

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ  
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Дисциплина: «Языки программирования»

Отчет по лабораторной работе №16  
**Работа с данными формата JSON в языке Python**

Выполнила студентка группы ИТС-б-о-  
21-1

Тимофеева Марина Сергеевна

«\_\_»\_\_\_\_\_20\_\_г.

Подпись студента\_\_\_\_\_

Проверил: Доцент, к.т.н, доцент

кафедры      инфокоммуникаций

Воронкин Роман Александрович

---

(подпись)

Ставрополь, 2022

**Цель работы:** приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий - <https://github.com/AllsaL1sa/primer1.3><https://github.com/AllsaL1sa/primer1.6>

**Ход работы:**

### Задание

Для своего варианта лабораторной работы 2.8 необходимо дополнительно реализовать сохранение и чтение данных из файла формата JSON. *Необходимо также проследить за тем, чтобы файлы генерируемый этой программой не попадали в репозиторий лабораторной работы.*

```
elif command.startswith('load '):
    # Разбить команду на части для выделения имени файла.
    parts = command.split(' ', maxsplit=1)

    with open(parts[1], 'r', encoding="utf-8") as f:
        list_shop = json.load(f)

elif command.startswith('save '):
    # Разбить команду на части для выделения имени файла.
    parts = command.split(' ', maxsplit=1)

    with open(parts[1], 'w', encoding="utf-8") as f:
        json.dump(list_shop, f)

elif command == 'help':
    print('Список команд:\n')
    print('add - добавить магазин.')
    print('list - вывести список магазинов.')
    print('product <Название> - запросить информацию о товаре.')
    print("load <имя файла> - загрузить данные из файла.")
    print("save <имя файла> - сохранить данные в файл.")
    print('help - Справочник.')
    print('exit - Завершить работу программы.')
else:
    print(f'Команда <{command}> не существует.', file=sys.stderr)
```

Рисунок 1. Измененная часть кода лаб.раб. 2.8

```
>>> add
Название магазина: fvfv
Название товара: dghhbatghb
Стоимость товара: 4566
>>> load sys.json
>>> list
```

№	Название магазина	Название товара	Стоимость
1	Марина	яблоко	69

Рисунок 2. Результат выполнения программой

```
1 [{"name_shop": "Марина", "name_prodyckt": "яблоко", "prise": "69"}]
```

Рисунок 3. Файл sys.json

## Задание повышенной сложности

Очевидно, что программа в примере 1 и в индивидуальном задании никак не проверяет правильность загружаемых данных формата JSON. В следствие чего, необходимо после загрузки из файла JSON выполнять валидацию загруженных данных. Валидацию данных необходимо производить с использованием спецификации JSON Schema, описанной на сайте <https://json-schema.org/>. Одним из возможных вариантов работы с JSON Schema является использование пакета `jsonschema`, который не является частью стандартной библиотеки Python. Таким образом, необходимо реализовать валидацию загруженных данных с помощью спецификации JSON Schema.

```
elif command.startswith('load '):
    parts = command.split(' ', maxsplit=1)

    with open(parts[1], 'r', encoding="utf-8") as f:
        list_shop = json.load(f)
        validate(instance=list_shop, schema=schema)
```

Рисунок 4. Измененная часть кода и результат выполнения программой второго задания

## Контрольные вопросы:

1. Для чего используется JSON?

JSON представляет собой хорошую альтернативу XML и требует куда меньше форматирования контента. Это информативное руководство поможет вам быстрее разобраться с данными, которые вы можете использовать с JSON и основной структурой с синтаксисом этого же формата.

2. Какие типы значений используются в JSON?

Запись, массив, число, литералы, строка

3. Самостоятельно ознакомьтесь с форматом данных JSON5? В чем отличие этого формата от формата данных JSON?

JSON5 — предложенное расширение формата json в соответствии с синтаксисом ECMAScript 5, вызванное тем, что json используется не только для общения между программами, но и создаётся/редактируется вручную. Файл JSON5 всегда является корректным кодом ECMAScript 5. JSON5 обратно совместим с JSON

4. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

JSON5 расширяет формат обмена данными JSON, чтобы сделать его немного более удобным в качестве языка конфигурации:

- Комментарии в стиле JavaScript (как однострочные, так и многострочные) являются законными.
- Ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript
- Объекты и массивы могут заканчиваться запятыми.
- Строки могут заключаться в одинарные кавычки, и допускаются многострочные строковые литералы.

5. Какие средства предоставляет язык Python для сериализации данных в формате JSON?

Модуль `json` предоставляет удобный метод `dump()` для записи данных в файл. Существует также метод `dumps()` для записи данных в обычную строку. Типы данных Python кодируются в формат JSON в соответствии с интуитивно понятными правилами преобразования

6. В чем отличие функций `json.dump()` и `json.dumps()`?

`dump` отличается от `dumps` тем, что `dump` записывает объект Python в файл JSON, а `dumps` сериализует объект Python и хранит его в виде строки.

7. Какие средства предоставляет язык Python для десериализации данных из формата JSON?

В модуле `json` определены методы `load()` и `loads()`, предназначенные для преобразования кодированных в формате JSON данных в объекты Python. Подобно операции сериализации, также существует таблица преобразования типов, определяющая правила для обратного декодирования данных.

8. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кириллицу?

Параметр `ensure_ascii`

**Вывод:** приобрела навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.