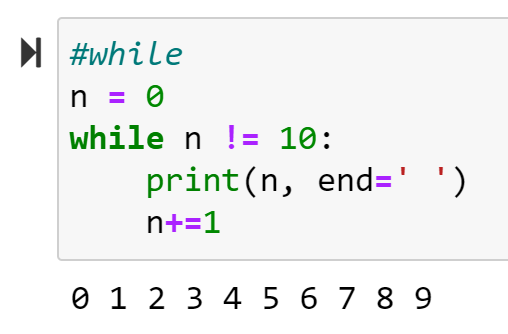
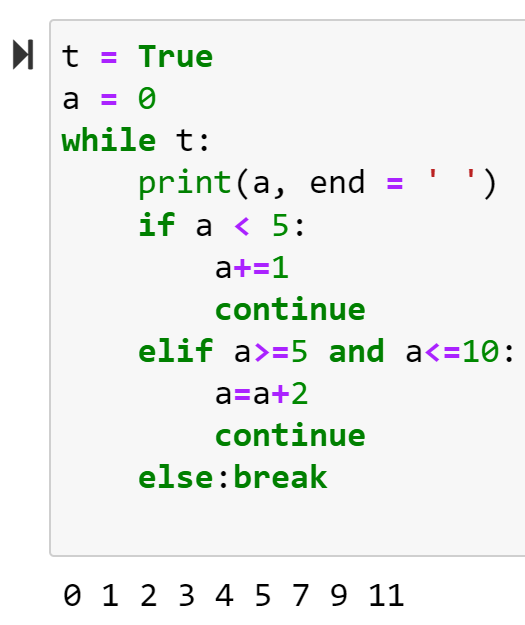
**Циклы, функции Python и работа с библиотеками**

*Цикл while*

Циклический оператор while используется достаточно редко, но все же бывает полезен. Определенный блок кода выполняется, пока выполняется заданное условие. Пример:

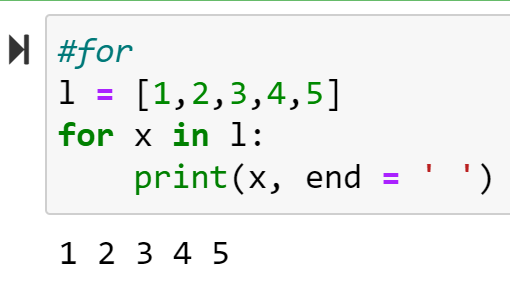


end в выводе позволяет указать, какой знак будет стоять после вывода, по умолчанию – это \n (переход на новую строку). У циклов есть continue и break. Если в цикле используется continue, то цикл будет переходить сразу к следующей итерации, не выполняя код ниже. В случае break – происходит полное прерывание цикла. Пример:

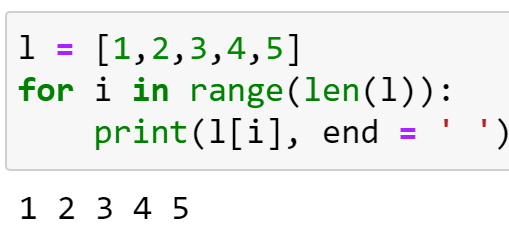


*Цикл for*

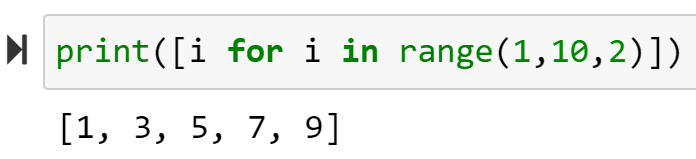
Цикл с использованием ключевого слова for – это прохождение по элементам некой последовательности. Например, необходимо вывести элементы, хранящиеся в списке:



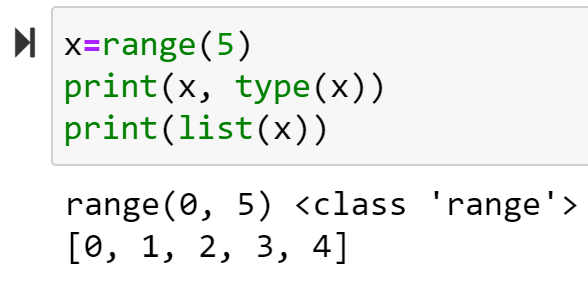
Второй способ – это вывод элементов списка по их индексу:



range – это генератор чисел, в данном случае он генерирует индексы от 0 до длины списка-1, то есть 0,1,2,3,4, затем вывод происходит через обращение к элементу списка по индексу. Примеры использования генератора:



