Работа с файлами данных

Материалы:

- Макрушин С.В. "Лекция 4: Форматы данных"
- https://docs.python.org/3/library/json.html
- https://docs.python.org/3/library/pickle.html
- https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/bs4ru.html
- Уэс Маккини. Python и анализ данных

v 0.2 05.10.22

Задачи для совместного разбора

1. Вывести все адреса электронной почты, содержащиеся в адресной книге `addres-book.json`

2. Вывести телефоны, содержащиеся в адресной книге `addres-book.json`

3. По данным из файла `addres-book-q.xml` сформировать список словарей с телефонами каждого из людей.

Лабораторная работа №3.1

JSON

In [35]:
```
import json
import pandas as pd
```

1.1 Считайте файл `contributors_sample.json`. Воспользовавшись модулем `json`, преобразуйте содержимое файла в соответствующие объекты python. Выведите на экран информацию о первых 3 пользователях.

In [38]:
```
with open("contributors_sample.json", "r") as file:
    data = json.load(file)
data[:5]
```

Out[38]:

```
[{'username': 'uhebert',
  'name': 'Lindsey Nguyen',
  'sex': 'F',
  'address': '01261 Cameron Spring\nTaylorfurt, AK 97791',
  'mail': 'jsalazar@gmail.com',
  'jobs': ['Energy engineer',
   'Engineer, site',
   'Environmental health practitioner',
   'Biomedical scientist',
   'Jewellery designer'],
  'id': 35193},
 {'username': 'vickitaylor',
  'name': 'Cheryl Lewis',
  'sex': 'F',
  'address': '66992 Welch Brooks\nMarshallshire, ID 56004',
  'mail': 'bhudson@gmail.com',
  'jobs': ['Music therapist',
   'Volunteer coordinator',
   'Designer, interior/spatial'],
  'id': 91970},
 {'username': 'sheilaadams',
  'name': 'Julia Allen',
  'sex': 'F',
  'address': 'Unit 1632 Box 2971\nDPO AE 23297',
  'mail': 'darren44@yahoo.com',
  'jobs': ['Management consultant',
   'Engineer, structural',
   'Lecturer, higher education',
   'Theatre manager',
   'Designer, textile'],
  'id': 1848091},
 {'username': 'nicole82',
  'name': 'Gina Stevens',
  'sex': 'F',
  'address': '9880 Michelle Bridge\nNew Kimberlybury, WY 02583',
  'mail': 'stevensonsarah@hotmail.com',
  'jobs': ['Mechanical engineer', 'Retail banker', 'Barrister'],
  'id': 50969},
 {'username': 'jean67',
  'name': 'Nicholas Harrington',
  'sex': 'M',
  'address': '9080 Monica Crescent Suite 820\nNorth Deanbury, HI 28977',
  'mail': 'denise42@gmail.com',
  'jobs': ['Network engineer',
   'Youth worker',
   'Primary school teacher',
   'Engineer, broadcasting (operations)'],
  'id': 676820}]
```


In [40]:
```
for user in data[:3]:
    print(json.dumps(user, indent=4, ensure_ascii=False))
```

```
{
    "username": "uhebert",
    "name": "Lindsey Nguyen",
    "sex": "F",
    "address": "01261 Cameron Spring\nTaylorfurt, AK 97791",
    "mail": "jsalazar@gmail.com",
    "jobs": [
        "Energy engineer",
        "Engineer, site",
        "Environmental health practitioner",
        "Biomedical scientist",
        "Jewellery designer"
    ],
    "id": 35193
}
{
    "username": "vickitaylor",
    "name": "Cheryl Lewis",
    "sex": "F",
    "address": "66992 Welch Brooks\nMarshallshire, ID 56004",
    "mail": "bhudson@gmail.com",
    "jobs": [
        "Music therapist",
        "Volunteer coordinator",
        "Designer, interior/spatial"
    ],
    "id": 91970
}
{
    "username": "sheilaadams",
    "name": "Julia Allen",
    "sex": "F",
    "address": "Unit 1632 Box 2971\nDPO AE 23297",
    "mail": "darren44@yahoo.com",
    "jobs": [
        "Management consultant",
        "Engineer, structural",
        "Lecturer, higher education",
        "Theatre manager",
        "Designer, textile"
    ],
    "id": 1848091
}
```


