**Лабораторная работа №1**

**Цель работы**: Познакомиться с понятием набора данных, их источниками и возможными представлениями (форматом).

**Задачи**:

1. Разобраться с содержанием понятия «набор данных» и целью их использования.
2. Найти в интернете источники данных, в том числе российские.
3. Скачать и проанализировать состав одного из найденных и заинтересовавших Вас набора данных.

<https://all-python.ru/osnovy/csv.html>

<https://blog.skillfactory.ru/glossary/csv-format/> !!!!

<https://www.computerhope.com/issues/ch001356.htm>

Оглавление

[1 О формате CSV 2](#_Toc160710230)

[1.1 Определение формата CSV 2](#_Toc160710231)

[1.2 Простые примеры файлов в CSV формате 2](#_Toc160710232)

[1.3 Создание CSV-файла блокноте и перенос данных в Excel 3](#_Toc160710233)

[1.4 Детали использования таблиц формата csv 4](#_Toc160710234)

[1.5 Общие правила форматирования данных в CSV файле 6](#_Toc160710235)

[2 О формате JSON и его использовании 7](#_Toc160710236)

[2.1 Как появился JSON 8](#_Toc160710237)

[2.2 Преимущества JSON 8](#_Toc160710238)

[2.3 Структура и синтаксис JSON 8](#_Toc160710239)

[2.4 Как работать с JSON 9](#_Toc160710240)

[2.5 Вспомним всё, что мы узнали про JSON 10](#_Toc160710241)

[2.7 Преобразование документа Word в формат json 10](#_Toc160710242)

[3 Конвертация файлов 10](#_Toc160710243)

[3.1 Запись данных формата Json в Excel 10](#_Toc160710244)

[3.2 Loginom и Json 11](#_Toc160710245)

[3.3 Преобразование документа Word в формат json с использованием Python 11](#_Toc160710246)

[3.4 Конвертируйте текстовый файл в CSV с помощью **Python Pandas** 11](#_Toc160710247)

[4 Задание к практическим занятиям 12](#_Toc160710248)

[4.1 Знакомство с системами Anaconda, Jupyter Notebook и Pandas. 12](#_Toc160710249)

[4.2 Знакомство с Нижегородстатом 12](#_Toc160710250)

[4.3 Реализовать с помощью Pandas (по материалам лекции) преобразование произвольного текстового файла в формат CSV, а затем в Excel (https://www.geeksforgeeks.org/convert-csv-to-excel-using-pandas-in-python/). 13](#_Toc160710251)

# 1 О формате CSV

## 1.1 Определение формата CSV

Файл CSV – **это особый вид файла, который позволяет структурировать большие объемы данных**.

По сути, он является обычным текстовым файлом, однако **каждый новый элемент отделен от предыдущего запятой или другим разделителем**. **Обычно каждая запись начинается с новой строки**. Данные CSV можно легко экспортировать в электронные таблицы или базы данных. Программист может расширять CSV файл, добавляя новые строки.

CSV (значения, разделенные запятыми) - это простой формат файла, в котором хранятся табличные данные, такие как электронная таблица или база данных.

Файлы в формате CSV можно импортировать в программы, которые хранят данные в таблицах, такие как Microsoft Excel или OpenOffice Calc, и экспортировать из них.

**CSV расшифровывается как "comma (запятая)-** **separate (отдельные) (values) значения". Поля данных этого формата разделяются запятой.**

CSV (Comma-Separated Values) — **это текстовый формат для представления табличных данных**. Строка таблицы соответствует строке текста, которая содержит поля, разделенные запятыми. **Тип файлов предназначен для передачи объемных текстовых данных между различными программами и сервисами**.

## 1.2 Простые примеры файлов в CSV формате

**Пример CSV файла, где в качестве разделителя используется запятая:**

Имя,Профессия,Год рождения

Виктор,Токарь, 1995

Сергей,Сварщик,1983

**Как видно из примера, в строке обычно указывается, какая информация будет находиться в каждом столбце**.

**Кроме того, после последнего элемента строки запятая не ставиться**. Обработчик текста (интерпретатор) определяет конец строки по символу переноса.