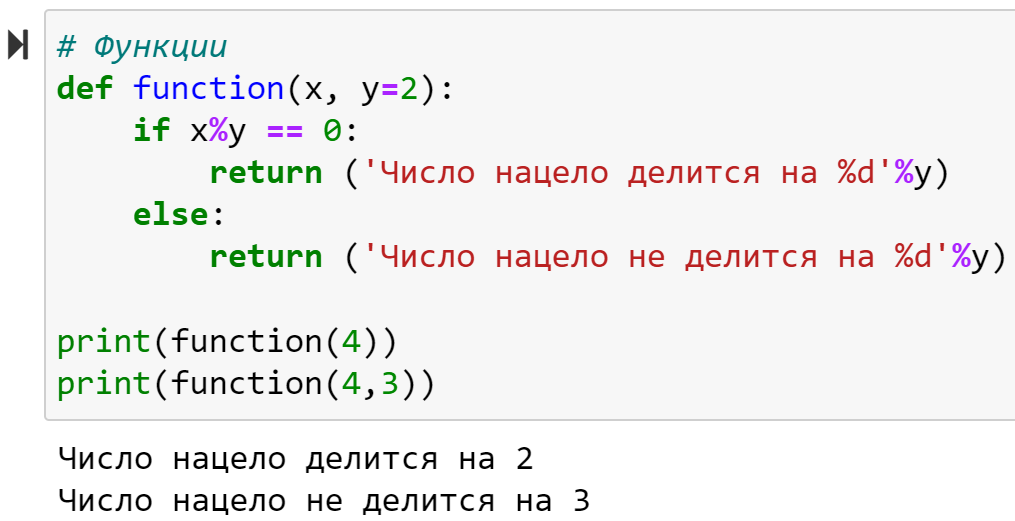
range(start, stop, step). start – начало, stop – конечное значение (обратите внимание, что stop не входит в интервал генерации чисел), step – шаг.



*Функции*

Функция – блок кода, который выполняется при его вызове в ходе выполнения программы.

Функция в Python начинается с ключевого слова def, далее идет имя функции и (). В () указываются аргументы функции. Сначала идут аргументы, который не даны по умолчанию, затем аргументы, которые даны по умолчанию. Тело функции выделяется отступом. Пример:

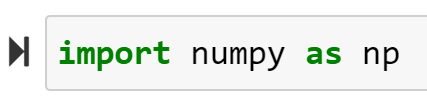


Ключевое слово return – это результат, который возвращает функция.

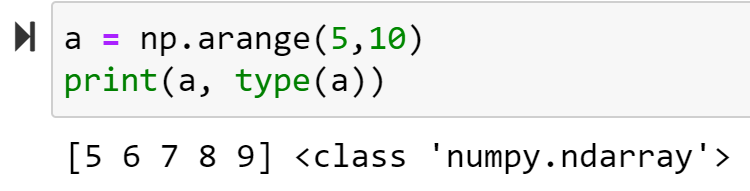
*Библиотека numpy*

numpy это библиотека для работы с матрицами и векторами.

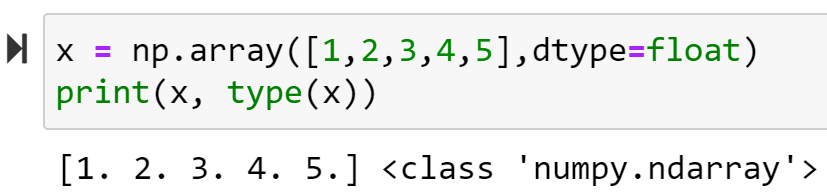
Для работы с библиотекой ее необходимо импортировать с помощью команды import.



Генерация массива в numpy:

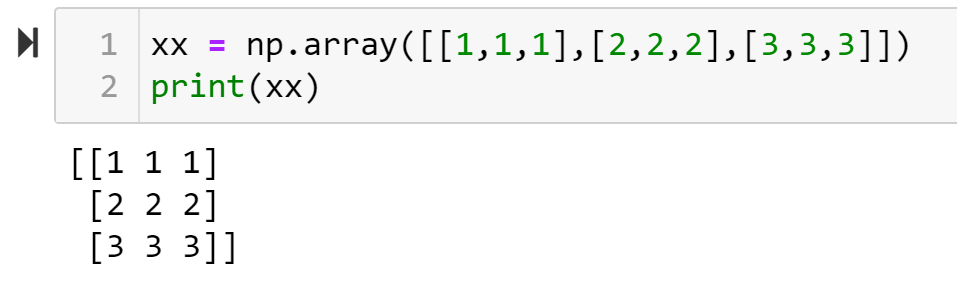


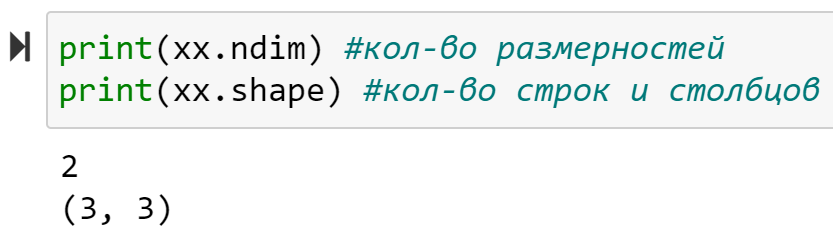
Можно привести список к numpy массиву:

