

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.16

Дисциплина: «Программирование на Python»

Тема: «Работа с данными формата JSON в языке Python»

Вариант 8

Выполнила: студентка

2 курса, группы ИВТ-б-о-21-1

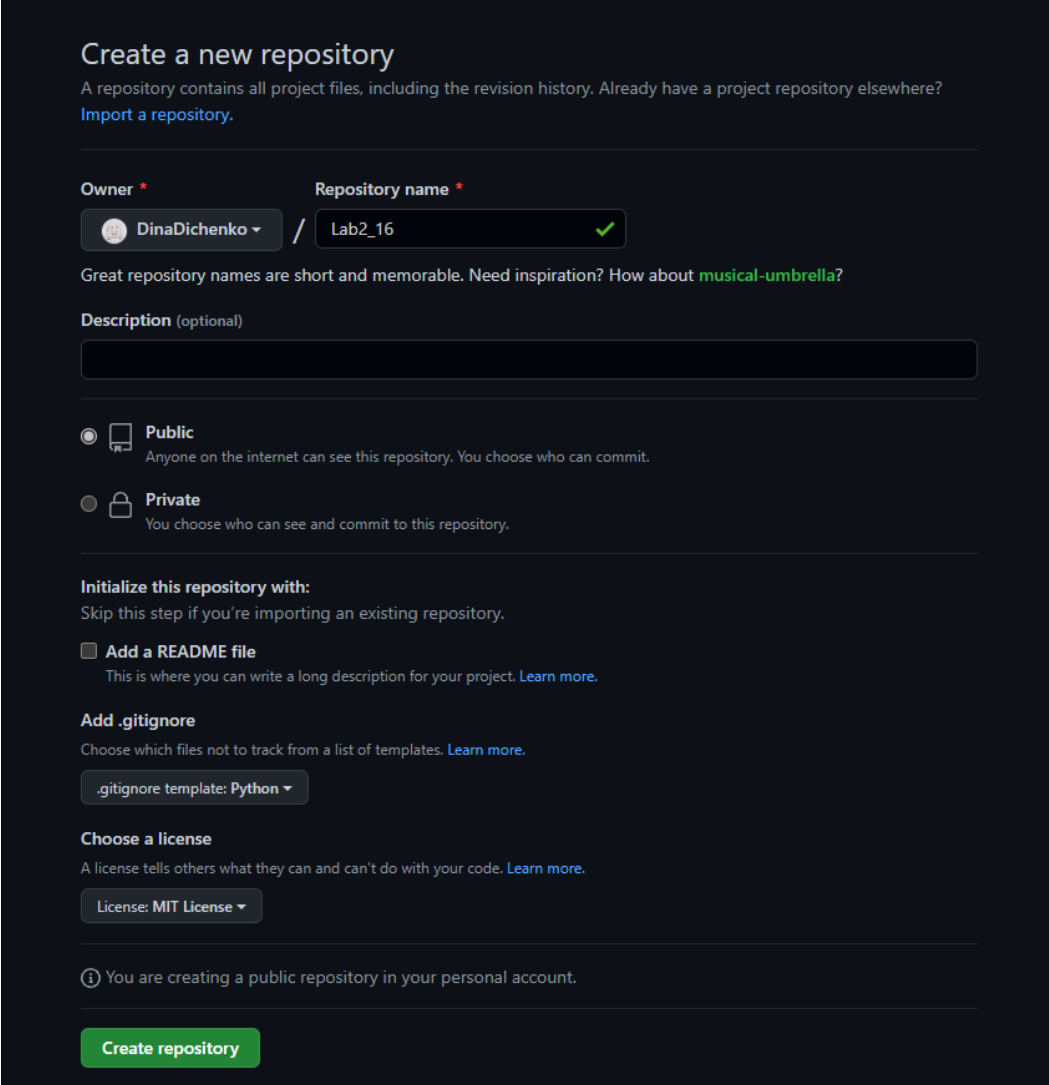
Диченко Дина Алексеевна

Ставрополь 2022

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

Практическая часть:


1. Создала репозиторий.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * **Repository name ***

 DinaDichenko / Lab2_16 ✓

Great repository names are short and memorable. Need inspiration? How about [musical-umbrella?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

 You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировала репозиторий.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>git clone https://github.com/DinaDichenko/Lab2_16.git
Cloning into 'Lab2_16'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>
```

Рисунок 2. Клонирование репозитория

3. Изменила файл .gitignore.

```
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
.idea/
.idea

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/dataSources.xml
```

Рисунок 3. Изменение файла .gitignore

4. Организовала репозиторий в соответствии с моделью ветвления git-glow.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16>git commit -m "gitignore"
[main 8056132] gitignore
1 file changed, 155 insertions(+), 3 deletions(-)

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16>git branch develop

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16>git checkout develop
Switched to branch 'develop'

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16>git flow init

Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] main

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/super/Desktop/Dina/ВУЗ/Программирование на python/Lab2_16/.git/hooks] hoo

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16>_
```

Рисунок 4. Организация репозитория в соответствии git-glow

5. Проработала примеры лабораторной работы.

```
>> save
Неизвестная команда save
>> save primer
>> save primer.json
>> exit
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_16\prog> c:; cd 'c:\Users\super\Python310\python.exe' 'c:\Users\super\.vscode\extensions\ms-python.python-2022.18.2\pythonFile
ние на python\Lab2_16\prog\primer1.py'
>> list
Список работников пуст.
>> load primer.json
>> list
```

No	Ф.И.О.	Должность	Год
1	Petr D R	Don	2021

```
>> |
```

Рисунок 5. Результат работы программы

6. 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.

```
+-----+
| No | Название | Время |
+-----+
>>> load ind.json
>>> lis
Неизвестная команда lis
>>> list
```

No	Название	Время
1	f	12:12:30

Рисунок 6. Результат работы программы

7. 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.

