**1 этап**

**Базовое задание**

Бот должен отвечать на сообщения:  
— "/start" → "Это бот-погода. Поможет узнать погоду в любом городе. Какой город интересует?"  
— "Москва" → "Сейчас отличная погода!"   
— "Москва завтра" → "Завтра еще лучше!"  
— Любое другое сообщение → "Я тебя не понял"

**2 этап**

**Базовое задание**

Доработаем, чтобы бот работал в несколько этапов:  
— "/start" → "Это бот-погода. Поможет узнать погоду в любом городе. Какой город интересует?"  
— "Москва" → "Сейчас отличная погода!"   
— "Москва завтра" → "Сейчас отличная погода!"  
— Любое другое сообщение → "Я тебя не понял"

Обратите внимание, что теперь бот до того как задаст вопрос — не должен реагировать на ответы на вопросы

**Дополнительное**

Бот должен распознавать сообщения с датой в виде "Москва 15.09". Нужно определить дату и ответить ботом, если дата отличается не больше чем на три дня от текущей.

Полезные ссылки:

* <https://docs.python.org/3/library/datetime.html>
* <https://dateutil.readthedocs.io/en/stable/>

**3 этап**

**Базовое задание**

Доработаем нашего бота, чтобы он брал вопросы из API

Основная ссылка:

<http://81.2.244.179/api/weather>

Пример использования:

import requests

api\_url = 'https://stepik.akentev.com/api/weather'

requests.get(

api\_url,

params={'city': 'Москва'}

).json()

В ответе приходит json, который можно перевести в структуры языка через .json()метод.

В поле description лежит описание погодных условий. В поле temp - температура в, а в weather общие погодные условия.

{'description': 'слегка облачно', 'temp': 20.78, 'weather': 'Clouds'}

Полезные ссылки:

* Запрос можно выполнить в браузере, но для удобного просмотра можно воспользоваться онлайн-редактором <https://jsoneditoronline.org/>

**Дополнительное**

Пользователь может узнать погоду на сегодня, завтра, послезавтра ... и так не более чем на 3 дня вперед. В запросе должен передаваться параметр forecast. 0 - сегодня, 1 - завтра, 2 - послезавтра, 3 - послепослезавтра

requests.get(

api\_url,

params={'city': 'Москва', 'forecast': 1}

).json()