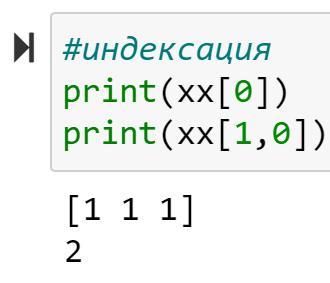


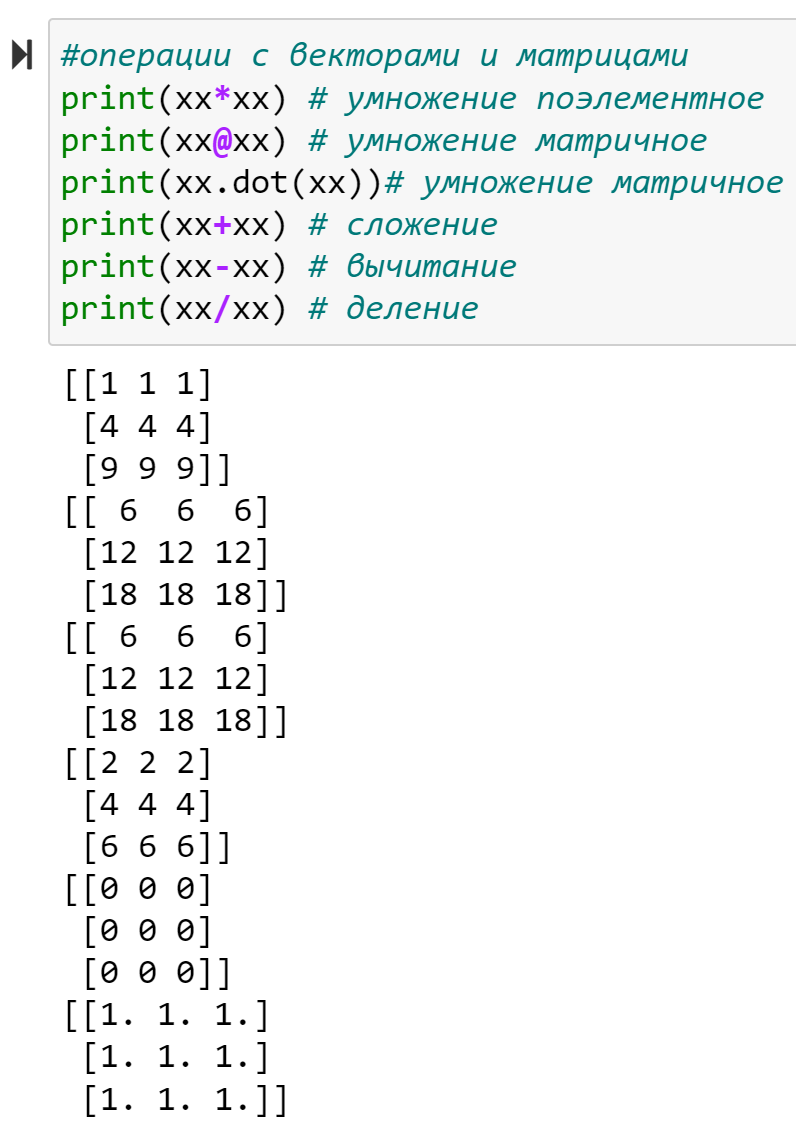
dtype позволяет указать тип хранимых элементов, в данном случае – это вещественные числа.

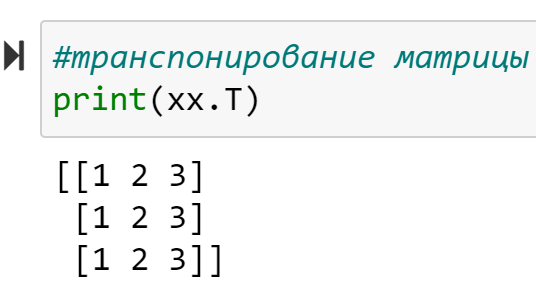
Двумерные (и не только) массивы:



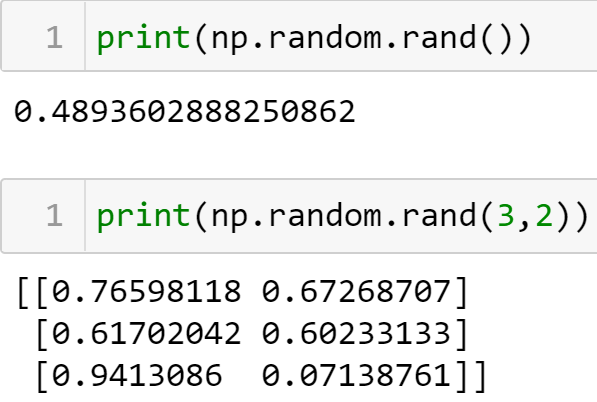








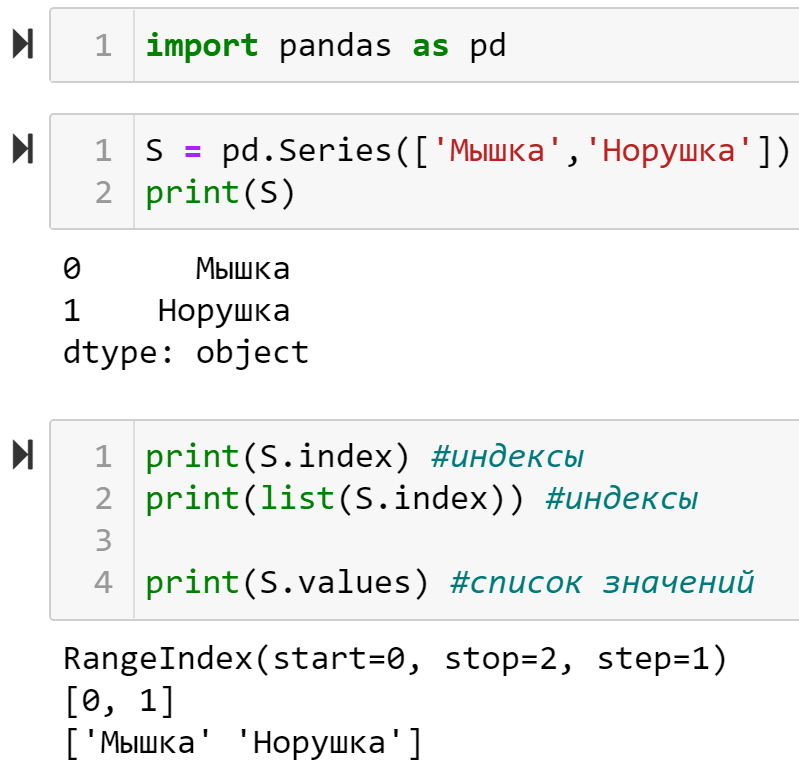
np.random.rand() – генерация случайных значений из равномерного распределения от 0 до 1.

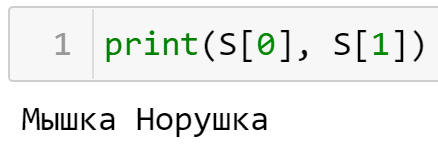


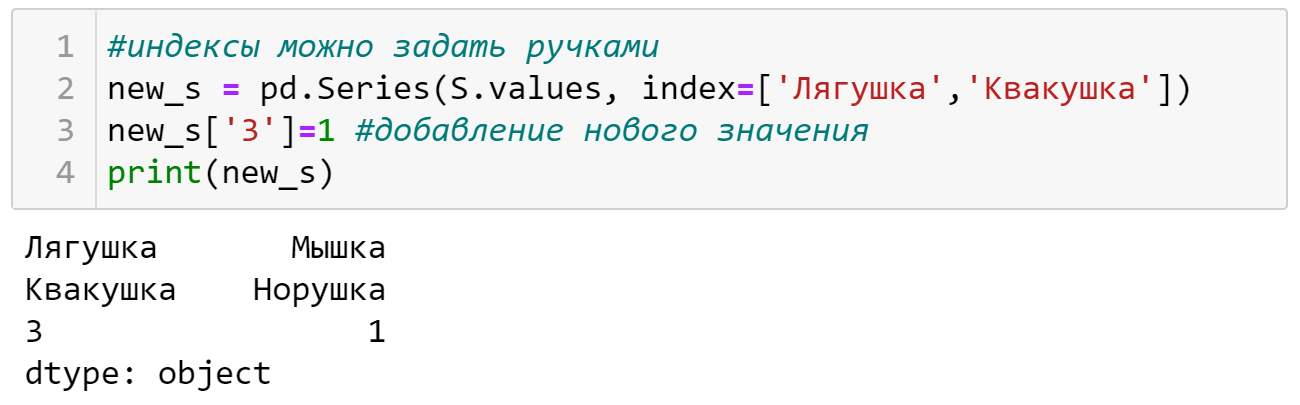
np.random.rand() генерация случайных значений из нормального распределения с дисперсией, равной 1, и средним, равным 0.

