Тема «Структуры данных»
Самостоятельная работа
Создание и работа со списками (листами)

Выполните указанные ниже задания в виде отдельного файла Jupyter notebook (в имени файла укажите вашу фамилию и инициалы, а также номер модуля и название данной темы), сохраните его и прикрепите к данному заданию в системе.

В предыдущих шагах мы использовали переменные, содержащие одно значение. Естественно, это крайне неудобно при работе с набором данных. В этом блоке мы рассмотрим одну из основных структур – списки (листы).

Этот тип хранения данных очень удобен из-за простоты фильтрации, вычислений и всевозможных операций с его элементами. Также его легко получать из строк и переводить обратно в текст (например, когда необходимо результат вычислений записать в CSV-файл). По ходу знакомства с листами будем решать следующую задачу:

Дана строчка из CSV-файла, которая содержит набор поисковых запросов, разделенных запятой:

"смотреть сериалы онлайн,новости спорта,афиша кино,курс доллара,сериалы этим летом,курс по питону,сериалы про спорт"

Необходимо отфильтровать этот список запросов, оставив только те, что содержат слово 'сериалы'.

Для хранения нескольких значений используются списки.
Для того, чтобы его создать используется синтаксис вида:
myList = ['один','два','три']
В качестве элементов могут выступать любые типы данных
list_of_some_values = [123, 'питон', True]

Создайте переменную queriesList, куда запишите следующие значения
- смотреть сериалы онлайн
- новости спорта
- смотреть новости
- чемпионат мира по футболу

Что выдаст система на команду print(queriesList)?

С листами очень удобно работать при необходимости фильтрации, преобразования и объединения данных.

Посмотрим основные операции для работы со списками на примере списка значений:

justNumbers = [2,1, 5, 4, 3]

Основные простые операции:

Количество элементов в листе

len(justNumbers)

Что получится в результате?

Сумма элементов листа

sum(justNumbers)

Что получится в результате?

Сортировка элементов листа по возрастанию

sorted(justNumbers)

Что получится в результате?

Если элементы списка представляют собой слова, то сортировка будет происходить по алфавиту

Для сортировки по убыванию добавляют параметр reverse = True

sorted(justNumbers, reverse = True)

[5, 4, 3, 2, 1]

Добавьте элемент к списку

justNumbers.append(55)

Что выведет print(justNumbers)?

Создайте лист из трех слов: 'один', 'два', 'три'

Выведите на экран этот список, отсортированный по алфавиту (используйте следующий формат: ['...', '...', '...'], где вместо многоточия должны стоять элементы списка)

В большинстве случаев данные будут поступать вам не в виде списков, а строк. Например, при чтении таблицы с данными из файла. Если таблица имеет разделитель (например, как запятая в нашей задаче), то для перевода строки в лист можно использовать функцию split. В скобках указываем разделитель элементов (в нашем примере это запятая):

queries_string = "смотреть сериалы онлайн,новости спорта,афиша кино,курс доллара,сериалы этим летом,курс по питону,сериалы про спорт"

Наберите в новой ячейке
print(queries_string.split(','))
Что получится в результате?

Вам дана следующая строка с названиями файлов, которые идут не по порядку: "003_logs_2017-11-03;001_logs_2017-11-01;005_logs_2017-11-05;002_logs_2017-11-02;004_logs_2017-11-04"
Запишите эту строку в переменную fileString
Вам необходимо:
1. Преобразовать эту строку в лист (назовите его fileList)
Напишите код для преобразования

2. Отсортировать список по возрастанию дат (т. е. чтобы первым был файл 001_logs_2017-11-01, а последним - 005_logs_2017-11-05)
Напишите код для преобразования

3. Добавить в лист следующий элемент: 006_logs_2017-11-06

Для перевода списка в строку используется функция join. Желаемый разделитель указываем в кавычках. Например, дан список с поисковыми запросами:
queriesList = ['смотреть сериалы онлайн', 'новости спорта', 'афиша кино', 'курс доллара', 'сериалы этим летом', 'курс по питону', 'сериалы про спорт']
Вы можете перевести лист обратно в строку с запятой в качестве разделителя.
print(','.join(queriesList))

Этот способ очень удобен при записи результатов в файл, особенно, если количество записываемых данных заранее неизвестно. Например, после фильтрации количество оставшихся запросов может быть любым. Если вам необходимо в качестве разделителя использовать знак табуляции (это удобно при дальнейшем экспорте в Excel), то используем служебный знак \t:
print('\t'.join(queriesList))

В прошлом упражнении вы получили список results = ['001_logs_2017-11-01', '002_logs_2017-11-02', '003_logs_2017-11-03', '004_logs_2017-11-04', '005_logs_2017-11-05', '006_logs_2017-11-06']

Преобразуйте этот список в строку, используя в качестве разделителя две вертикальные черты ||.

В работе с данными очень часто возникает необходимость выбирать элементы листа на основе их позиции. Например, второй справа. Или выбрать только четные, начиная с третьего элемента. Для таких операций с листами в питоне есть простые и удобные способы фильтрации.

Возьмем другой пример списка:

sequence = ['Google Adwords', 'Yandex Direct', 'Facebook', 'VK', 'Partner']

Для получения нужного элемента укажите его номер:

sequence[1]

Что будет результатом?

Обратите внимание, что вместо первого по счету элемент 'Google Adwords' мы почему-то получили элемент 'Yandex Direct'. В этом примере видна очень важная особенность нумерации элементов во всех основных языках программирования: номера всех элементов нумеруются от нулевого. Т. е. элемент 'Google Adwords' имеет порядковый номер 0:

sequence[0]

'Google Adwords'

Если мы хотим получить элементы с первого по третий, то пишем так:

sequence[0:3]

Что получится?

Из-за того, что вся нумерация смещается на 1 влево, то Python возьмет все элементы от нулевого до третьего минус 1. Что мы и видим в примере. Такой алгоритм может показаться странным, однако в дальнейшем мы увидим, что он очень удобен при работе с различными интервалами.

В прошлом примере можно не указывать 0, т. е. просто написать sequence[:3]

['Google Adwords', 'Yandex Direct', 'Facebook']

Как получить следующий результат?

['Facebook', 'VK']

Выведем все элементы от третьего по счету до последнего в листе

sequence = ['Google Adwords', 'Yandex Direct', 'Facebook', 'VK', 'Partner']

sequence[2:]

Какой результат вы получите?

Также можно задавать интервал, с которым мы проходимся по списку. Например, для получения только четных чисел из последовательности от 1 до 10 используйте следующий синтаксис:

```
justNumbers = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
justNumbers[ 1::2 ]
```

Что выдаст система?