



SKILLFACTORY

Библиотека requests и json

<code>requests.get(<url>)</code>	Функция отправки гет запроса
<code>requests.post(<url>, <data>)</code>	Функция отправки пост-запроса
<code>Requests.Response</code>	Объект который возвращается после запроса. Его содержимое (html или json) хранится в поле <code>.content</code>
JSON	Особый формат записи объектов. Может быть похож на словарь или список в python.
<code>json.loads(<json строка>)</code>	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

Fullstack разработчик на Python

<code>json.dumps(<python объект>)</code>	Преобразует python объект в json
--	----------------------------------

Парсинг данных с помощью lxml

Html	Древовидный язык гипертекстовой разметки. Состоит из тегов <code><тег (атрибуты)>текст<тег ></code>
<code>pip3 install lxml</code>	Установка библиотеки для парсинга
<code>import lxml.html</code>	Импортирование библиотеки для парсинга html
<code>.parse(<имя файла>, lxml.html.HtmlParser())</code>	Функция которая возвращает дерево объект для парсинга.
<code>.find(<название элемента>)</code>	Найти тег в дереве (возвращает первый попавшийся)
<code>.findall(<название элемента>)</code>	Найти все теги (возвращает список)
<code>lxml.html.document_fromstring(<html код>)</code>	Возвращает объект для парсинга через строку
<code>.xpath(<путь к элементу (можно скопировать из браузера)>)</code>	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
<code>@bot.message_handler(filters)</code>	Обработчик событий с определёнными фильтрами
filters	Бывает двух видов, <code>content_types</code> (типы воспринимаемого контента) и <code>commands</code> (команды это то что вводится через /) Оба являют собой списки
<code>bot.send_message(<id чата>, <текст сообщения>)</code>	Отправить сообщение в чат
<code>bot.reply_to(<сообщение>, <текст сообщения>)</code>	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 - <code>sudo apt-get install redis-server</code> 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	<code>pip3 install redis</code>
Создание курсора подключения	<pre>import redis red = redis.Redis('ваш хост', порт(число), password='ваш пароль')</pre>
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 <code>redis-cli</code> в терминале. Хост - то что стоит до двоеточия в поле endpoint, порт - то что стоит после двоеточия в endpoint.

Fullstack разработчик на Python

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

Модуль Контрольный проект по ООП