1.2 Выведите уникальные почтовые домены, содержащиеся в почтовых адресах людей

In [43]:
```python
unique_domains = set()
for user in data:
    mail = user['mail']
    unique_domains.add(mail.split("@")[-1])

unique_domains
```

Out[43]:
```
{'gmail.com', 'hotmail.com', 'yahoo.com'}
```

1.3 Напишите функцию, которая по `username` ищет человека и выводит информацию о нем. Если пользователь с заданным `username` отсутствует, возбудите исключение `ValueError`

In [46]:
```python
def find_user(data, username):
    for user in data:
        if user['name'].strip().lower() == username.strip().lower():
            return user
    raise ValueError(f'Пользователь {username} не найден')
```

In [48]:
```python
find_user(data, "Egor Shipilov")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[48], line 1
----> 1 find_user(data, "Egor Shipilov")

Cell In[46], line 5, in find_user(data, username)
      3     if user['name'].strip().lower() == username.strip().lower():
      4         return user
----> 5 raise ValueError(f'Пользователь {username} не найден')

ValueError: Пользователь Egor Shipilov не найден
```

In [50]:
```python
find_user(data, "Julia Allen")
```

Out[50]:
```
{'username': 'sheilaadams',
 'name': 'Julia Allen',
 'sex': 'F',
 'address': 'Unit 1632 Box 2971\nDPO AE 23297',
 'mail': 'darren44@yahoo.com',
 'jobs': ['Management consultant',
  'Engineer, structural',
  'Lecturer, higher education',
  'Theatre manager',
  'Designer, textile'],
 'id': 1848091}
```

1.4 Посчитайте, сколько мужчин и женщин присутсвует в этом наборе данных.

In [53]:
```
sex_cnt = {}
for user in data:
    if user['sex'] == "F":
        sex_cnt['F'] = sex_cnt.get('F', 0) + 1
    else:
        sex_cnt['M'] = sex_cnt.get('M', 0) + 1

sex_cnt
```

Out[53]:
```
{'F': 2136, 'M': 2064}
```

1.5 Создайте `pd.DataFrame` `contributors`, имеющий столбцы `id`, `username` и `sex`.

In [56]:
```
contributors = pd.DataFrame(data, columns = ["id", "username", "sex"])
contributors.set_index('id', inplace=True)
contributors.head(10)
```

Out[56]:

id	username	sex
35193	uhebert	F
91970	vickitaylor	F
1848091	sheilaadams	F
50969	nicole82	F
676820	jean67	M
64918	james67	F
113941	woodmarissa	M
398160	sampsontammy	M
35635	jonathan18	M
718054	michael53	M

1.6 Загрузите данные из файла `recipes_sample.csv` (**ЛР2**) в таблицу `recipes`. Объедините `recipes` с таблицей `contributors` с сохранением строк в том случае, если информация о человеке отсутствует в JSON-файле. Для скольких человек информация отсутствует?

In [59]:
```
recipes = pd.read_csv("recipes_sample.csv")
recipes.set_index('id', inplace=True)
recipes.head()
```

Out[59]:

id	name	minutes	contributor_id	submitted	n_steps	description	n_ingredients
44123	george s at the cove black bean soup	90	35193	2002-10-25	NaN	an original recipe created by chef scott meska...	18.0
67664	healthy for them yogurt popsicles	10	91970	2003-07-26	NaN	my children and their friends ask for my homem...	NaN
38798	i can t believe it s spinach	30	1533	2002-08-29	NaN	these were so go, it surprised even me.	8.0
35173	italian gut busters	45	22724	2002-07-27	NaN	my sister-in-law made these for us at a family...	NaN
84797	love is in the air beef fondue sauces	25	4470	2004-02-23	4.0	i think a fondue is a very romantic casual din...	NaN

