

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №11

Работа с данными формата JSON в языке Python

По дисциплине «Теории программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

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

Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/18_laba.git.
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. Изменил пример из лабораторной работы 2.8, добавив возможность сохранения списка в файл формата JSON и чтения данных из файла JSON.
4. С помощью команды `save` сохранил таблицу в формате txt.

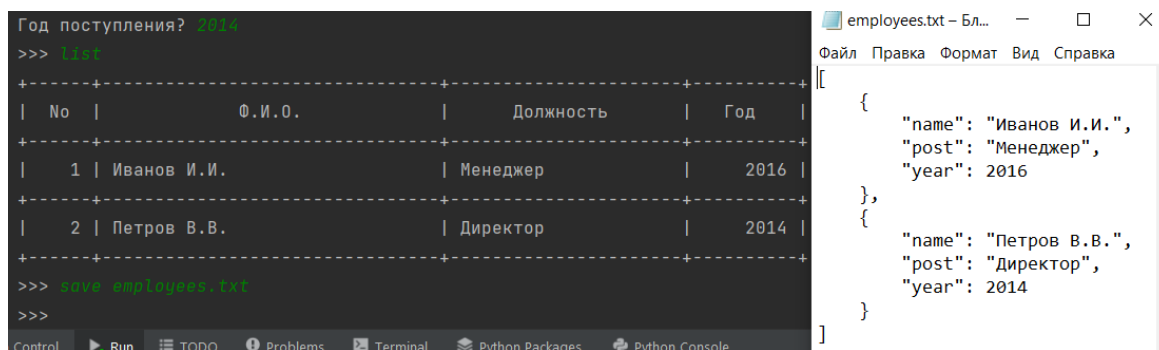


Рисунок 1. Результат сохранения файла через `save`

5. Затем завершил программу и загрузил таблицу через `load`.

