SKILLFACTORY

Библиотека requests и json

requests.get(<url>)	Функция отправки гет запроса
requests.post(<url>, <data>)	Функция отправки пост-запроса
Requests.Response	Объект который возвращается после запроса. Его содержимо (html или json) хранится в поле .content
JSON	Особый формат записи объектов. Может быть похож на словарь или список в python.
json.loads(<json строка>)	Преобразует текст ответа json в python-объект (словарь или список, в зависимости от самого ответа)
Документация по библиотеке requests	https://requests.readthedocs.io/en/master/
Документация по библиотеке json	https://docs.python.org/3/library/json.html

Тестировщик-автоматизатор на Python

json.dumps(<python объект>)	Преобразует python объект в json

Парсинг данных с помощью lxml

Html	Древовидный язык гипертекстовой разметки. Состоит из тегов <тег (атрибуты)>текст<тег>
pip3 install lxml	Установка библиотеки для парсинга
import lxml.html	Импортирование библиотеки для парсинга html
.parse(<имя файла>, lxml.html.HtmlParser())	Функция которая возвращает дерево объект для парсинга.
.find(<название элемента>)	Найти тег в дереве (возвращает первый попавшийся)
.findall(<название элемента>)	Найти все теги (возвращает список)
lxml.html.document_fromstring(<html код>)	Возвращает объект для парсинга через строку
.xpath(<путь к элементу(можно скопировать из браузера)>)	Возвращает элемент, путь к которому был указан

Модуль Контрольный проект по ООП

Написание телеграмм бота

Зарегистрировать бота	@botfather
Токен	Строка для того чтобы обработчики срабатывали именно для данного конкретного бота
@bot.mesage_handler(filters)	Обработчик событий с определёнными фильтрами
filters	Бывает двух видов, content_types (типы воспринимаемого контента) и commands (команды это то что вводится через /) Оба являют собой списки
bot.send_message(<id чата>, <текст сообщения>)	Отправить сообщение в част
bot.reply_to(<сообщение>, <текст сообщения>)	Ответить на сообщение
Все типы content_types	Voice – голосовые сообщения, text – текст, audio – аудио файлы, photo – картинки и т.д.
Полная документация	https://github.com/eternnoir/pyTelegramBotAPI

Кэширование, redis

Установка ПО для начала работы	Linux – sudo apt-get install redis-server Windows – надо будет создавать базу данных в облаке https://app.redislabs.com/ MacOS – подходит любой из двух вариантов
Установка библиотеки для работы с redis в python	pip3 install redis
Создание курсора подключения	import redis red = redis.Redis('ваш хост', порт(число), password='ваш пароль')
Как узнать данные для подключения?	Для облачной базы данных данные указаны в описании. Хост – то что стоит до двоеточия в поле endpoint, порт - то что стоит после двоеточия в endpoint. На линукс достаточно ввести redis-cli в терминале. Хост – то что стоит до двоеточия в поле endpoint, порт - то что стоит после двоеточия в endpoint.

Запись нового ключа в кэш	red.set(<имя поля>, <значение>)
Получение значения из кэша	red.get(<имя поля>)
Удаление данных из кэша	red.delete(<имя ключа>)