

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 «Технологии программирования и алгоритмизация»**

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р. А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

**Ход работы:**

**Ссылка на репозиторий:** <https://github.com/DIMITRY-GALYAS1/Laba-2.16.git>

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Проработал пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import json
import sys
from datetime import date

def get_worker():
    """
    Запросить данные о работнике.
    """
    name = input("Фамилия и инициалы? ")
    post = input("Должность? ")
    year = int(input("Год поступления? "))

    # Создать словарь.
    return {
        'name': name,
        'post': post,
        'year': year,
    }

def display_workers(staff):
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if staff:
        # Заголовок таблицы
```

Рисунок 1. Код примера

```
>>> list
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1  | Некифоров А.А.   | Старший специалист  |      2017     |
|  2  | Сидоров Н.Ф.     | Руководитель        |      2014     |
+-----+-----+-----+-----+
```

Рисунок 2. Введенные данные

```
[
    {
        "name": "Некифоров А.А.",
        "post": "Старший специалист",
        "year": 2017
    },
    {
        "name": "Сидоров Н.Ф.",
        "post": "Руководитель",
        "year": 2014
    }
]
```

Рисунок 3. Результат сохранения данных в файл

3. Выполнил индивидуальное задание.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import json
import sys
import jsonschema

def get_student():
    """
    Запросить данные о студенте.
    """
    name = input("Фамилия и инициалы? ")
    number = input("Номер группы? ")
    z = input("Успеваемость: ")

    # Создать словарь.
    return {
        'name': name,
        'number': number,
        'z': z,
    }

def display_students(students):
    """
    Отобразить список студентов.
    """
    # Проверить, что список студентов не пуст.
    if students:
        # Заголовок таблицы
```

Рисунок 4. Код индивидуального задания

```

{
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "number": {
          "type": "string"
        },
        "z": {
          "type": "string"
        }
      },
      "required": [
        "name",
        "number",
        "z"
      ],
      "additionalProperties": false
    }
  ]
}

```

Рисунок 5. Схема

```

[
  {
    "name": "Некифоров Н.С.",
    "number": "1",
    "z": "2 3 3 3 3"
  },
  {
    "name": "Чалов Г.В.",
    "number": "2",
    "z": "3 4 5 5 5"
  }
]

```

Рисунок 6. Результат сохранения данных в файл

```

>>> add
Фамилия и инициалы? Некифоров Н.С.
Номер группы? 1
Успеваемость: 2 3 3 3 3
>>> add
Фамилия и инициалы? Чалов Г.В.
Номер группы? 2
Успеваемость: 3 4 5 5 5
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          | Группа | Успеваемость |
+-----+-----+-----+-----+
|  1 | Некифоров Н.С.          |  1     | 2 3 3 3 3 |
|  2 | Чалов Г.В.              |  2     | 3 4 5 5 5 |
+-----+-----+-----+-----+
>>> save data.json
>>> load data.json
Файл успешно загружен
Валидация выполнена успешно
>>>

```

Рисунок 7. Загрузка файла и его валидация

### 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 ключей объектов.

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

- `json.load()`
- `json.loads();`
- `json.tool();`
- `json.dump();`
- `json.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 приобрел навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.