In [60]:
```
df = recipes.merge(contributors, on="id", how="left")
missing_info_cnt = (df['username'].isna().sum())

df.head()
```

Out[60]:

id	name	minutes	contributor_id	submitted	n_steps	description	n_ingredients	username	sex
44123	george s at the cove black bean soup	90	35193	2002-10-25	NaN	an original recipe created by chef scott meska...	18.0	NaN	NaN
67664	healthy for them yogurt popsicles	10	91970	2003-07-26	NaN	my children and their friends ask for my homem...	NaN	NaN	NaN
38798	i can t believe it s spinach	30	1533	2002-08-29	NaN	these were so go, it surprised even me.	8.0	NaN	NaN
35173	italian gut busters	45	22724	2002-07-27	NaN	my sister-in-law made these for us at a family...	NaN	NaN	NaN
84797	love is in the air beef fondue sauces	25	4470	2004-02-23	4.0	i think a fondue is a very romantic casual din...	NaN	NaN	NaN

In [63]:
```
missing_info_cnt
```

Out[63]:
```
29830
```

pickle

In [66]:
```
import pickle
import os
```

2.1 На основе файла `contributors_sample.json` создайте словарь следующего вида:

```
{
    должность: [список username людей, занимавших эту должность]
}
```

In [69]:
```
jobs = set()
for user in data:
    for job in user['jobs']:
        jobs.add(job)
```

In [71]:
```
posts = dict()
for user in data:
    for job in jobs:
        if job in user['jobs']:
            if job not in posts:
                posts[job] = []
            posts[job].append(user["username"])

for key, value in list(posts.items())[:5]:
    print(f"{key}: {value}\n")
```

```
Environmental health practitioner: ['uhebert', 'jonathanchristian', 'xjohnson', 'dsmith', 'james01', 'nancytaylo
r', 'ztaylor', 'andrewwoods', 'susan54', 'fmaldonado', 'james74', 'bakerjacob', 'stephanie81', 'whitejoseph', 'q
olson', 'hknox', 'gonzalesdaniel', 'tranronald', 'jessegreen', 'stephanie69', 'ellisdennis', 'melaniejohnson', '
bradleyalexander', 'chadandrews', 'thomas33', 'john93', 'tanderson', 'dward', 'kathleenbarnett']

Energy engineer: ['uhebert', 'annmoore', 'garysilva', 'martinezashley', 'sextonsheila', 'pjames', 'smithjonathan
', 'wardjames', 'cwheeler', 'ucarlson', 'robert71', 'johnsontheresa', 'amanda41', 'stacey47', 'timothynelson', '
rogersmichael', 'melissa94', 'wmcdaniel', 'charles74', 'smithjennifer', 'clintonjones']

Jewellery designer: ['uhebert', 'lopezantonio', 'ojames', 'leonnicole', 'daltonmelissa', 'joseph26', 'donnacurry
', 'stewartpamela', 'williamsbill', 'garciaduane', 'megan77', 'victoriachavez', 'richardherman', 'elainerodrigue
z', 'mark85', 'dali', 'bfernandez', 'heatherphillips', 'sdaniel', 'brandigreen', 'zachary61', 'sarahleonard', 'h
arrisonjeffery']

Biomedical scientist: ['uhebert', 'smithheather', 'epittman', 'scotttyrone', 'limadeline', 'robertsmith', 'fried
manronald', 'sarahirwin', 'xmiller', 'jeremy66', 'bakertammie', 'bnorris', 'betty20', 'ambermartinez', 'stephani
e16', 'leebenjamin', 'derekschmidt', 'katkins', 'ginacantrell', 'richard23', 'barbara93', 'katherinewalters', 'b
randi41']

Engineer, site: ['uhebert', 'nancy12', 'andrea03', 'catherineross', 'wesley32', 'natalieross', 'rossdoris', 'chr
istophersmith', 'dbooker', 'ericarobertson', 'trantricia', 'tpugh', 'jasonvelez', 'samantha36', 'brandidaniels',
'tenglish', 'reyesbrett', 'austin18', 'vjohnson', 'zmejia', 'daniel04', 'cynthia20', 'morgan15', 'avaldez', 'jes
sica92', 'laurieholloway', 'baileyvictoria']
```

2.2 Сохраните результаты в файл `job_people.pickle` и в файл `job_people.json` с использованием форматов pickle и JSON соответственно. Сравните объемы получившихся файлов. При сохранении в JSON укажите аргумент `indent`.