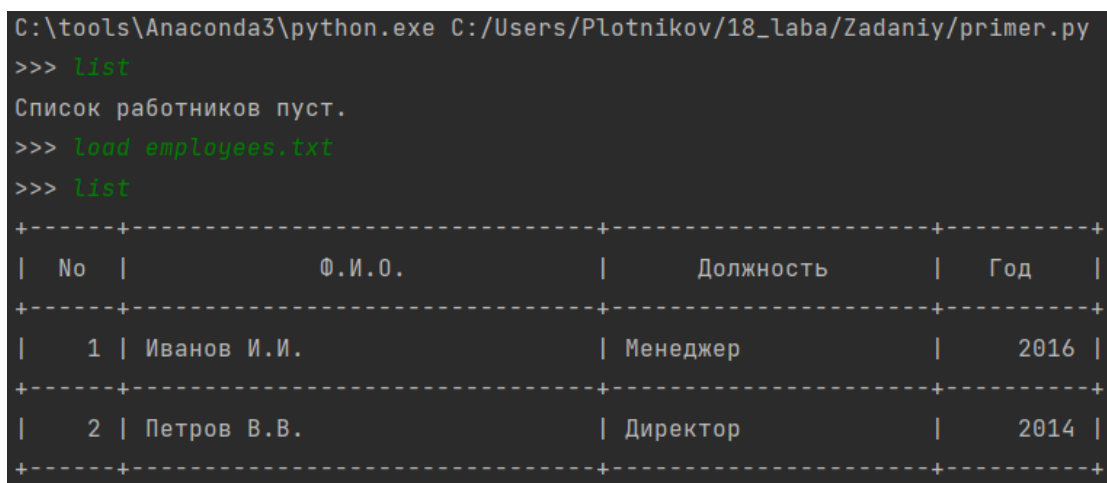


Рисунок 2. Результат загрузки содержимого файла через `load`

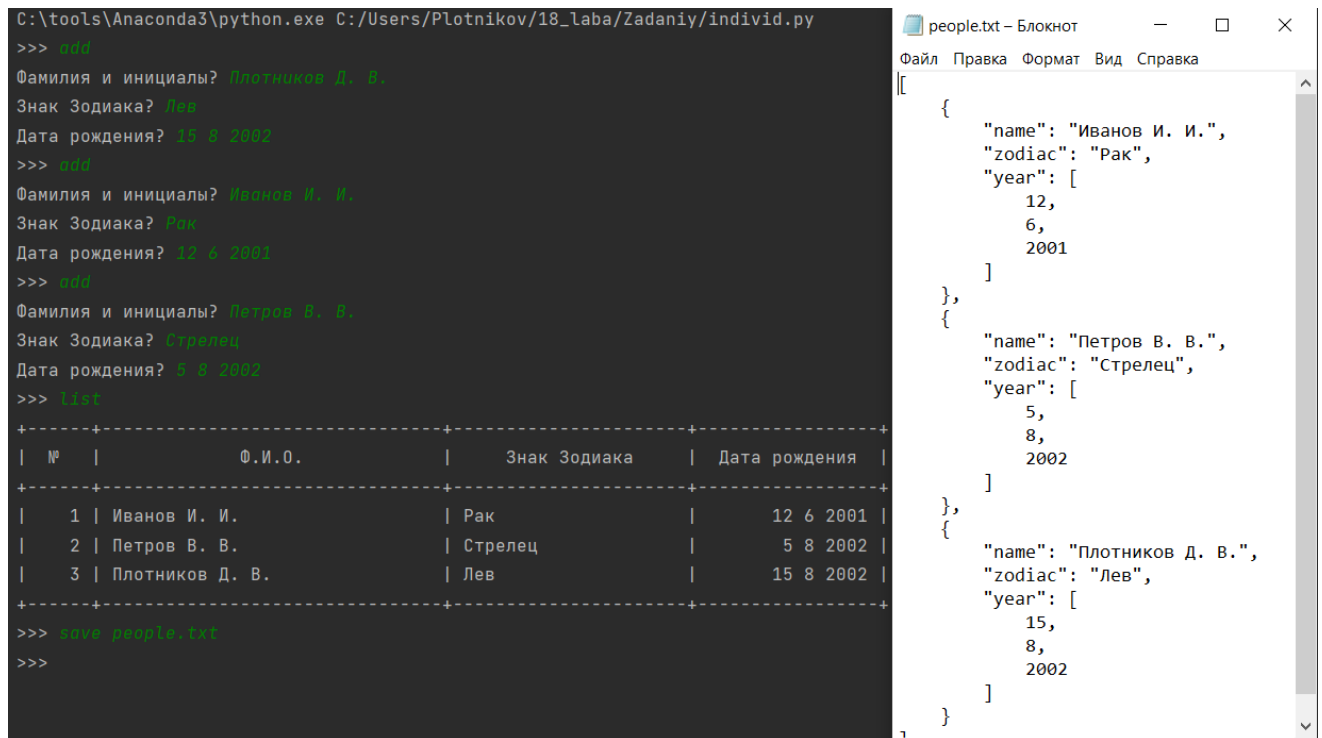
Индивидуальные задания

Вариант 13

Задание 1

1. Условие: 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.

2. Добавил 2 функции, отвечающие за сохранение и загрузку файлов и проверил работоспособность кода.