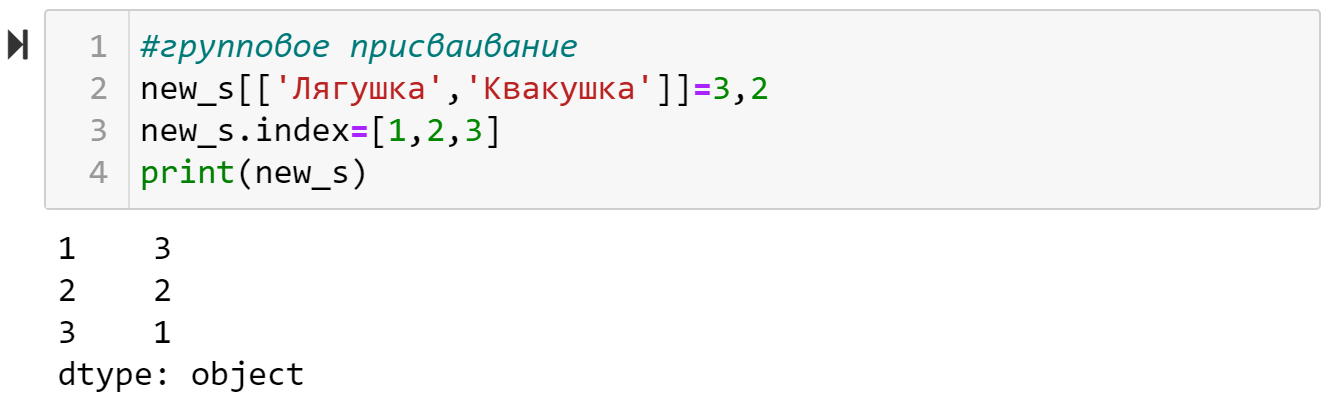
*Библиотека pandas*

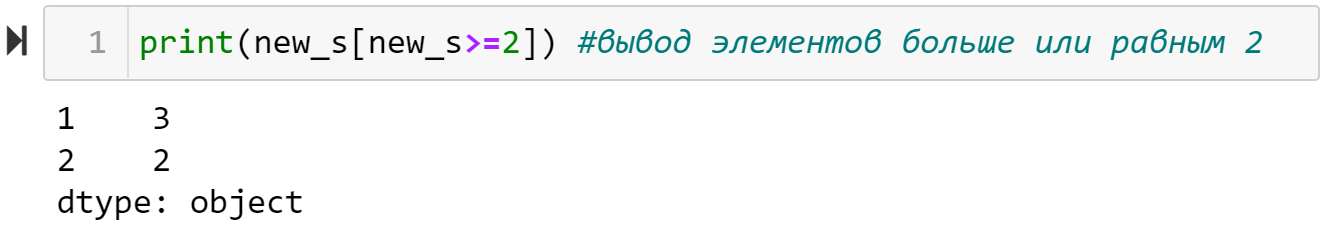
Pandas – это библиотека для обработки и анализа данных. В pandas есть две структуры объекта: DataFrame и Series. Series – это аналог одномерного массива, который имеет индексы, по сути, это ассоциативный массив, как словарь. DataFrame имеет табличную структуру, столбцы у DataFrame – это Series.





