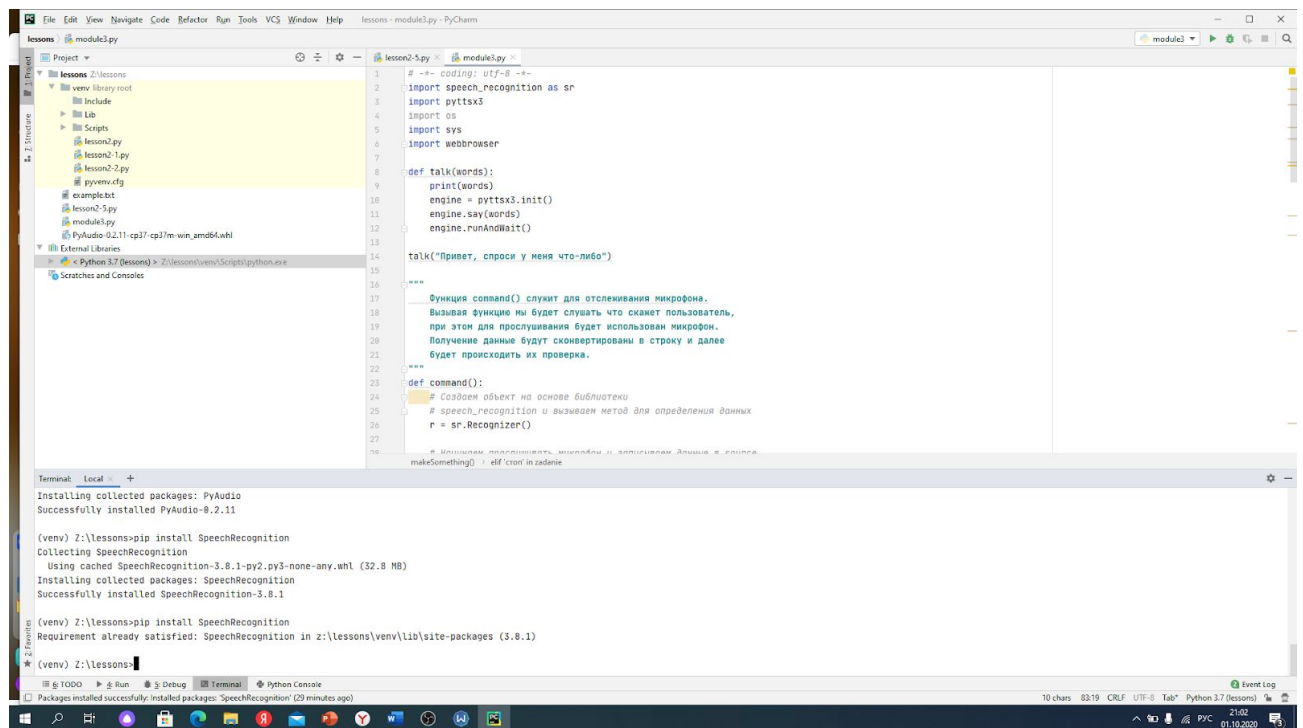
Для распознавания голоса вам необходимо установить библиотеки:

SpeechRecognition - команда `pip install SpeechRecognition`;

pytsx3 - команда `pip install pytsx3` (или для второй версии функции talk gTTS - команда `pip install gTTS`) проверить работу функции talk следует в первой части занятия, дальше не двигаться до тех пор, пока голосовое воспроизведение не заработает тем или иным способом.

PyAudio - команда `pip install PyAudio`.

Все библиотеки можно устанавливать через терминал в ваш проект через программу PyCharm.



После установки всех библиотек начните прописывать код самой программы. Ниже мы приведем весь код программы с комментариями, дабы вам было проще в нём разобраться:

Скачать библиотеку под вашу версию языка и системы можно по данной ссылке:

<https://www.lfd.uci.edu/~gothke/pythonlibs/#pyaudio>

В данном занятии я работала с версией библиотеки PyAudio-0.2.11-cp37-cp37m-win_amd64.whl

