

Практическая работа № 5

Работа с файлами (продолжение)

2. CSV (Comma Separated Values)

По сути, он является обычным текстовым файлом, в котором 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. В Python есть специальная библиотека для работы с такими файлами. Давайте прочитаем данные из отчета об анкетировании на первом занятии.

In [2]:

```
import csv
```

In [2]:

```
with open('Entering_quiz_2.csv', encoding = 'utf8') as csvfile:
    reader_object = csv.reader(csvfile, delimiter = ",")
    for row in reader_object:
        print(row)
```

```
['Отметка времени', 'Баллы', 'Укажите Вашу основную специальность', 'На ка
ком курсе учитесь?', 'Оцените уровень Вашей подготовки в программировании
вообще', 'Оцените уровень Вашей подготовки в программировании на Python',
'Как у Вас с математикой?', 'А что насчет машинного обучения?', 'Чего ждет
е от этого курса?', 'Напишите свой вариант', 'Как с английским?', 'Какую о
перационную систему сейчас используете?']
['04.10.2022 12:41:12', '', 'ссссс', '3', 'ниндзя', 'средний', 'Отлично!',
'Люблю, умею, практикую', 'Сам не знаю, просто все побежали - и я побежал,
Хочу делать нейронные сети, NLP (работа с естественным языком), Техническо
е зрение', 'тралала', '', '']
['04.10.2022 14:30:49', '', 'Промышленная электроника', 'работаю', 'ниндз
я', 'начальный', 'Терпимо', 'Впервые слышу', 'Основы ООП, Функциональное п
рограммирование', '', '', '']
['08.10.2022 11:59:02', '', 'Прикладная математика и информатика', '4', 'н
ачальный', 'начальный', 'Отлично!', 'Интересовался, проходил курсы / учили
этому в универе', 'Типы данных, базовые конструкции и основы Python, Позна
комиться с библиотеками для обработки данных, Хочу делать нейронные сети,
NLP (работа с естественным языком), DeepLearning!!! (что бы это ни значил
о))), Машинное обучение', ':)', 'Достаточно для чтения технической докумен
таментов', 'Python']
```

Здорово. Но ничего не понятно. Давайте попробуем разобрать данные

In [4]:

```
with open('Entering_quiz_2.csv', encoding = 'utf8') as csvfile:
    reader_object = csv.reader(csvfile, delimiter = ",")
    count = 0
    ans = []
    for row in reader_object:
        if count == 0:
            print(f'Файл содержит столбцы: {"", ".join(row)}')
            header = row
            count += 1
        else:
            ans.append(row)
            count += 1
```

Файл содержит столбцы: Отметка времени, Баллы, Укажите Вашу основную специальность, На каком курсе учиться?, Оцените уровень Вашей подготовки в программировании вообще, Оцените уровень Вашей подготовки в программировании на Python, Как у Вас с математикой?, А что насчет машинного обучения?, Чего ждете от этого курса?, Напишите свой вариант, Как с английским?, Какую операционную систему сейчас используете?

In []:

In [5]:

header

Out[5]:

```
['Отметка времени',
 'Баллы',
 'Укажите Вашу основную специальность',
 'На каком курсе учиться?',
 'Оцените уровень Вашей подготовки в программировании вообще',
 'Оцените уровень Вашей подготовки в программировании на Python',
 'Как у Вас с математикой?',
 'А что насчет машинного обучения?',
 'Чего ждете от этого курса?',
 'Напишите свой вариант',
 'Как с английским?',
 'Какую операционную систему сейчас используете?']
```

In [6]:

len(ans)

Out[6]:

79

In [9]:

```
print(ans[35])
```

```
['08.10.2022 14:16:24', '', 'самолет и вертолет строения ', '4', 'нулевой',
'нулевой', 'Терпимо', 'Что-то слышал, но не интересовался', 'Типы данных, ба
зовые конструкции и основы Python, Основы ООП, Функциональное программирован
ие, Научиться манипулировать данными, строить красивые графики и извлекать и
з них полезную информацию, Хочу делать нейронные сети, NLP (работа с естеств
енным языком), Техническое зрение, DeepLearning!!! (что бы это ни значил
о))), Машинное обучение, Распознавание объектов на видео', '', 'Достаточно д
ля чтения технической документации', 'Windows']
```