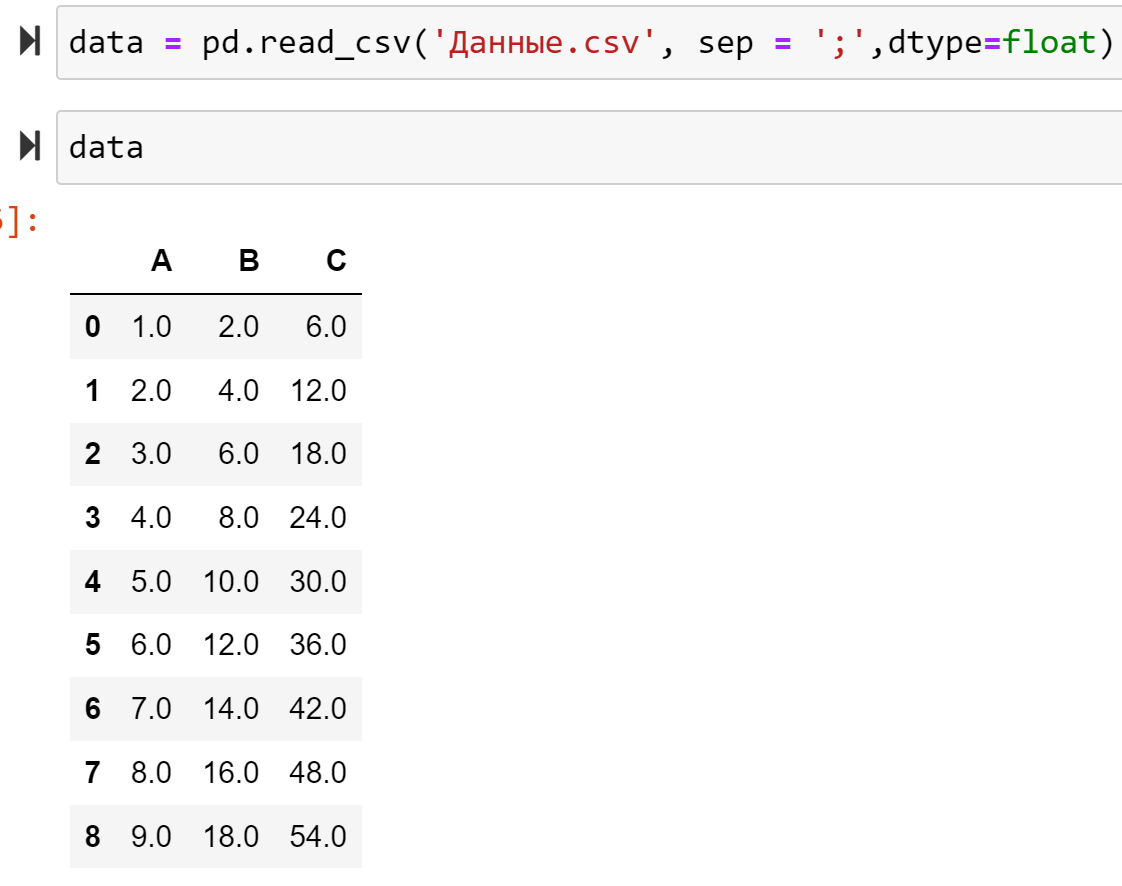






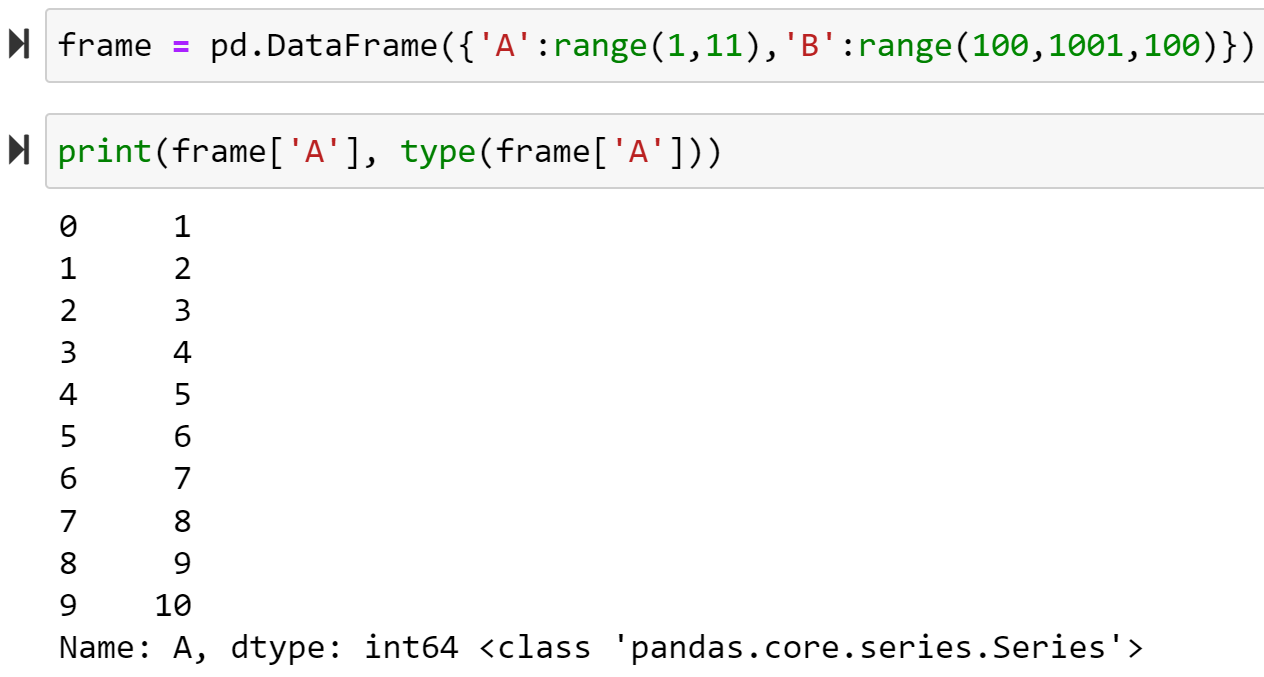
**DataFrame**

Загрузка данных из файла .csv в DataFrame.