Важно помнить, что CSV – это обычный текстовый файл, который **не поддерживает** символы в кодировках, отличающихся от ASCII или Unicode.

Таблица 1



Приведенные в таблице 1 данные могут быть представлены в файле в формате CSV следующим образом (рисунок 1):

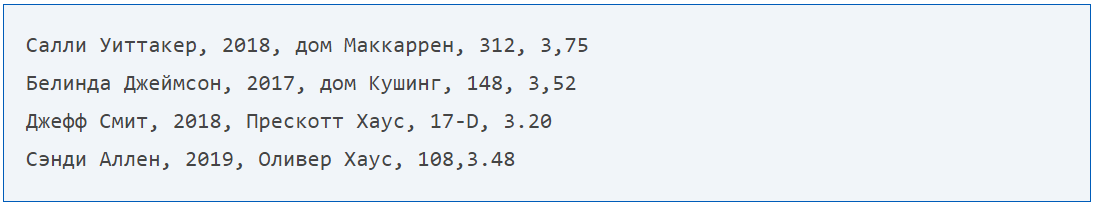


Рисунок 1. Представление данных из таблицы 1 в текстовом файле.

## 1.3 Создание CSV-файла блокноте и перенос данных в Excel

1 Чтобы создать CSV-файл с помощью текстового редактора, сначала выберите свой любимый текстовый редактор, такой как Notepad или другой, и откройте новый файл. Затем введите текстовые данные, которые вы хотите, чтобы файл содержал, разделяя каждое значение запятой, а каждую строку новой строкой (рисунок 2).

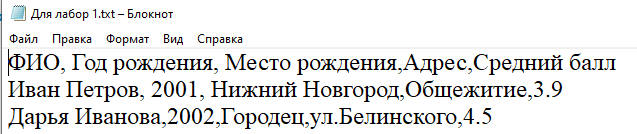


Рисунок 2

2. Сохраните этот файл с расширением .csv выбрав для русского языка кодировку ANSI. Затем вы можете открыть файл с помощью Microsoft Excel или другой программы для работы с электронными таблицами. Это создало бы таблицу данных, подобную следующей (рисунок3):

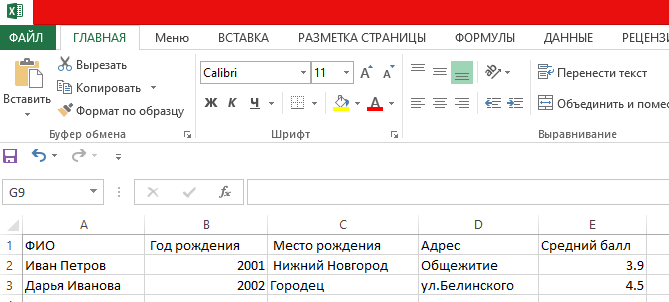


Рисунок 3

Открытие csv файла на русском языке в Calc требует кодировки «кириллица(windows-1251), как показано на рисунке 4:

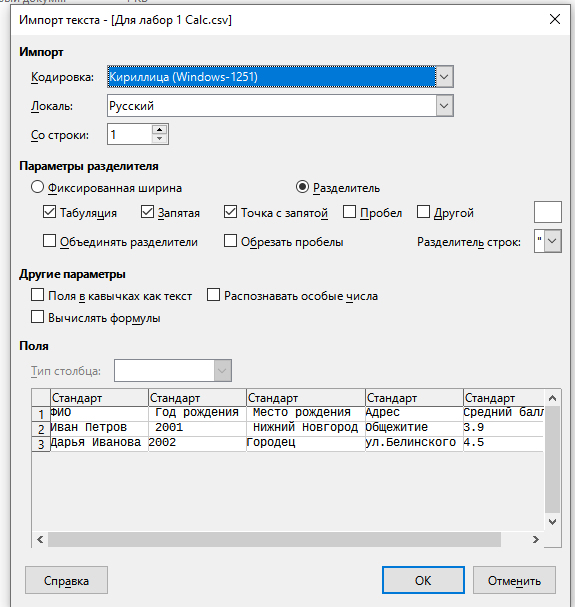


Рисунок 4.