Видим, что у нас есть пара бессодержательных столбцов. Давайте их удалим

In [63]:

```
for item in ans:
    del item[0:2]
ans[37]
```

Out[63]:

```
['Терпимо',
'Что-то слышал, но не интересовался',
'Типы данных, базовые конструкции и основы Python, Основы ООП, Функциональн
ое программирование, Научиться манипулировать данными, строить красивые граф
ики и извлекать из них полезную информацию, Хочу делать нейронные сети, NLP
(работа с естественным языком), Техническое зрение, DeepLearning!!! (что бы
это ни значило))), Машинное обучение, Распознавание объектов на видео',
'',
'Достаточно для чтения технической документации',
'Windows']
```

In [10]:

```
del header[0:2]
```

In [11]:

```
header
```

Out[11]:

```
['Укажите Вашу основную специальность',
'На каком курсе учиться?',
'Оцените уровень Вашей подготовки в программировании вообще',
'Оцените уровень Вашей подготовки в программировании на Python',
'Как у Вас с математикой?',
'А что насчет машинного обучения?',
'Чего ждете от этого курса?',
'Напишите свой вариант',
'Как с английским?',
'Какую операционную систему сейчас используете?']
```

Итак. У нас имеется два списка с заголовками и данными столбцов. При этом список с ответами содержит вложенные списки по каждому участнику анкетирования. А что, если мы хотим сформировать

словарь, в котором ключами будут вопросы, а значениями будут списки ответов всех участников? В библиотеке csv есть инструмент DictReader, который позволяет, во-первых, задать именна столбцов, если они неопределены или нам не нравятся, во-вторых, обращаться к данным не по индексу, а по имени. С его помощью можно легко сформировать нужный словарь.

In [24]:

```
with open('Entering_quiz_2.csv', encoding = 'utf8') as csvfile:
    names = ['t','s', 'spec','class','Pr_lvl','Py_lvl','Mth_lvl','ML_lvl','Wait','Own','Eng']
    reader_object = csv.DictReader(csvfile, delimiter = ",", fieldnames = names)
    count = 0
    ans = [],[]
    for row in reader_object:
        if count == 0:
            header = row
            count += 1
        else:
            ans[0].append(row['Py_lvl'])
            ans[1].append(row['class'])
            count += 1
```

ans

Out[24]:

```
[[ 'средний',
  'начальный',
  'начальный',
  'средний',
  'нулевой',
  'средний',
  'начальный',
  'нулевой',
  'начальный',
  'нулевой',
  'средний',
  'нулевой',
  'средний',
  'нулевой',
  'начальный',
  'нулевой',
  'нулевой',
  'начальный' ]]
```

In [25]:

```
my_dict={}
my_dict['class'] = ans[1]
my_dict['Py_lvl'] = ans[0]
my_dict
```

```
2',
'2',
'4',
'2',
'2',
'2',
'2',
'2',
'2',
'2',
'2',
'3',
'2',
'2',
'2',
'2',
'4',
'3',
'2'],
'Py_lvl': ['средний',
'начальный',
'начальный',
```

In [13]:

header

Out[13]:

```
{'t': 'Отметка времени',
's': 'Баллы',
'spec': 'Укажите Вашу основную специальность',
'class': 'На каком курсе учитесь?',
'Pr_lvl': 'Оцените уровень Вашей подготовки в программировании вообще',
'Py_lvl': 'Оцените уровень Вашей подготовки в программировании на Python',
'Mth_lvl': 'Как у Вас с математикой?',
'ML_lvl': 'А что насчет машинного обучения?',
'Wait': 'Чего ждете от этого курса?',
'Own': 'Напишите свой вариант',
'Eng_lvl': 'Как с английским?',
'OS': 'Какую операционную систему сейчас используете?'}
```

Как сохранить данные в csv-файл?

In [26]:

```
with open('new_csv.csv', mode = 'w', encoding = 'utf-8') as w_file:
    file_writer = csv.writer(w_file, delimiter = ",", lineterminator="\r")
    file_writer.writerow(['Py_lvl'])
    file_writer.writerow(ans[0])
```

А если хотим несколько столбцов? Тогда нужно немножко переформировать список:

In [41]:

```
new_list = []
for idx,item in enumerate(ans[0]):
    new_list.append([item, ans[1][idx]])

with open('new_csv_2.csv', mode = 'w', encoding = 'utf-8') as w_file:
    file_writer = csv.writer(w_file, delimiter = ",", lineterminator="\r")
    file_writer.writerow(['Py_lv1', 'class'])
    file_writer.writerows(new_list)
```

Самостоятельно

Сохранить в отдельный csv-файл данные о специальностях, уровне знания математики и английского. Затем считайте этот файл и выведите его первые 10 строк.