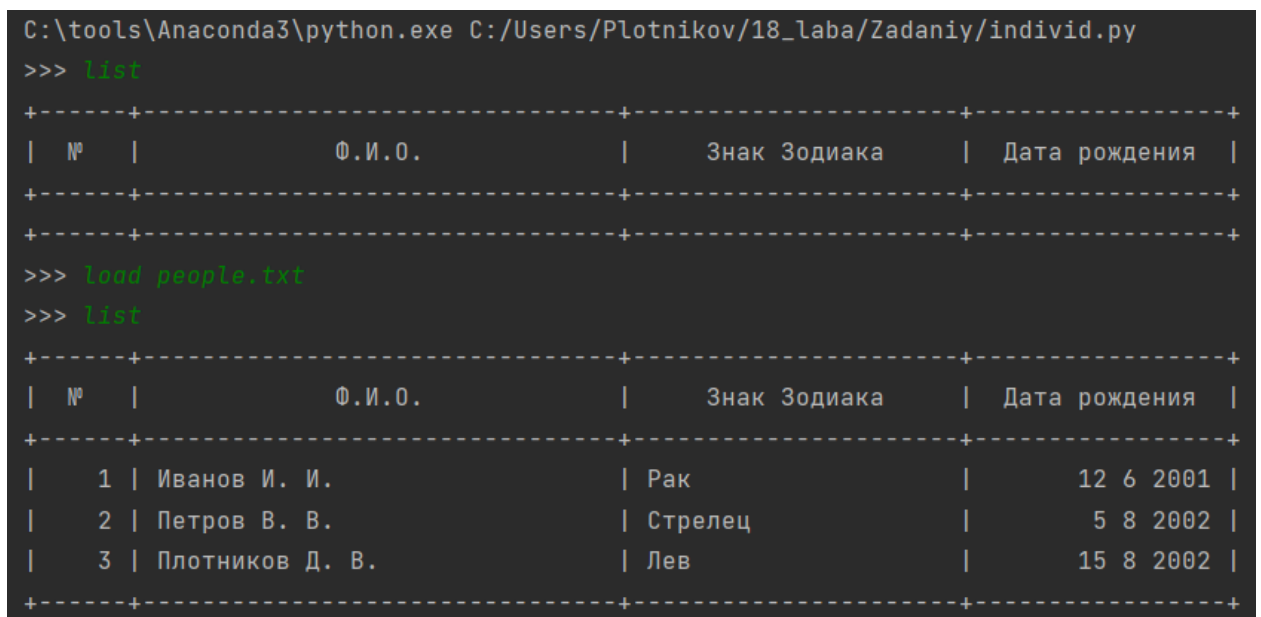


The screenshot shows a Python terminal window on the left and a text editor window titled 'people.txt - Блокнот' on the right. The terminal shows the execution of a script where three people are added and then listed. The text editor shows the JSON data saved to 'people.txt'.

```
C:\tools\Anaconda3\python.exe C:/Users/Plotnikov/18_laba/Zadaniy/individ.py
>>> add
Фамилия и инициалы? Плотников Д. В.
Знак Зодиака? Лев
Дата рождения? 15 8 2002
>>> add
Фамилия и инициалы? Иванов И. И.
Знак Зодиака? Рак
Дата рождения? 12 6 2001
>>> add
Фамилия и инициалы? Петров В. В.
Знак Зодиака? Стрелец
Дата рождения? 5 8 2002
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Иванов И. И. | Рак | 12 6 2001 |
| 2 | Петров В. В. | Стрелец | 5 8 2002 |
| 3 | Плотников Д. В. | Лев | 15 8 2002 |
+-----+-----+-----+-----+
>>> save people.txt
>>>
```

```
[
  {
    "name": "Иванов И. И.",
    "zodiac": "Рак",
    "year": [
      12,
      6,
      2001
    ]
  },
  {
    "name": "Петров В. В.",
    "zodiac": "Стрелец",
    "year": [
      5,
      8,
      2002
    ]
  },
  {
    "name": "Плотников Д. В.",
    "zodiac": "Лев",
    "year": [
      15,
      8,
      2002
    ]
  }
]
```

Рисунок 3. Результат сохранения файла через save



The screenshot shows a Python terminal window where the 'load' function is used to load data from 'people.txt' and then the 'list' function is used to display the data. The output is identical to the one in Figure 3.

```
C:\tools\Anaconda3\python.exe C:/Users/Plotnikov/18_laba/Zadaniy/individ.py
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Иванов И. И. | Рак | 12 6 2001 |
| 2 | Петров В. В. | Стрелец | 5 8 2002 |
| 3 | Плотников Д. В. | Лев | 15 8 2002 |
+-----+-----+-----+-----+
>>> load people.txt
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Иванов И. И. | Рак | 12 6 2001 |
| 2 | Петров В. В. | Стрелец | 5 8 2002 |
| 3 | Плотников Д. В. | Лев | 15 8 2002 |
+-----+-----+-----+-----+
```

Рисунок 4. Результат загрузки содержимого файла через load

Задание 2

1. Условие: выполнить валидацию данных при загрузке файла с помощью библиотеки jsonschema.

