In [1]:
```
import numpy as np
```

Тема “Работа с данными в Pandas”

Задание 1

Импортируйте библиотеку Pandas и дайте ей псевдоним pd. Создайте датафрейм authors со столбцами author_id и author_name, в которых соответственно содержатся данные: [1, 2, 3] и ['Тургенев', 'Чехов', 'Островский'].

Затем создайте датафрейм book со столбцами author_id, book_title и price, в которых соответственно содержатся данные:
[1, 1, 1, 2, 2, 3, 3], ['Отцы и дети', 'Рудин', 'Дворянское гнездо', 'Толстый и тонкий', 'Дама с собачкой', 'Гроза', 'Таланты и поклонники'], [450, 300, 350, 500, 450, 370, 290].

In [2]:
```
import pandas as pd
```

In [3]:
```
author = pd.DataFrame({'author_id': [1, 2, 3],'author_name': ['Typr
```

In [4]:
```
author
```

Out[4]:

	author_id	author_name
0	1	Тургенев
1	2	Чехов
2	3	Островский

Вопрос преподователю: есть ли разницы в ms (милисекунда) при выполненнии кода строкой (как я сделал в in3) или если я сделал бы создание датафрейма в формате:

author = pd.DataFrame({ 'author_id': [1, 2, 3], 'author_name': ['Тургенев', 'Чехов', 'Островский'], }) author

In [5]:
```
book = pd.DataFrame({
    'author_id': [1, 1, 1, 2, 2, 3, 3],
    'book_title': ['Отцы и дети', 'Рудин', 'Дворянское гнездо', 'To
    'price': [450, 300, 350, 500, 450, 370, 290],
})
```

In [6]:
```
book
```

Out[6]:

	author_id	book_title	price
0	1	Отцы и дети	450
1	1	Рудин	300
2	1	Дворянское гнездо	350
3	2	Толстый и тонкий	500
4	2	Дама с собачкой	450
5	3	Гроза	370
6	3	Таланты и поклонники	290

Задание 2

Получите датафрейм authors_price, соединив датафреймы authors и books по полю author_id.

In [7]:
```
authors_price = pd.merge(author, book, on='author_id', how='outer')
```

In [8]:
```
authors_price
```

Out[8]:

	author_id	author_name	book_title	price
0	1	Тургенев	Отцы и дети	450
1	1	Тургенев	Рудин	300
2	1	Тургенев	Дворянское гнездо	350
3	2	Чехов	Толстый и тонкий	500
4	2	Чехов	Дама с собачкой	450
5	3	Островский	Гроза	370
6	3	Островский	Таланты и поклонники	290

Задание 3

Создайте датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорогими книгами.

In [9]:
```
top5 = authors_price.nlargest(5, 'price')
top5
```

Out[9]:

	author_id	author_name	book_title	price
3	2	Чехов	Толстый и тонкий	500
0	1	Тургенев	Отцы и дети	450
4	2	Чехов	Дама с собачкой	450
5	3	Островский	Гроза	370
2	1	Тургенев	Дворянское гнездо	350

Задание 4

Создайте датафрейм authors_stat на основе информации из authors_price. В датафрейме authors_stat должны быть четыре столбца: author_name, min_price, max_price и mean_price, в которых должны содержаться соответственно имя автора, минимальная, максимальная и средняя цена на книги этого автора.

In [10]:
```
authors_stat = authors_price.groupby('author_name').agg({'price': [
authors_stat
```

Out[10]:

	price		
	min_price	max_price	mean_price
author_name			
Островский	290	370	330.000000
Тургенев	300	450	366.666667
Чехов	450	500	475.000000

In [11]:
```
authors_stat.columns = authors_stat.columns.droplevel(0)
```

In [12]:
```
authors_stat
```

Out[12]:

	min_price	max_price	mean_price
author_name			
Островский	290	370	330.000000
Тургенев	300	450	366.666667
Чехов	450	500	475.000000

In [13]:
```
authors_stat.reset_index(inplace=True)
```

In [14]:
```
authors_stat
```

Out[14]:

	author_name	min_price	max_price	mean_price
0	Островский	290	370	330.000000
1	Тургенев	300	450	366.666667
2	Чехов	450	500	475.000000

Задание 5**

Создайте новый столбец в датафрейме authors_price под названием cover, в нем будут располагаться данные о том, какая обложка у данной книги - твердая или мягкая. В этот столбец поместите данные из следующего списка: ['твердая', 'мягкая', 'мягкая', 'твердая', 'твердая', 'мягкая', 'мягкая']. Просмотрите документацию по функции pd.pivot_table с помощью вопросительного знака.Для каждого автора посчитайте суммарную стоимость книг в твердой и мягкой обложке. Используйте для этого функцию pd.pivot_table. При этом столбцы должны называться "твердая" и "мягкая", а индексами должны быть фамилии авторов. Пропущенные значения стоимостей заполните нулями, при необходимости загрузите библиотеку Numpy. Назовите полученный датасет book_info и сохраните его в формат pickle под названием "book_info.pkl". Затем загрузите из этого файла датафрейм и назовите его book_info2. Удостоверьтесь, что датафреймы book_info и book_info2 идентичны.

In [15]:
```
authors_price['cover'] = ['твердая', 'мягкая', 'мягкая', 'твердая',
authors_price
```

Out[15]:

	author_id	author_name	book_title	price	cover
0	1	Тургенев	Отцы и дети	450	твердая
1	1	Тургенев	Рудин	300	мягкая
2	1	Тургенев	Дворянское гнездо	350	мягкая
3	2	Чехов	Толстый и тонкий	500	твердая
4	2	Чехов	Дама с собачкой	450	твердая
5	3	Островский	Гроза	370	мягкая
6	3	Островский	Таланты и поклонники	290	мягкая

In [16]:
```
?pd.pivot_table
```

In [18]:
```
book_info = authors_price.pivot_table('price', index='author_name',
                                      aggfunc=np.sum, fill_value=0,
                                      )
book_info
```

Out[18]:

cover	мягкая	твердая	All
author_name			
Островский	660	0	660
Тургенев	650	450	1100
Чехов	0	950	950
All	1310	1400	2710

In [19]:
```
pd.to_pickle(book_info, 'book_info.pkl')
```

In [20]:
```
book_info2 = pd.read_pickle('book_info.pkl')
book_info2
```

Out[20]:

cover	мягкая	твердая	All
author_name			
Островский	660	0	660
Тургенев	650	450	1100
Чехов	0	950	950
All	1310	1400	2710

In [21]:
```
book_info.where(book_info.values==book_info2.values).notna()
```

Out[21]:

cover	мягкая	твердая	All
author_name			
Островский	True	True	True
Тургенев	True	True	True
Чехов	True	True	True
All	True	True	True

Сравнил :)

In [ ]:
```

```