In []:

In [4]:

In []:

3. JSON

JavaScript Object Notation

Определенным образом структурированный текстовый файл. Структура напоминает dict.

Для работы с таким форматом данных в Python есть специальная библиотека json

Нас интересуют следующие методы:

- load : преобразует файл, содержащий документ JSON в объект Python, используя специальную таблицу преобразования;
- loads : для считывания строки в формате JSON;
- dump : создает файл в формате JSON;
- dumps : создает строку в формате JSON;

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

In [6]:

```
import json
```

Перепишем данные нескольких участников опроса из нашего csv файла анкет в формате json.

In [26]:

```
from random import randint # Пусть это будут 3 случайные участника
numbers = set()
while len(numbers)<3:
    numbers.add(randint(1,79))
numbers
```

Out[26]:

```
{6, 14, 63}
```

In [34]:

```

keys = ['stud_1', 'stud_2', 'stud_3']
stud_dict={}
with open('Entering_quiz_2.csv', encoding = 'utf8') as csvfile:
    names = ['t', 's', 'spec', 'class', 'Pr_lvl', 'Py_lvl', 'Mth_lvl', 'ML_lvl', 'Wait', 'Own', 'Eng']
    reader_object = csv.DictReader(csvfile, delimiter = ",", fieldnames = names)
    count = 0
    i=0
    for row in reader_object:
        if count in numbers:
            stud_dict[keys[i]] = {

                'class':row['class'], 'Mth_lvl': row['Mth_lvl'], 'Eng_lvl' : row['Eng_lvl']
            }
            i+=1
        count +=1

stud_dict

```

Out[34]:

```

{'stud_1': {'class': '3',
  'Mth_lvl': 'Отлично!',
  'Eng_lvl': 'Достаточно для чтения технической документации',
  'Pr_lvl': 'средний'},
 'stud_2': {'class': '2',
  'Mth_lvl': 'Отлично!',
  'Eng_lvl': 'Достаточно для чтения технической документации',
  'Pr_lvl': 'средний'},
 'stud_3': {'class': '2',
  'Mth_lvl': 'Терпимо',
  'Eng_lvl': 'Плохо, но я стараюсь',
  'Pr_lvl': 'нулевой'}}

```

In [40]:

```

stud_json = json.dumps(stud_dict)
stud_json

```

Out[40]:

```

'{"stud_1": {"class": "3", "Mth_lvl": "\u041e\u0442\u043b\u0438\u0447\u043d\u043e!", "Eng_lvl": "\u0414\u043e\u0441\u0442\u0430\u0442\u043e\u0447\u043d\u043e \u0434\u043b\u044f \u0447\u0442\u0435\u043d\u0438\u044f \u0442\u0435\u0445\u043d\u0438\u0447\u0435\u0441\u043a\u043e\u0439 \u0434\u043e\u043a\u0443\u043c\u0435\u043d\u0442\u0430\u0446\u0438\u0438", "Pr_lvl": "\u0441\u0440\u0435\u0434\u043d\u0438\u0439"}, "stud_2": {"class": "2", "Mth_lvl": "\u041e\u0442\u043b\u0438\u0447\u043d\u043e!", "Eng_lvl": "\u0414\u043e\u0441\u0442\u0430\u0442\u043e\u0447\u043d\u043e \u0434\u043b\u044f \u0447\u0442\u0435\u043d\u0438\u044f \u0442\u0435\u0445\u043d\u0438\u0447\u0435\u0441\u043a\u043e\u0439 \u0434\u043e\u043a\u0443\u043c\u0435\u043d\u0442\u0430\u0446\u0438\u0438", "Pr_lvl": "\u0441\u0440\u0435\u0434\u043d\u0438\u0439"}, "stud_3": {"class": "2", "Mth_lvl": "\u0422\u0435\u0440\u043f\u0438\u043c\u043e", "Eng_lvl": "\u041f\u043b\u043e\u0445\u043e, \u043d\u043e \u044f \u0441\u0442\u0430\u0440\u0430\u044e\u0441\u044c", "Pr_lvl": "\u043d\u0443\u043b\u0435\u0432\u043e\u0439"}}'