In [74]:
```
with open("job_people_new.json", "w") as file_js:
```

```
    json.dump(posts, file_js ,indent=4)
```

In [76]:
```
with open("job_people_new.pickle", "wb") as file_pk:
    pickle.dump(posts, file_pk)
```

In [78]:
```
pickle_size = os.path.getsize('job_people_new.pickle')
json_size = os.path.getsize('job_people_new.json')

pickle_size, json_size
```

Out[78]: (132047, 388626)

2.3 Считайте файл `job_people.pickle` и продемонстрируйте, что данные считались корректно.

In [81]:
```
with open('job_people_new.pickle', 'rb') as f:
    loaded_data = pickle.load(f)

for key, value in list(loaded_data.items())[:5]:
    print(f"{key}: {value}\n")
```

```
Environmental health practitioner: ['uhebert', 'jonathanchristian', 'xjohnson', 'dsmith', 'james01', 'nancytaylo
r', 'ztaylor', 'andrewwoods', 'susan54', 'fmaldonado', 'james74', 'bakerjacob', 'stephanie81', 'whitejoseph', 'q
olson', 'hknox', 'gonzalesdaniel', 'tranronald', 'jessegreen', 'stephanie69', 'ellisdennis', 'melaniejohnson', '
bradleyalexander', 'chadandrews', 'thomas33', 'john93', 'tanderson', 'dward', 'kathleenbarnett']

Energy engineer: ['uhebert', 'annmoore', 'garysilva', 'martinezashley', 'sextonsheila', 'pjames', 'smithjonathan
', 'wardjames', 'cwheeler', 'ucarlson', 'robert71', 'johnsontheresa', 'amanda41', 'stacey47', 'timothynelson', '
rogersmichael', 'melissa94', 'wmcdaniel', 'charles74', 'smithjennifer', 'clintonjones']

Jewellery designer: ['uhebert', 'lopezantonio', 'ojames', 'leonnicole', 'daltonmelissa', 'joseph26', 'donnacurry
', 'stewartpamela', 'williamsbill', 'garciaduane', 'megan77', 'victoriachavez', 'richardherman', 'elainerodrigue
z', 'mark85', 'dali', 'bfernandez', 'heatherphillips', 'sdaniel', 'brandigreen', 'zachary61', 'sarahleonard', 'h
arrisonjeffery']

Biomedical scientist: ['uhebert', 'smithheather', 'epittman', 'scotttyrone', 'limadeline', 'robertsmith', 'fried
manronald', 'sarahirwin', 'xmiller', 'jeremy66', 'bakertammie', 'bnorris', 'betty20', 'ambermartinez', 'stephani
e16', 'leebenjamin', 'derekschmidt', 'katkins', 'ginacantrell', 'richard23', 'barbara93', 'katherinewalters', 'b
randi41']

Engineer, site: ['uhebert', 'nancy12', 'andrea03', 'catherineross', 'wesley32', 'natalieross', 'rossdoris', 'chr
istophersmith', 'dbooker', 'ericarobertson', 'trantricia', 'tpugh', 'jasonvelez', 'samantha36', 'brandidaniels',
'tenglish', 'reyesbrett', 'austin18', 'vjohnson', 'zmejia', 'daniel04', 'cynthia20', 'morgan15', 'avaldez', 'jes
sica92', 'laurieholloway', 'baileyvictoria']
```

