

Чтение файлов

Рассматриваем задачу сквозной аналитики. Будем работать с этими файлами:

- visit_log.csv
- purchase_log.txt

Способ 1. Делаем все пошагово

Опции при открытии файлов:

- 'r' открыть для чтения. Файл при этом не меняется
- 'w' открыть файл для записи. Файл при этом полностью чистится (!)
- 'a' открыть файл для добавления. Файл не меняется, все записи добавляются в конец

In [22]:

```
# открываем файл для чтения (опция r)
f = open("C:/Users/k.seleznev/Python_learning/notebooks/04. допфайлы/visit_log.csv", 'r', e
```

In [2]:

```
f
```

Out[2]:

```
<_io.TextIOWrapper name='C:/Users/k.seleznev/Python_learning/notebooks/04. д
опфайлы/visit_log.csv' mode='r' encoding='utf-8'>
```

In [3]:

```
f.__sizeof__()
```

Out[3]:

```
192
```

In [4]:

```
# прочитать первую строку
f.readline()
```

Out[4]:

```
'user_id,source\n'
```

In [5]:

```
# прочитать еще одну
f.readline()
```

Out[5]:

```
'6450655ae8,other\n'
```

In [6]:

```
# можно и так  
next(f)
```

Out[6]:

```
'b4ea53e670,other\n'
```

In [7]:

```
# прочитать все содержимое файла в переменную content  
content = f.readlines()
```

In [8]:

```
content[:5]
```

Out[8]:

```
['0e37347152,other\n',  
 '96064ae9e0,other\n',  
 'e1bd168161,context\n',  
 '71bcf169b4,other\n',  
 'e0aee73c5d,other\n']
```

In [9]:

```
content[-1]
```

Out[9]:

```
'c23704ea54,other\n'
```

In [10]:

```
f.seek(0)
```

Out[10]:

```
0
```

In [11]:

```
f.readline()
```

Out[11]:

```
'user_id,source\n'
```

In [12]:

```
# построчное чтение файла
for line in f:
    print(line)

# break
```

e0aee73c5d,other

1f8845e157,context

78d22c5c78,other

dd85040770,other

2301051b25,other

86835b81ac,context

90d9a575a2,other

1b75c4f23f,other

d94b908660,None

f697206af5,other

In [23]:

```
i = 0
for line in f:
    print(line)

    i += 1
    if i > 5:
        break
```

user_id,source

6450655ae8,other

b4ea53e670,other

0e37347152,other

96064ae9e0,other

e1bd168161,context

In [24]:

```
for i, line in enumerate(f):  
    print(i, line)  
  
    if i > 5:  
        break
```

```
0 71bcf169b4,other  
1 e0aee73c5d,other  
2 1f8845e157,context  
3 78d22c5c78,other  
4 dd85040770,other  
5 2301051b25,other  
6 86835b81ac,context
```

Часто используется конструкция

```
another_line.strip().split(',')
```

In [25]:

```
another_line = f.readline()  
another_line
```

Out[25]:

```
'90d9a575a2,other\n'
```

In [26]:

```
# удаляем перенос строки и лишние пробелы  
another_line.strip()
```

Out[26]:

```
'90d9a575a2,other'
```

In [27]:

```
' ;ceda3a4603,other\n;--!--'.strip(';--!')
```

Out[27]:

```
'ceda3a4603,other\n'
```

In [28]:

```
# разбиваем столбцы
another_line.strip().split(',')
```

Out[28]:

```
['90d9a575a2', 'other']
```

In [29]:

```
# закрытие файла
f.close()
```

In [30]:

```
# после закрытия не получится прочитать
f.readline()
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-30-f561c2044435> in <module>
      1 # после закрытия не получится прочитать
----> 2 f.readline()
```

ValueError: I/O operation on closed file.

In [32]:

```
# прочитать все строки файла в список (т. е. в оперативную память)
f = open('C:/Users/k.seleznev/Python_learning/notebooks/04. допфайлы/visit_log.csv', 'r')

content = f.readlines()

f.close()
```

In [33]:

```
content[:5]
```

Out[33]:

```
['user_id,source\n',
 '6450655ae8,other\n',
 'b4ea53e670,other\n',
 '0e37347152,other\n',
 '96064ae9e0,other\n']
```

Запись в файл

Зачем все-таки открывать и закрывать файлы?

In [33]:

```
f = open('results.csv', 'w')
```

In [34]:

```
f.write('Начинаю запись первой строки...\n')
```

Out[34]:

32

In [35]:

```
f.write('Начинаю запись второй строки...\n')
```

Out[35]:

32

Здесь кто-то еще хочет в него записать

In [36]:

```
my_friend_results = open('results.csv', 'w')
```

In [37]:

```
my_friend_results.write('Тут еще результаты есть')
```

Out[37]:

23

In [38]:

```
# пишем свой результат  
f.close()
```

In [39]:

```
# и наш результат перезаписывается  
my_friend_results.close()
```

Контекстный менеджер

Как обезопасить себя от подобных накладок

In [40]:

```
with open('results.csv', 'w') as f:  
    f.write('Начинаю запись первой строки...\n')  
    f.write('Начинаю запись второй строки...\n')  
  
    my_friend_results = open('results.csv', 'w')  
    my_friend_results.write('Тут еще результаты есть')  
    my_friend_results.close()
```

А можно читать и сразу в файл записывать?