Для более простой установки после скачивания файла, можно скопировать его в основную папку вашего проекта, затем установить его, запустив код в терминале:

pip install PyAudio-0.2.11-cp37-cp37m-win_amd64.whl

Исходный код

```
# -*- coding: utf-8 -*-
```

```
import speech_recognition as sr
import pyttsx3
import os
import sys
import webbrowser
```

```
def talk(words):
    print(words)
    engine = pyttsx3.init()
```

```
engine.say(words)
engine.runAndWait()
```

```
talk("Привет, спроси у меня что-либо")
```

```
"""
```

```
Функция command() служит для отслеживания микрофона.
Вызывая функцию мы будем слушать что скажет пользователь,
при этом для прослушивания будет использован микрофон.
Получение данные будут сконвертированы в строку и далее
будет происходить их проверка.
```

```
"""
```

```
def command():
```

```
# Создаем объект на основе библиотеки
# speech_recognition и вызываем метод для определения данных
r = sr.Recognizer()
```

```
# Начинаем прослушивать микрофон и записываем данные в source
with sr.Microphone() as source:
```

```
# Просто вывод, чтобы мы знали когда говорить
talk("Говорите")
# Устанавливаем паузу, чтобы прослушивание
# началось лишь по прошествию 1 секунды
r.pause_threshold = 1
# используем adjust_for_ambient_noise для удаления
# посторонних шумов из аудио дорожки
#r.adjust_for_ambient_noise(source, duration=1)
# Полученные данные записываем в переменную audio
# пока мы получили лишь mp3 звук
audio = r.listen(source)
```

```
try: # Обрабатываем все при помощи исключений
```

```
"""
```

```
Распознаем данные из mp3 дорожки.
Указываем что отслеживаемый язык русский.
Благодаря lower() приводим все в нижний регистр.
Теперь мы получили данные в формате строки,
которые спокойно можем проверить в условиях
```

```
"""
```

```
zadanie = r.recognize_google(audio, language="ru-RU").lower()
# Просто отображаем текст что сказал пользователь
talk("Вы сказали: " + zadanie)
```

```
# Если не смогли распознать текст, то будет вызвана эта ошибка
except sr.UnknownValueError:
```

```
# Здесь просто проговариваем слова "Я вас не поняла"
```

```

        # и вызываем снова функцию command() для
        # получения текста от пользователя
        talk("Я вас не поняла")
        zadanie = command()

    # В конце функции возвращаем текст задания
    # или же повторный вызов функции
    return zadanie

# Данная функция служит для проверки текста,
# что сказал пользователь (zadanie - текст от пользователя)
def makeSomething(zadanie):
    # Попросту проверяем текст на соответствие
    # Если в тексте что сказал пользователь есть слова
    # "открыть сайт", то выполняем команду
    if 'открыть сайт' in zadanie:
        # Проговариваем текст
        talk("Уже открываю")
        # Указываем сайт для открытия
        url = 'https://music.yandex.ru'
        # Открываем сайт
        webbrowser.open(url)
    # если было сказано "стоп", то останавливаем прогу
    elif 'стоп' in zadanie:
        # Проговариваем текст
        talk("Да, конечно, без проблем")
        # Выходим из программы
        sys.exit()
    # Аналогично
    elif 'имя' in zadanie:
        talk("Меня зовут Катенька")

# Вызов функции для проверки текста будет
# осуществляться постоянно, поэтому здесь
# прописан бесконечный цикл while
while True:
    makeSomething(command())

```

Вторая версия функции talk:

```

# Подключение всех необходимых библиотек
# Нам нужно: speech_recognition, os, sys, webbrowser
# Для первой библиотеки прописываем также псевдоним
import speech_recognition as sr
import os

```

```
import sys
import webbrowser

# Функция, позволяющая проговаривать слова
# Принимает параметр "Слова" и проговаривает их
def talk(words):
    print(words) # Дополнительно выводим на экран
    os.system("say " + words) # Проговариваем слова

# Вызов функции и передача строки
# именно эта строка будет проговорена компьютером
talk("Привет, чем я могу помочь вам?")
```

Задание:

Установить необходимые библиотеки, взять за основу исходный код из данного занятия. Адаптировать скрипт текстового консультанта-помощника для магазина из модуля 1 для голосового помощника. Протестировать скрипт на ком-то из знакомых, доработать диалог.

Дополнительные материалы:

<https://habr.com/ru/company/toshibarus/blog/490732/>

<https://cyberguru.tech/программирование/машинное-обучение/окончательное-руководство-по-распоз>

Мастер-класс разработан на основании данного видео-материала:

[**https://www.youtube.com/watch?v=2eoudIBVW9w**](https://www.youtube.com/watch?v=2eoudIBVW9w)