Лабораторная работа №3.2

XML

In [85]:
```
from bs4 import BeautifulSoup
```

3.1 По данным файла `steps_sample.xml` сформируйте словарь с шагами по каждому рецепту вида `{id_рецепта: ["шаг1", "шаг2"]}`. Сохраните этот словарь в файл `steps_sample.json`

In [88]:
```
with open('steps_sample.xml', 'r', encoding='utf-8') as file:
    soup = BeautifulSoup(file, 'xml')

recipes_sp = soup.find_all('recipe')
```

In [89]:
```
slov_steps = dict()

for recipe in recipes_sp:
    id = recipe.find("id").text
    steps = []
    step_elements = recipe.find_all("step")
    for step in step_elements:
        steps.append(step.text.strip())
    slov_steps[id] = steps

for key, value in list(slov_steps.items())[:5]:
    print(f"{key}: {value}\n")
```

```
44123: ['in 1 / 4 cup butter , saute carrots , onion , celery and broccoli stems for 5 minutes', 'add thyme , or
egano and basil', 'saute 5 minutes more', 'add wine and deglaze pan', 'add hot chicken stock and reduce by one-t
hird', 'add worcestershire sauce , tabasco , smoked chicken , beans and broccoli florets', 'simmer 5 minutes', '
add cream , simmer 5 minutes more and season to taste', 'drop in remaining butter , piece by piece , stirring un
til melted and serve immediately', 'smoked chicken: on a covered grill , slightly smoke boneless chicken , cooki
ng to medium rare', 'chef meskan uses applewood chips and does not allow the grill to become too hot']

67664: ['mix all the ingredients using a blender', 'pour into popsicle molds', 'freeze and enjoy !']

38798: ['combine all ingredients in a large bowl and mix well', 'shape into one-inch balls', 'cover and refriger
ate or freeze until ready to bake', 'preheat oven to 350 degrees', 'place on ungreased baking sheet and bake unt
il light brown']

35173: ['lay out sandwich rolls on jelly roll pans / cookie sheets', 'melt butter , mix in italian dressing mix'
, 'using a pastry or bbq brush , graciously apply seasoned butter to the top of the"bottom bun" and the top of t
he top bun', "don't miss this step , i don't know why , but it does make a difference !", 'here is where i creat
e an assembly line w / the bottoms of the buns', 'layer each bun w / ham , then swiss cheese , turkey , then che
ddar cheese , pepperoni , then mozzerella cheese', 'place"lids" on buns and place in 425 degree oven for approxi
mly 12-15 minutes , or until you see the tops start to turn golden brown']

84797: ['honey mustard sauce: whisk all the ingredients together serve warm or cold', 'easy bbq sauce: combine a
ll ingredients in a pot& cook over low heat until the sugar is dissolved', 'serve warm or cold', 'garlic dill sa
uce: mix all the ingredients and chill until ready to serve']
```

In [90]:
```python
with open("steps_sample.json", "w") as file:
    json.dump(slov_steps, file, ensure_ascii=False, indent=4)
```

3.2 По данным файла `steps_sample.xml` сформируйте словарь следующего вида: `кол-во_шагов_в_рецепте: [список_id_рецептов]`

In [92]:
```python
# 1 способ
slov_count = dict()

for recipe in recipes_sp:
    id = recipe.find("id").text
    steps = []
    count = 0
    step_elements = recipe.find_all("step")
    for step in step_elements:
        steps.append(step.text.strip())
    slov_count[id] = len(steps)

for key, value in list(slov_count.items())[:10]:
    print(f"{key}: {value}")
```

```
44123: 11
67664: 3
38798: 5
35173: 7
84797: 4
44045: 6
107229: 8
95926: 4
453467: 12
306168: 6
```

In [93]:
```python
# 2 способ
count = dict()
for key, value in slov_steps.items():
    count[key] = len(value)

for key, value in list(count.items())[:10]:
    print(f"{key}: {value}")
```

```
44123: 11
67664: 3
38798: 5
35173: 7
84797: 4
44045: 6
107229: 8
95926: 4
453467: 12
306168: 6
```

3.3 Получите список рецептов, в этапах выполнения которых есть информация о времени (часы или минуты). Для отбора подходящих рецептов обратите внимание на атрибуты соответствующих тэгов.