2. В терминале с помощью команды \$ pip install jsonschema установил библиотеку jsonschema и импортировал ее.

3. Внес данные в таблицу и сохранил файл, затем перешел в данный файл и изменил его структуру, поменяв «name» на «ame». И проверил работоспособность валидатора, загрузив файл.

```
C:\tools\Anaconda3\python.exe C:/Users/Plotnikov/18_laba/Zadaniy/individ2.py
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Знак Зодиака      |  Дата рождения  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
>>> load people.txt
Ошибка валидации

Process finished with exit code 1
```

Рисунок 5. Результат обнаружения ошибки

4. Далее вернул изначальную структуру в таблице и проверил заново.

```
C:\tools\Anaconda3\python.exe C:/Users/Plotnikov/18_laba/Zadaniy/individ2.py
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Знак Зодиака      |  Дата рождения  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
>>> load people.txt
Ошибок не обнаружено
```

Рисунок 6. Результат сообщения о правильности загрузки файла

Контрольные вопросы:

1. Для чего используется JSON?

– JSON используется для обмена данными, которые являются структурированными и хранятся в файле или в строке кода.

2. Какие типы значений используются в JSON?

- string;
- number;
- object;

- array;
- boolean;
- null.

3. Как организована работа со сложными данными в JSON?

– Данные также могут быть вложены в формате JSON, используя JavaScript массивы, которые передаются как значения. При помощи вложенных массивов и объектов можно создать сложную иерархию данных.

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

Формат обмена данными JSON5 (JSON5) — это надмножество JSON, которое направлено на смягчение некоторых ограничений JSON путем расширения его синтаксиса для включения некоторых продуктов из ECMAScript 5.1.

JSON5 получил следующие новшества:

- строки могут охватывать несколько строк, экранируя новые символы строк;
- числа могут быть шестнадцатеричными;
- допускаются однострочные и многострочные комментарии;
- ключи объектов могут быть без кавычек, если они являются законными идентификаторами ECMAScript;
- объекты и массивы могут заканчиваться запятыми в конце.

Существует одно заметное отличие от JSON: методы `load()` и `loads()`

поддерживают выборочную проверку (и отклонение) дубликатов ключейобъектов.

5. Какие средства языка программирования Python могут бытьиспользованы для работы с данными в формате JSON5?

- `json5.load()`;
- `json5.loads()`;
- `json5.tool()`;
- `json5.dump()`;

- `json5.dumps()`.

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

- Процесс кодирования данных в необходимый формат называется сериализацией. Для того чтобы записать эти данные в файл с форматом JSON в Python, используются функция `dump()` и `dumps()`.

7. 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.

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

- Когда есть файл JSON, который необходимо преобразовать в объект Python, тогда проводится десериализация. Для десериализации по аналогии используются две функции: `load()` и `loads()`.

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

- При записи достаточно передать `ensure_ascii=False`, чтобы не экранировать не-ascii символы.

10. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных? Приведите схему данных для примера 1.

Схема JSON – это словарь, который позволяет аннотировать и проверять документы JSON.

Преимущества:

- описывает ваш существующий формат(ы) данных;
- обеспечивает четкую читаемую документацию для человека и машины;
- проверяет данные, которые полезны для автоматизированного тестирования и обеспечения качества предоставляемых клиентом данных.

Пример схемы:

```
Schema = {  
    "type": "object", "employees": {  
        "name": {"type": "string"},  
        "post": {"type": "string"},  
        "year": {"type": "string", "format": "date"}  
    }  
}
```

Вывод: в ходе выполнения лабораторной работы успешно приобрел навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.