Напишите функцию, которая фильтрует файл visit_log.csv по источнику context и пишет результат в visits_context.csv. Используйте функцию из второго упражнения для проверки результата.

In []:

```
with open('visit_log.csv', 'r') as f:
    with open('visits_context.csv', 'w') as f2write:
        for line in f:
            if 'context' in line:
                f2write.write(line)
```

In []:

In []:

Чтение списков и словарей из файла

Смотрим что в файле purchase_log.txt. Похоже тут не строки, а словари

In [34]:

```
with open('C:/Users/k.seleznev/Python_learning/notebooks/04. допфайлы/purchase_log.txt') as f:
    print([next(f) for x in range(5)])
```

```
[{"user_id": "user_id", "category": "category"}\n', '{"user_id": "1840e0b9d4", "category": "P\ud83d\ude02P\ud83d\ude02P\ud83d\ude02"}\n', '{"user_id": "4e4f90fcfb", "category": "P\ud83d\ude02P\ud83d\ude02P\ud83d\ude02"}\n', '{"user_id": "afea8d72fc", "category": "P\ud83d\ude02P\ud83d\ude02P\ud83d\ude02"}\n', '{"user_id": "373a6055fe", "category": "P\ud83d\ude02P\ud83d\ude02P\ud83d\ude02"}\n']
```

In [35]:

```
import json
```

In [36]:

```
# перевод строки в словарь
dict_in_string = '{"a": 1, "b": 2}'

type(json.loads(dict_in_string))
```

Out[36]:

```
dict
```

In [37]:

```
json.loads(dict_in_string)
```

Out[37]:

```
{'a': 1, 'b': 2}
```

In [38]:

```
# перевод строки в список
list_in_string = '[1, 2, 3]'

json.loads(list_in_string)[-1]
```

Out[38]:

3

In [48]:

```
json.loads(list_in_string)
```

Out[48]:

[1, 2, 3]

In [49]:

```
i = 0
with open('purchase_log.txt') as f:
    for line in f:
        line = line.strip()

        dict_ = json.loads(line)
        print(dict_, type(dict_))

        i += 1
    if i > 5:
        break
```

```
{'user_id': 'user_id', 'category': 'category'} <class 'dict'>
{'user_id': '1840e0b9d4', 'category': 'Продукты'} <class 'dict'>
{'user_id': '4e4f90fcfb', 'category': 'Электроника'} <class 'dict'>
{'user_id': 'afea8d72fc', 'category': 'Электроника'} <class 'dict'>
{'user_id': '373a6055fe', 'category': 'Бытовая техника'} <class 'dict'>
{'user_id': '9b2ab046f3', 'category': 'Электроника'} <class 'dict'>
```

In []:

```
dict_ = 1
```

Из словаря в строку тоже можно

In [50]:

```
data = {'user_id': '1840e0b9d4', 'category': 'Продукты'}
```

In [51]:

```
str(data)
```

Out[51]:

```
"{'user_id': '1840e0b9d4', 'category': 'Продукты'}"
```


In [52]:

```
json.dumps(data)
```

Out[52]:

```
'{"user_id": "1840e0b9d4", "category": "\\u041f\\u0440\\u043e\\u0434\\u0443\\u043a\\u0442\\u044b"}'
```

In [53]:

```
type(json.dumps(data))
```

Out[53]:

str

Модуль pickle

Запись объекта сразу в файл как поток байтов

In [54]:

```
import pickle
```

In [55]:

```
data = {'user_id': '1840e0b9d4', 'category': 'Продукты'}
```

In [56]:

```
with open('data.pickle', 'wb') as f:  
    pickle.dump(data, f)
```

In [57]:

```
# прочитать объект из такого файла  
  
with open('data.pickle', 'rb') as f:  
    dict_ = pickle.load(f)  
  
dict_, dict_['user_id']
```

Out[57]:

```
({'user_id': '1840e0b9d4', 'category': 'Продукты'}, '1840e0b9d4')
```

Чтение файлов из папки

In [62]:

```
import os
```

In [64]:

```
# чтение файлов и папок
for file in os.listdir('data'):
    if '.csv' in file:
        print(file)
```

```
file_2.csv
file_3.csv
file_1.csv
```

In [65]:

```
# чтение всех файлов и папок, в том числе вложенных
for root, directory, file in os.walk('data'):
    print(root, directory, file)
```

```
data ['more_data'] ['.DS_Store', 'file_2.csv', 'file_3.csv', 'file_1.csv']
data/more_data [] ['file_4.csv', 'file_5.csv']
```

Каталог пакетов pip (Python Package Index)

Варианты установки

1. С помощью Anaconda Navigator
2. В командной строке (Terminal на маке)

```
pip install package_name
```

3. Скачать версию с github и установить вручную

```
pip install .
```

Пакеты устанавливаются в определенное окружение! Вам пригодится virtualenv

In []:

```
!pip install pyyaml
```

In []:

```
import plotly
```