## 1.4 Детали использования таблиц формата csv

**В созданном вами файле CSV отдельные поля данных были разделены запятыми. Но что, если сами данные содержат запятые?**

Если поля данных в вашем CSV файле содержат запятые, вы можете защитить их, заключив эти поля данных в двойные кавычки ("). Запятые, которые являются частью ваших данных, хранятся отдельно от запятых, которые разделяют сами поля.

Например, предположим, что одно из наших текстовых полей представляет собой созданное пользователем описание, которое допускает **использование запятых в описании**. Если бы наши данные выглядели следующим образом:



Рисунок 5. Результат открытия csv файла в Excel

Чтобы сохранить запятые в нашем столбце "Примечания", мы можем заключить эти поля в кавычки. Например:

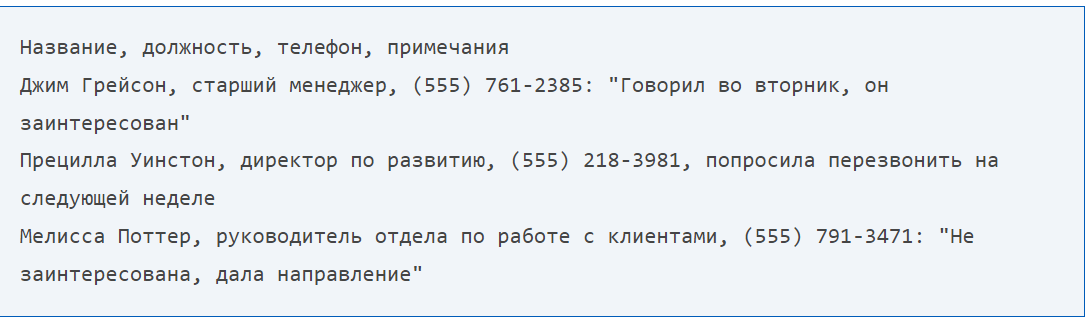


Рисунок 6.

**Как вы можете видеть, в кавычки заключены только поля, содержащие запятые**.

То же самое касается новых строк, которые могут быть частью данных вашего поля. Любые поля, содержащие новую строку как часть своих данных, должны быть заключены в двойные кавычки.

Если ваши поля содержат двойные кавычки как часть своих данных, внутренние кавычки необходимо удвоить, чтобы их можно было правильно интерпретировать. Например, учитывая следующие данные (рисунок 7):

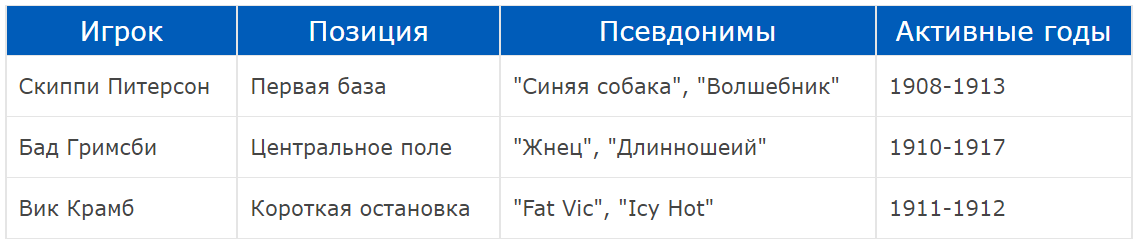


Рисунок 7.

Мы можем представить его в CSV-файле следующим образом:

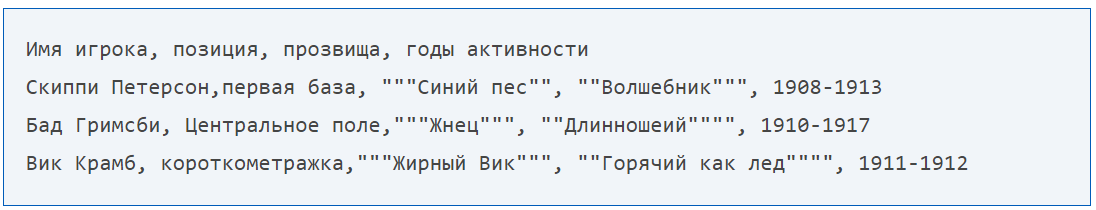


Рисунок 8

Здесь все поле данных заключено в кавычки, а внутренним кавычкам предшествует (экранируется) дополнительная двойная кавычка.

## 1.5 Общие правила форматирования данных в CSV файле

* Каждая строка в документе олицетворяет собой строку таблицы. Разделение между строками обозначается с использованием символов возврата каретки (CR, LF, CRLF или других в зависимости от кодировки файла).
* Все строки таблицы обязаны содержать одинаковое количество полей. При этом отмечается, что поле может содержать сколько угодно символов, включая пробелы.
* В первой строке (на усмотрение автора) прописываются имена столбцов. Следует убедиться, что число полей в первой строке соответствует числу полей в остальной таблице.
* Если поле не содержит данные, допускается обозначить его просто разделительными запятыми, без пробелов между ними.
* Пробелы перед и после запятой-разделителя не отображаются в таблице и будут опущены.
* Последнее поле документа не должно завершаться запятой. Здесь также можно выбрать, использовать или не использовать символ возврата каретки (переноса строки).
* Любые данные внутри одной ячейки могут быть окружены двойными кавычками. Если же кавычки применяются не ко всему полю, вставлять их в текст внутри этого поля не рекомендуется.
* При наличии в тексте ячейки двойных кавычек, запятых и символов возврата каретки, содержимое всей ячейки следует заключить в двойные кавычки.
* Когда двойные кавычки присутствуют внутри текста ячейки, их следует экранировать при помощи еще одной пары двойных кавычек. Например, «начало »«выделенной фразы»« конец».
* Некоторые авторы используют символ «/» для экранирования кавычек, однако это не является правильным подходом и может привести к некорректному отображению в большинстве программ.
* Хотя формат CSV не определяет типы данных и не предоставляет средства для их обозначения, программа, в которой открывается файл, может интерпретировать их. Например, строковые данные, состоящие из цифр, могут быть восприняты как числа программой [Microsoft Excel](https://blog.skillfactory.ru/glossary/excel/), что может повлиять на округление или удаление начальных нулей. Чтобы этого избежать, в документах, которые будут использоваться в Excel, перед числами, заключенными в кавычки, можно добавить знак «=».
* Некоторые приложения могут не поддерживать форматирование с использованием двойных кавычек. Это следует учитывать при подготовке и экспорте документов.
* При записи дробных чисел в формате CSV важно помнить, что разделитель десятичной части может различаться в зависимости от региональных настроек: в одних странах используется точка, в других — запятая.
* Некоторые способы записи числовых данных могут быть восприняты программами как даты, валюты или формулы, что влияет на их тип и обработку. Эти аспекты также зависят от региональных настроек.

**Вывод:** *соблюдение правил форматирования в таблице CSV имеет решающее значение. Ошибки могут привести к тому, что таблица станет непонятной или даже нечитаемой. В результате может получиться бессвязный «клубок» текста без структуры и смысла*.

# 2 О формате JSON и его использовании

JSON (JavaScript Object Notation) — **стандартный текстовый формат для хранения и передачи структурированных данных**.

Он основан на синтаксисе объекта в JavaScript, но не привязан к нему. Работу с ним поддерживают многие современные языки программирования: Python, Java и другие.

JSON – объект JavaScript. Своеобразный метод хранения и передачи информации структурированного характера. За счет элементарного синтаксиса позволяет сохранять все, что угодно:

* строки;
* массивы;
* числа;
* те или иные объекты.

Перечисленные элементы будут представлены в текстовом формате. JSON является простейшим для понимания открытым стандартом. Соглашения из него известны не только тем программистам, которые на практике применяют JS. Другие языки тоже сталкиваются с рассматриваемым стандартом.

## 2.1 Как появился JSON

<https://skillbox.ru/media/code/json-chto-eto-za-format-i-kak-s-nim-rabotat/>

В 2014 году орган по стандартизации ECMA принял JSON и официально специфицировал его. Сегодня этот формат является основным для передачи данных при взаимодействии веб-сервера и браузера.

## 2.2 Преимущества JSON

JSON не единственный формат передачи данных. Некоторые разработчики используют XML или YAML. Но в сравнении с ними у JSON есть ряд преимуществ:

**Простой и легко читаемый синтаксис**, который понятен и компьютеру, и человеку. Необязательно даже разбираться в JavaScript, чтобы научиться записывать и читать данные. Скоро вы это увидите сами.

**Лёгкость и компактность**. JSON-файлы весят меньше, чем файлы других форматов, и загружаются быстрее. Согласно некоторым исследованиям, современные браузеры обрабатывают JSON в десятки раз быстрее XML.

**Отсутствие зависимости от конкретного языка** программирования. Для работы с JSON можно использовать почти любой популярный язык программирования, например, Python или Java.

**Универсальность в типах данных**. JSON можно использовать для хранения массивов, упорядоченных списков и коллекций пар «ключ — значение». Но и это не всё. Формат подходит для представления сложных структур данных, включая деревья и иерархические структуры.

**Широкая поддержка в браузерах**. Все современные браузеры поддерживают работу с JSON.

**Самостоятельное документирование**. В JSON можно включить метаданные, которые описывают представленные данные, упрощая работу с ними.

## 2.3 Структура и синтаксис JSON

Синтаксис JSON повторяет синтаксис обычного объекта в JavaScript: данные записываются в виде пар «ключ — значение» и разделяются запятыми. Важно запомнить, что после последней пары запятая не ставится — это распространённая ошибка, которую можно долго искать.

**Ключи** — строковые переменные, а **значени**я могут быть строками, числами, булевыми значениями, объектами, массивами или null. Функции и даты в обычном JSON не поддерживаются.

Вот как выглядит простой JSON-файл (рисунок 9):



Рисунок 9

## 2.4 Как работать с JSON

Умение работать с JSON-файлами — один из базовых навыков для разработчиков, **поскольку именно этот формат является стандартом для обмена данными в веб- и мобильных приложениях**.

Как правило, работа с JSON сводится к двум действиям:

* сериализации и
* десериализации.

Поговорим о них подробнее.

**Сериализация** — это преобразование данных из внутреннего формата, например, из JavaScript-объекта, в формат JSON, который может быть отправлен на сервер. В большинстве языков программирования для этого есть встроенные функции или библиотеки. Например, в JavaScript преобразование объекта **person** в JSON-файл выглядит так (рисунок 10):



Рисунок 10

**Десериализация**, наоборот, преобразует данные **из формата JSON обратно во внутренний формат**, **с которым программист может работать в своём коде**:

const jsonString = '{"name":"John Doe","age":30,"address":{"city":"New York","zip":"10001"}}';

const person = JSON.parse(jsonString); // **Преобразуем JSON в объект person**

console.log(person); // Выводит данные в консоль

## 2.5 Вспомним всё, что мы узнали про JSON

JSON — **стандартный текстовый формат для хранения структурированных данных и обмена ими**. Чаще всего используется в вебе.

Данные в JSON представлены в виде пар «ключ — значение».

Для хранения данных в JSON используются две структуры — объекты и массивы. Объекты — неупорядоченные коллекция, массивы — упорядоченная.

JSON Schema используется для определения требований к данным: разработчик может указать обязательные поля и их допустимые форматы.

Для работы с JSON используется два основных подхода: сериализация, позволяющая преобразовать JavaScript-код или код другого языка программирования в JSON-формат, и десериализация, работающая наоборот: из JSON формата данные переводятся в обычный код.

<https://otus.ru/journal/vse-chto-nuzhno-znat-o-json-osobennosti-opisanie-pervye-proekt/>

<https://www.oracle.com/cis/database/what-is-json/>

## 2.7 Преобразование документа Word в формат json

<https://blog.aspose.com/ru/total/convert-word-to-json-in-python/>

# 3 Конвертация файлов

## 3.1 Запись данных формата Json в Excel

<https://www.oracle.com/cis/database/what-is-json/>

## 3.2 Loginom и Json

<https://examples.loginom.ru/json/>

JSON — текстовый формат обмена данными, основанный на JavaScript, который является основным форматом, используемым при передаче данных с помощью REST-запросов[[1]](#footnote-1),[[2]](#footnote-2). Данные, полученные в формате JSON, в исходном виде непригодны для дальнейшего анализа в Loginom. Но эту проблему можно решить с помощью компонента JavaScript.