In [95]:
```python
with_time = []
for recipe in recipes_sp:
    id = recipe.find('id').text
    has_time = False
```

```
    for step in recipe.find_all('step'):
        if step.has_attr('has_minutes') or step.has_attr('has_hours'):
            has_time = True
            break
    if has_time:
        with_time.append(id)

with_time[:10]
```

Out[95]:
```
['44123',
 '35173',
 '453467',
 '306168',
 '50662',
 '118843',
 '149593',
 '200148',
 '310570',
 '95534']
```

3.4 Загрузите данные из файла `recipes_sample.csv` (**ЛР2**) в таблицу `recipes`. Для строк, которые содержат пропуски в столбце `n_steps`, заполните этот столбец на основе файла `steps_sample.xml`. Строки, в которых столбец `n_steps` заполнен, оставьте без изменений.

In [97]:
```
data_recipes = pd.read_csv("recipes_sample.csv")
data_recipes.head()
```

Out[97]:

	name	id	minutes	contributor_id	submitted	n_steps	description	n_ingredients
0	george s at the cove black bean soup	44123	90	35193	2002-10-25	NaN	an original recipe created by chef scott meska...	18.0
1	healthy for them yogurt popsicles	67664	10	91970	2003-07-26	NaN	my children and their friends ask for my homem...	NaN
2	i can t believe it s spinach	38798	30	1533	2002-08-29	NaN	these were so go, it surprised even me.	8.0
3	italian gut busters	35173	45	22724	2002-07-27	NaN	my sister-in-law made these for us at a family...	NaN
4	love is in the air beef fondue sauces	84797	25	4470	2004-02-23	4.0	i think a fondue is a very romantic casual din...	NaN

In [98]:
```
data_recipes['n_steps'] = data_recipes.apply(
    lambda row: count[str(row['id'])] if pd.isna(row['n_steps']) else row['n_steps'],
    axis=1
)
```

In [99]:
```
data_recipes.head()
```

Out[99]:

	name	id	minutes	contributor_id	submitted	n_steps	description	n_ingredients
0	george s at the cove black bean soup	44123	90	35193	2002-10-25	11.0	an original recipe created by chef scott meska...	18.0
1	healthy for them yogurt popsicles	67664	10	91970	2003-07-26	3.0	my children and their friends ask for my homem...	NaN
2	i can t believe it s spinach	38798	30	1533	2002-08-29	5.0	these were so go, it surprised even me.	8.0
3	italian gut busters	35173	45	22724	2002-07-27	7.0	my sister-in-law made these for us at a family...	NaN
4	love is in the air beef fondue sauces	84797	25	4470	2004-02-23	4.0	i think a fondue is a very romantic casual din...	NaN

3.5 Проверьте, содержит ли столбец `n_steps` пропуски. Если нет, то преобразуйте его к целочисленному типу и сохраните результаты в файл `recipes_sample_with_filled_nsteps.csv`

In [101…
```
miss_cnt = data_recipes['n_steps'].isna().sum()
miss_cnt
```

Out[101… `0`

In [102…
```
data_recipes['n_steps'] = data_recipes['n_steps'].astype("int64")
data_recipes.head()
```

Out[102…

	name	id	minutes	contributor_id	submitted	n_steps	description	n_ingredients
0	george s at the cove black bean soup	44123	90	35193	2002-10-25	11	an original recipe created by chef scott meska...	18.0
1	healthy for them yogurt popsicles	67664	10	91970	2003-07-26	3	my children and their friends ask for my homem...	NaN
2	i can t believe it s spinach	38798	30	1533	2002-08-29	5	these were so go, it surprised even me.	8.0
3	italian gut busters	35173	45	22724	2002-07-27	7	my sister-in-law made these for us at a family...	NaN
4	love is in the air beef fondue sauces	84797	25	4470	2004-02-23	4	i think a fondue is a very romantic casual din...	NaN