```


Если добавить отступы, будет аккуратней:

Для обратного преобразования есть функция `loads`:

In [41]:

```
dict_back = json.loads(stud_json)
dict_back
```

Out[41]:

```
{'stud_1': {'class': '3',
  'Mth_lvl': 'Отлично!',
  'Eng_lvl': 'Достаточно для чтения технической документации',
  'Pr_lvl': 'средний'},
 'stud_2': {'class': '2',
  'Mth_lvl': 'Отлично!',
  'Eng_lvl': 'Достаточно для чтения технической документации',
  'Pr_lvl': 'средний'},
 'stud_3': {'class': '2',
  'Mth_lvl': 'Терпимо',
  'Eng_lvl': 'Плохо, но я стараюсь',
  'Pr_lvl': 'нулевой'}}
```

In [42]:

```
type(dict_back)
```

Out[42]:

```
dict
```

Можно одной командой и преобразовать, и сразу записать в файл в формате json:

In [43]:

```
with open('stud.json','w') as f:
    json.dump(stud_dict, f, indent = 3)
```

Появился готовый файл. Мы можем его считать.

In [44]:

```
with open('stud.json','r') as f:
    data = json.load(f)
```

In [45]:

```
print(data)
type(data)
```

```
{'stud_1': {'class': '3', 'Mth_lvl': 'Отлично!', 'Eng_lvl': 'Достаточно для
чтения технической документации', 'Pr_lvl': 'средний'}, 'stud_2': {'class':
'2', 'Mth_lvl': 'Отлично!', 'Eng_lvl': 'Достаточно для чтения технической до
кументации', 'Pr_lvl': 'средний'}, 'stud_3': {'class': '2', 'Mth_lvl': 'Терп
имо', 'Eng_lvl': 'Плохо, но я стараюсь', 'Pr_lvl': 'нулевой'}}
```

Out[45]:

```
dict
```

Тадам! Это обычный словарь. Можно обращаться к его элементам по ключу, можно итерироваться и пр. Это удобно, т.к. нет нужды в дополнительных преобразованиях данных.

In [46]:

```
import requests
```

In [47]:

```
url = 'http://ip-api.com/json/'  
loc_dict = requests.get(url).json()
```

In [48]:

```
loc_dict
```

Out[48]:

```
{'status': 'success',  
 'country': 'Russia',  
 'countryCode': 'RU',  
 'region': 'ROS',  
 'regionName': 'Rostov Oblast',  
 'city': 'Taganrog',  
 'zip': '347900',  
 'lat': 47.2189,  
 'lon': 38.9168,  
 'timezone': 'Europe/Moscow',  
 'isp': 'nr2.neo.nas.tgn.pg19.ru',  
 'org': '',  
 'as': 'AS49037 Mikhail Mayorov',  
 'query': '151.0.51.128'}
```

Самостоятельно

1

1) Создайте словарь, содержащий только 5 полей из loc_dict (поля на свой выбор). Запишите его в отдельный файл в формате json.

2) Считайте данные из своего файла.

3) Создайте csv-файл и запишите свои данные туда.

4) Считайте свой csv-файл, сохранив в отдельный список loc_fields названия полей, и в отдельный список loc_data значения.

In []:

2

Реализовать собственное key-values хранилище. Данные будут сохраняться в файле store.json. Добавление новых данных в хранилище и получение текущих значений осуществляется с помощью функции storage(). Вашей задачей будет написать реализацию функции storage(). Она может вызываться со следующими параметрами: storage(<имя ключа>, <значение>), где <имя ключа> - ключ по которому сохраняются/получаются значения, <значение> - сохраняемое значение.

Если в функцию переданы оба значения, происходит добавление переданного значения по ключу и сохранение данных в файле. Если передано только имя ключа, происходит чтение файла хранилища и вывод на печать значений, которые были сохранены по данному ключу. Обратите внимание, что значения по одному ключу не перезаписываются, а добавляются к уже сохраненным. Другими словами - по одному ключу могут храниться несколько значений. При выводе на печать, значения выводятся в порядке их добавления в хранилище. Формат вывода на печать для нескольких значений:

value_1, value_2

Обратите внимание на пробел после запятой. Если значений по ключу не было найдено, выведите пустую строку или None.

In []:

Выводы

In []: