



Файловая система



План занятия

1

Формат файлов CSV

2

Формат файлов JSON

Формат CSV, JSON

- CSV (от англ. Comma-Separated Values — значения, разделённые запятыми) — текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделённых запятыми.
- JSON (JavaScript Object Notation) — текстовый формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка программирования JavaScript.

CSV и JSON — наиболее распространённые форматы, используемые для приема, хранения неструктурированных или частично структурированных данных и обмена ими.

CSV

- **CSV** — текстовый формат, ориентированный на работу с данными несложных электронных таблиц, хранящихся в обычных тестовых файлах.
- **Модуль csv** упрощает синтаксический анализ (парсинг) CSV-файлов.

Подключение

```
import csv
```

Чтение CSV-файлов

- Файл `example.csv`, из которого будем читать:

05.04.2015 13:34;Яблоки;73

05.04.2015 3:41;Вишни;85

06.04.2015 12:46;Груши;14

Чтение данных из файла CSV

Чтобы прочитать данные из CSV-файла, необходимо создать объект Reader, который обеспечивает возможность итерирования по строкам файла:

```
import csv

exampleFile=open('example.csv',encoding='UTF-8')
exampleReader=csv.reader(exampleFile,delimiter=';')
exampleData=list(exampleReader)
print(exampleData)
exampleFile.close()
```

Результат работы программы:

```
[['05.04.2015 13:34', 'Яблоки', '73'],  
 ['05.04.2015 3:41', 'Вишни', '85'],  
 ['06.04.2015 12:46', 'Груши', '14']]
```

Отдельные ячейки

Теперь, когда у нас есть список списков, можно обращаться к отдельным ячейкам с помощью **exampleData[row][col]**.

В случае больших CSV-файлов целесообразно использовать объект Reader в цикле for.

Тем самым удастся избежать загрузки всего файла в оперативную память.

```
import csv

exampleFile = open('example.csv', encoding='UTF-8')
exampleReader = csv.reader(exampleFile, delimiter=';')
exampleData = list(exampleReader)

for row in exampleReader:
    string = 'Строка #' + str(exampleReader.line_num) + ' '
    for value in row:
        string = string + value + ' '
    print(string)

exampleFile.close()
```

Результат работы программы:

```
Строка #1 05.04.2015 13:34 Яблоки 73
Строка #2 05.04.2015 3:41 Вишни 85
Строка #3 06.04.2015 12:46 Груши 14
```

Для вывода номера строки используется свойство `line_num` объекта `Reader`. Цикл по объекту `Reader` можно выполнить только один раз. Для повторного чтения данных необходимо заново создать объект `Reader`, вызвав метод `csv.reader()`.

Запись CSV-файлов

```
import csv
exampleFile = open('output.csv', 'w', encoding='UTF-8', newline='')
exampleWriter = csv.writer(exampleFile, delimiter=';')
exampleData = [['05.04.2015 13:34', 'Яблоки', '73'], ['05.04.2015 3:41', 'Вишни', '85'], ['06.04.2015 12:46', 'Груши', '14']]
for row in exampleData:
    exampleWriter.writerow(row)
exampleFile.close()
```

Файл output.csv:

```
05.04.2015 13:34;Яблоки;73
05.04.2015 3:41;Вишни;85
06.04.2015 12:46;Груши;14
```

При работе в Windows функции `open()` надо передать именованный аргумент `newline = ''`, иначе в файле `output.csv` появятся лишние пустые строки. С помощью именованного аргумента `lineterminator` метода `csv.writer()` можно заменить разделитель строк.

JSON (JavaScript Object Notation) — текстовый формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка программирования JavaScript.

JSON основан на двух структурах данных:

- Коллекция пар ключ-значение (объект)
- Упорядоченный список значений (массив)

Подключение

```
import json
```

JSON в сравнении с CSV-файлами

Файлы в формате JSON имеют несколько преимуществ по сравнению с CSV-файлами.

- JSON поддерживает иерархические структуры, упрощая хранение связанных данных в одном документе и представление сложных связей.
- Большинство языков предоставляют упрощенные библиотеки сериализации JSON или встроенную поддержку десериализации JSON в объекты.
- JSON поддерживает списки объектов, помогая избежать беспорядочных преобразований списков в реляционную модель данных.
- JSON — это широко используемый формат файлов для баз данных NoSQL, таких как MongoDB, Couchbase и Azure Cosmos DB.

Как выглядит JSON файл?

В нотации JSON это выглядит так:

Объект — неупорядоченный набор пар ключ-значение. Объект начинается с { и заканчивается }. Ключ и значение разделяются двоеточием, сами пары ключ-значение разделяются запятой.

```
{"email" : "ivanov@mail.ru",  
  "name" : "Иванов Иван Иванович"}
```

Как выглядит JSON файл?

В качестве значений может использоваться массив:

Массив — упорядоченная коллекция значений.

Массив начинается с [и заканчивается].

Значения разделены запятой.

```
["ivanov@mail.ru", "petrov@mail.ru",  
"sidorov@mail.ru"]
```

Значение может быть строкой в двойных кавычках, числом, true, false, null, объектом или массивом.

Эти структуры могут быть вложенными.

JSON – Чтение и запись данных

Модуль `json` позволяет легко записывать и читать данные в формате JSON.

Для чтения данных в JSON-формате есть два метода:

- `json.load()` — метод считывает файл в формате JSON и возвращает объект Python
- `json.loads()` — метод считывает строку в формате JSON и возвращает объект Python

Для записи информации в формате JSON также два метода:

- `json.dump()` — метод записывает объект Python в файл в формате JSON
- `json.dumps()` — метод преобразует объект Python в строку в формате JSON

Таблица конвертации данных

Таблица конвертации данных Python в JSON

Python	JSON
dict	object
list, tuple	array
str	string
int, float	number(int, real)
True, False	true, false
None	null

Таблица конвертации JSON в данные Python

JSON	Python
object	dict
array	list
string	str
number(int, real)	int, float
true, false	True, False
null	None

Преобразуем json-строку в словарь

```
import json

string = '{"id":765,"email":"ivanov@mail.ru","surname":"Иванов","age":45,"admin":false,"friends":[123,456,789]}'
data = json.loads(string)
print(data["email"])
print(data["surname"])
print(data["admin"])
print(data["friends"])
```

ivanov@mail.ru
Иванов
False
[123, 456, 789]

Читаем json-данные из файла и преобразуем в словарь

```
import json

# читаем json-данные из файла и преобразуем в словарь

with open('data.json', encoding='UTF-8') as file:
    data = json.load(file)
print(data["email"])
print(data["surname"])
print(data["admin"])
print(data["friends"])
```

```
ivanov@mail.ru
Иванов
False
[123, 456, 789]
```

Преобразуем словарь в json-строку

```
import json

data = {"id": 765, "email": "ivanov@mail.ru", "surname": "Иванов", "age": 45, "admin": False,
        "friends": [123, 456, 789]}

# преобразуем словарь в json-строку
string = json.dumps(data)
print(string)

# преобразуем словарь в json-строку
string = json.dumps(data, ensure_ascii=False)
print(string)
```

```
{"id": 765, "email": "ivanov@mail.ru", "surname": "\u0418\u0432\u0430\u043d\u043e\u0432", "age": 45, "admin": false, "friends": [123, 456, 789]}
{"id": 765, "email": "ivanov@mail.ru", "surname": "Иванов", "age": 45, "admin": false, "friends": [123, 456, 789]}
```

Преобразуем словарь в json и записываем в файл

```
import json

data = {"id": 765, "email": "ivanov@mail.ru", "surname": "Иванов", "age": 45, "admin": False,
        "friends": [123, 456, 789]}

# преобразуем словарь в json и записываем в файл
with open('data.json', 'w', encoding='UTF-8') as file:
    json.dump(data, file)

# преобразуем словарь json и записываем в файл
with open('data.json', 'w', encoding='UTF-8') as file:
    json.dump(data, file, ensure_ascii=False)
```

Преобразуем словарь в json и записываем в файл

```
data.json x
1 {"id": 765,
2   "email": "ivanov@mail.ru",
3   "surname": "\u0418\u0432\u0430\u043d\u043e\u0432",
4   "age": 45,
5   "admin": false,
6   "friends": [123, 456, 789]}
```

```
data.json x
1 {"id": 765,
2   "email": "ivanov@mail.ru",
3   "surname": "Иванов",
4   "age": 45,
5   "admin": false,
6   "friends": [123, 456, 789]}
7
```



Домашнее задание

Задача №1

Пользователь будет вводить название и стоимость каждой своей покупки за день, до тех пор пока он не напишет “стоп”. Ваша задача записать это в json файл в формате:

```
{"название" : "яблоко",  
  "стоимость": "200"}
```

Задача №2

Прочитать файл из предыдущего задания и вывести стоимость всех покупок за день