Пример демонстрирует распаковку данных, полученных в формате JSON, с помощью Loginom. В качестве источника используется сервис <https://www.cbr-xml-daily.ru/> . Данные представлены курсами валют на заданную дату.

## 3.3 Преобразование документа Word в формат json с использованием Python

<https://blog.aspose.com/ru/total/convert-word-to-json-in-python/>

## 3.4 Конвертируйте текстовый файл в CSV с помощью **Python Pandas**

Давайте посмотрим, как преобразовать текстовый файл в CSV с помощью Python Pandas. Python считывает данные из текстового файла и создает фрейм данных со строками, равными количеству строк, присутствующих в текстовом файле, и столбцами, равными количеству полей, присутствующих в одной строке. Смотрите приведенный ниже пример для лучшего понимания.

Исходный текстовый файл в Notepad выглядит так (рисунок 11):

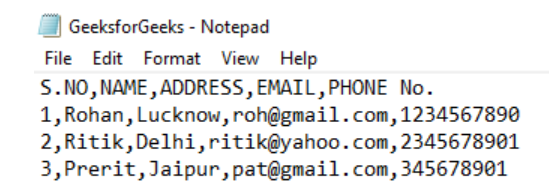


Рисунок 11. Исходный текстовый файл.

После обработки в Питоне получим так называемый фрейм данных (рисунок 12):

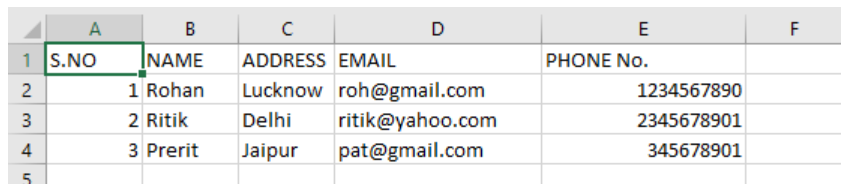


Рисунок 12. CSV-файл, сформированный из заданного текстового файла.

**Примечание**: Первый столбец в dataframe индексируется по умолчанию при чтении текстового файла.

Как только фрейм данных будет создан, мы сохраним этот фрейм данных в формате файла CSV, используя метод Dataframe.to\_csv() .

Примеры из <https://www.geeksforgeeks.org/convert-text-file-to-csv-using-python-pandas/> .

# 4 Задание к практическим занятиям

## 4.1 Знакомство с системами Anaconda, Jupyter Notebook и Pandas.

* Установить дистрибутив Anaconda.
* Установить пакет Jupyter Notebook.
* Установить пакет Pandas.
* Ознакомиться (по материалам Интернета) с системой обработки данных в системе Python Pandas. Использовать для этого следующие источники:
* <https://pythonist.ru/pandas-tutorial/>
* <https://www.geeksforgeeks.org/how-to-install-python-pandas-on-windows-and-linux/>
* <https://www.geeksforgeeks.org/how-to-use-jupyter-notebook-an-ultimate-guide/?ref=lbp>
* <https://practicum.yandex.ru/blog/chto-takoe-jupyter-notebook/>

В отчёте отметить назначение этих пакетов и их возможности

## 4.2 Знакомство с Нижегородстатом

* Войти на сайт территориального органа Федеральной службы государственной статистики по Нижегородской области (Нижегородстат) - <https://52.rosstat.gov.ru/about>.
* Ознакомиться с материалами и форматом их представления.

## 4.3 Реализовать с помощью Pandas (по материалам лекции) преобразование произвольного текстового файла в формат CSV, а затем в Excel (https://www.geeksforgeeks.org/convert-csv-to-excel-using-pandas-in-python/).

1. Rest — это обычный запрос клиент-сервер с использованием HTTP протокола. В роли клиента не обязательно выступает браузер, это может быть мобильное приложение, десктопное приложение или же другой веб-сайт. В качестве ответа сервер выдает не привычную html-страницу, а просто набор данных, оформленных в том или ином формате. Чаще всего это JSON или XML. Неразрывно с REST следует AJAX (Asynchronous Javascript and XML). [↑](#footnote-ref-1)
2. <https://tproger.ru/articles/osnovy-rest-teorija-i-praktika> [↑](#footnote-ref-2)