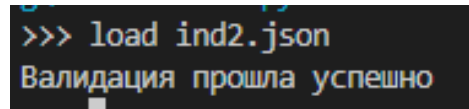
A screenshot of a terminal window with a dark background. It shows a command prompt where the command `>>> load ind2.json` has been entered. Below the command, the text "Валидация прошла успешно" (Validation passed successfully) is displayed in a light blue color.

Рисунок 7. Результат работы программы

Контрольные вопросы:

1. Для чего используется JSON?

JSON (англ. JavaScript Object Notation, обычно произносится как /'dʒeɪsən/ JAY-sən) – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Формат JSON был разработан Дугласом Крокфордом.

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

За счёт своей лаконичности по сравнению с XML формат JSON может быть более подходящим для сериализации сложных структур. Применяется в веб-приложениях как для обмена данными между браузером и сервером (AJAX), так и между серверами (программные HTTP-сопряжения).

Легкочитаемый и компактный, 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?

В качестве значений в JSON могут быть использованы:

запись — это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми.

массив (одномерный) — это упорядоченное множество значений. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми. Массив может быть пустым, т.е. не содержать ни одного значения. Значения в пределах одного массива могут иметь разный тип.

число (целое или вещественное).

литералы true (логическое значение «истина»), false (логическое значение «ложь») и null.

строка — это упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» (поддерживаются варианты ' ', \" \\", \t, \n, \r, \f и \b), или записаны шестнадцатеричным кодом в кодировке Unicode в виде \uFFFF.

3. Как организована работа со сложными данными в JSON?

JSON может содержать другие вложенные объекты в JSON, в дополнение к вложенным массивам.

Такие объекты и массивы будут передаваться, как значения назначенные ключам и будут представлять собой связку ключ-значение.

Использование вложенности в нашем JSON формате позволяет нам работать с наиболее сложными и иерархичными данными.

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

JSON5 — это расширение популярного формата файлов JSON, которое упрощает написание и поддержку вручную (например, для файлов конфигурации). Он не предназначен для межмашинного взаимодействия.

Формально формат обмена данными JSON5 является расширенным набором JSON (поэтому допустимые файлы JSON всегда будут действительными файлами JSON5), который расширяет свой синтаксис, чтобы включить некоторые продукты из ECMAScript 5.1 (ES5). Это также строгое подмножество ES5, поэтому действительные файлы JSON5 всегда будут действительными ES5.

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

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

`json.dump()` - конвертировать python объект в json и записать в файл.

`json.dumps()` - тоже самое, но в строку.

Обе эти функции принимают следующие необязательные аргументы:

Если `skipkeys = True`, то ключи словаря не базового типа (`str`, `int`, `float`, `bool`, `None`) будут проигнорированы, вместо того, чтобы вызывать исключение `TypeError`.

Если `ensure_ascii = True`, все не-ASCII символы в выводе будут экранированы последовательностями `\uXXXX`, и результатом будет строка, содержащая только ASCII символы. Если `ensure_ascii = False`, строки запишутся как есть.

Если `check_circular = False`, то проверка циклических ссылок будет пропущена, а такие ссылки будут вызывать `OverflowError`.

Если `allow_nan = False`, при попытке сериализовать значение с запятой, выходящее за допустимые пределы, будет вызываться `ValueError (nan, inf, -inf)` в строгом соответствии со спецификацией JSON, вместо того, чтобы использовать эквиваленты из JavaScript (`NaN`, `Infinity`, `-Infinity`).

Если `indent` является неотрицательным числом, то массивы и объекты в JSON будут выводиться с этим уровнем отступа. Если уровень отступа 0, отрицательный или `""`, то вместо этого будут просто использоваться новые

строки. Значение по умолчанию None отражает наиболее компактное представление. Если indent - строка, то она и будет использоваться в качестве отступа.

Если sort_keys = True , то ключи выводимого словаря будут отсортированы.

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

json.dump() - конвертировать python объект в json и записать в файл.

json.dumps() - тоже самое, но в строку.

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

json.load() # прочитать json из файла и конвертировать в python объект.

json.loads() # тоже самое, но из строки с json (s на конце от string/строка).

Обе эти функции принимают следующие аргументы:

object_hook - опциональная функция, которая применяется к результату декодирования объекта (dict). Используется будет значение, возвращаемое этой функцией, а не полученный словарь.

object_pairs_hook - опциональная функция, которая применяется к результату декодирования объекта с определённой последовательностью пар ключ/значение. Будет использован результат, возвращаемый функцией, вместо исходного словаря. Если задан так же object_hook , то приоритет отдаётся object_pairs_hook .

parse_float , если определён, будет вызван для каждого значения JSON с плавающей точкой. По умолчанию, это эквивалентно float(num_str) .

parse_int , если определён, будет вызван для строки JSON с числовым значением. По умолчанию эквивалентно int(num_str) .

`parse_constant` , если определён, будет вызван для следующих строк: `"-Infinity"`, `"Infinity"`, `"NaN"`. Может быть использовано для возбуждения исключений при обнаружении ошибочных чисел JSON.

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

```
json.dump(staff, fout, ensure_ascii=False, indent=4).
```

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

JSON Schema — один из языков описания структуры JSON-документа. Использует синтаксис JSON. Базируется на концепциях XML Schema, RelaxNG, Kwalify. JSON Schema — самоописательный язык: при его использовании для обработки данных и описания их допустимости могут использоваться одни и те же инструменты сериализации / десериализации.

Схема данных - - модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об изменениях входных величин моделировать возможные состояния объекта.

Вывод: в результате выполнения работы были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.