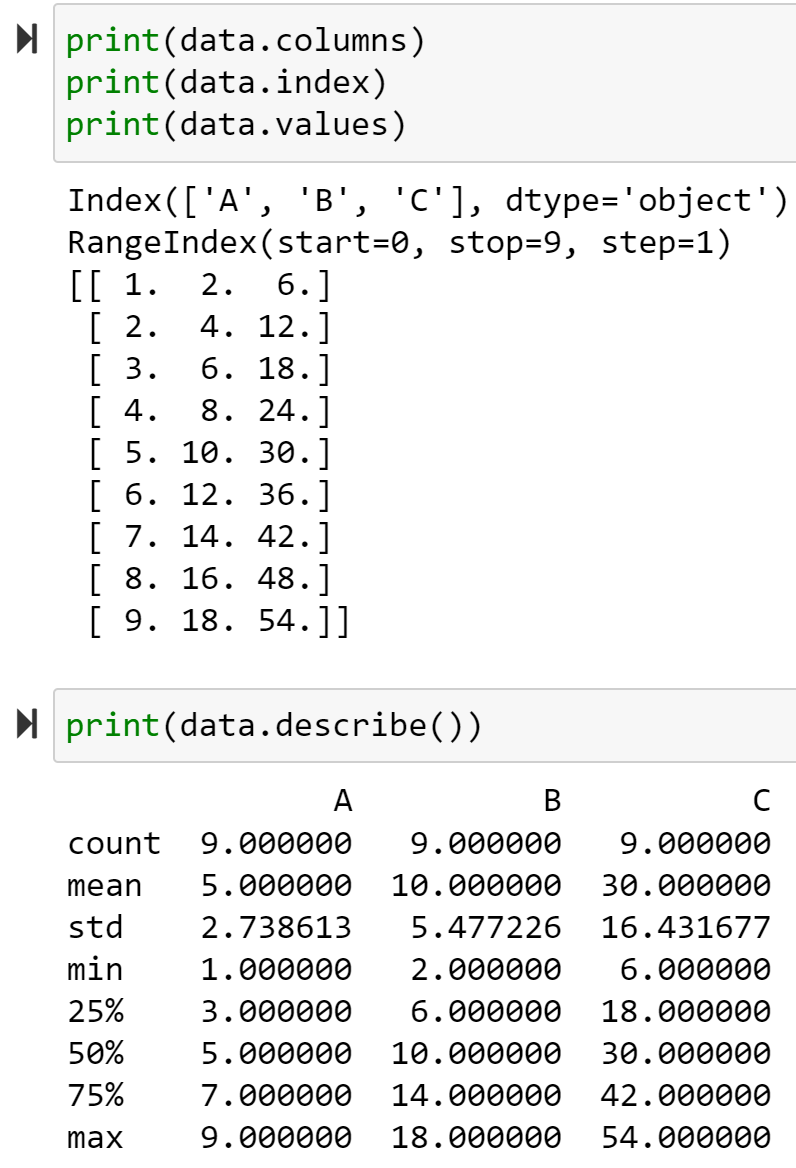


Аргумент sep позволяет указать разделитель. Также доступно чтение из txt (read\_txt) файла и xlsx файла (read\_excel).

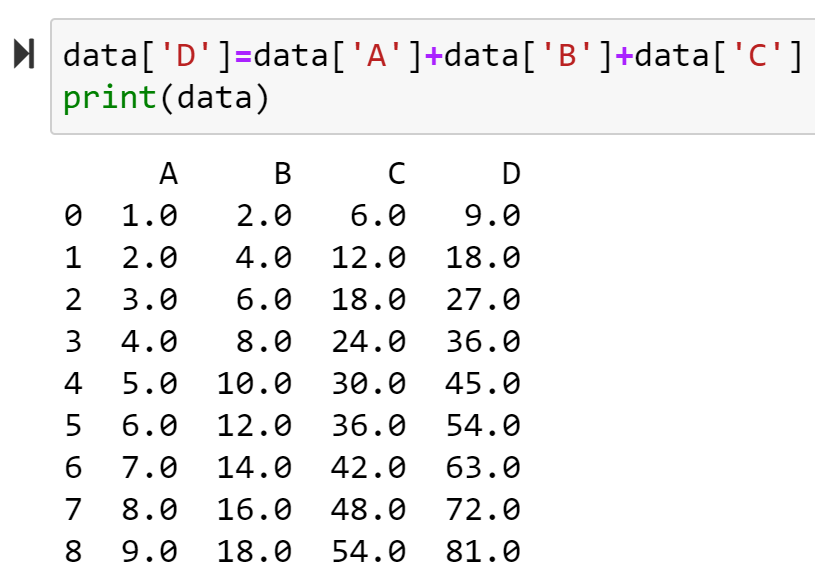
Задать столбцы и значения можно вручную:

****

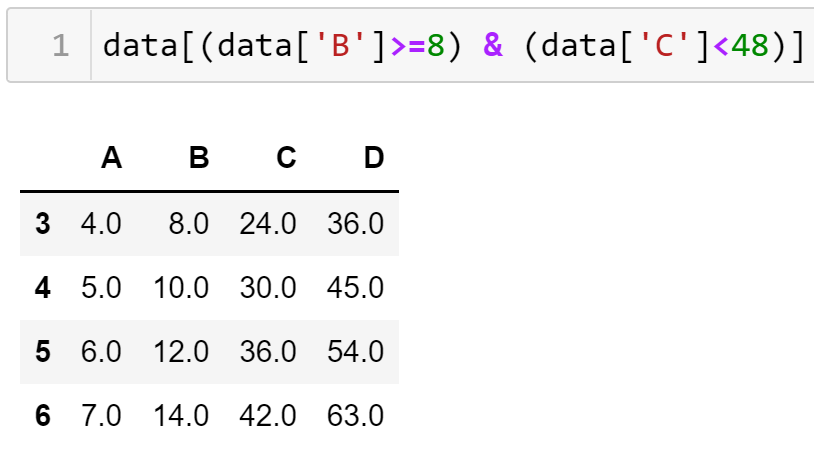
Полезным является метод describe(), который позволяет узнать основные статистические показатели по столбцам датафрейма.

****

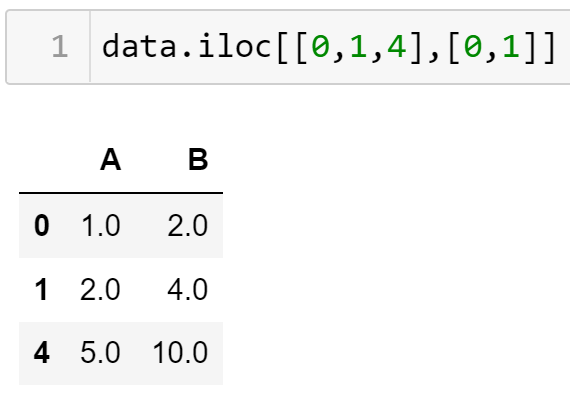
Можно создать новый столбец, используя известные данные:



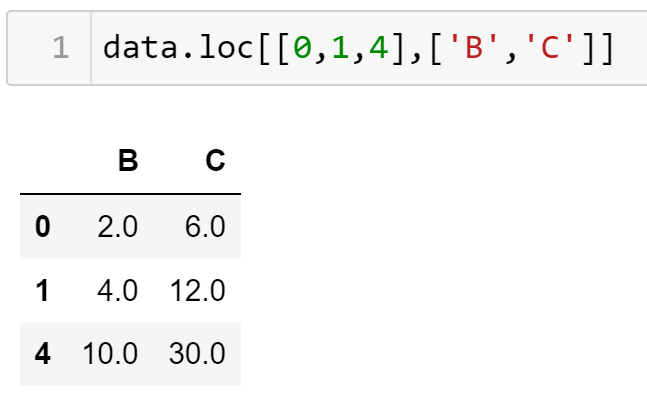
Значения в DataFrame или Series можно отфильтровать:



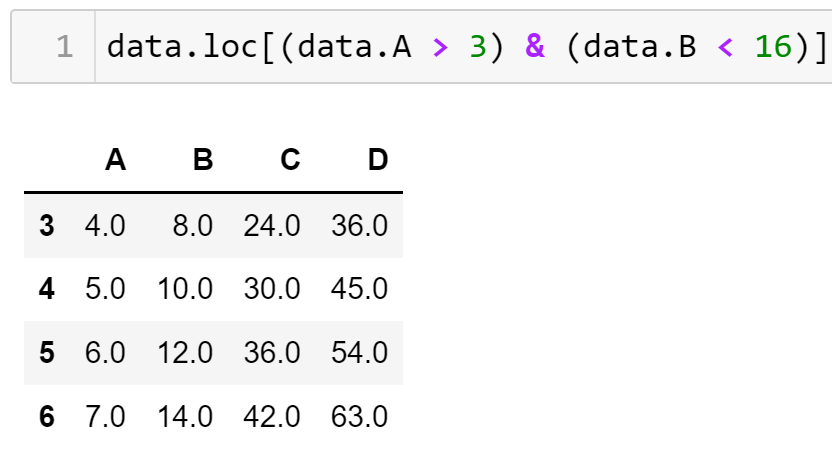
Обращение к датафрейму, метод iloc – индексация по позиции:



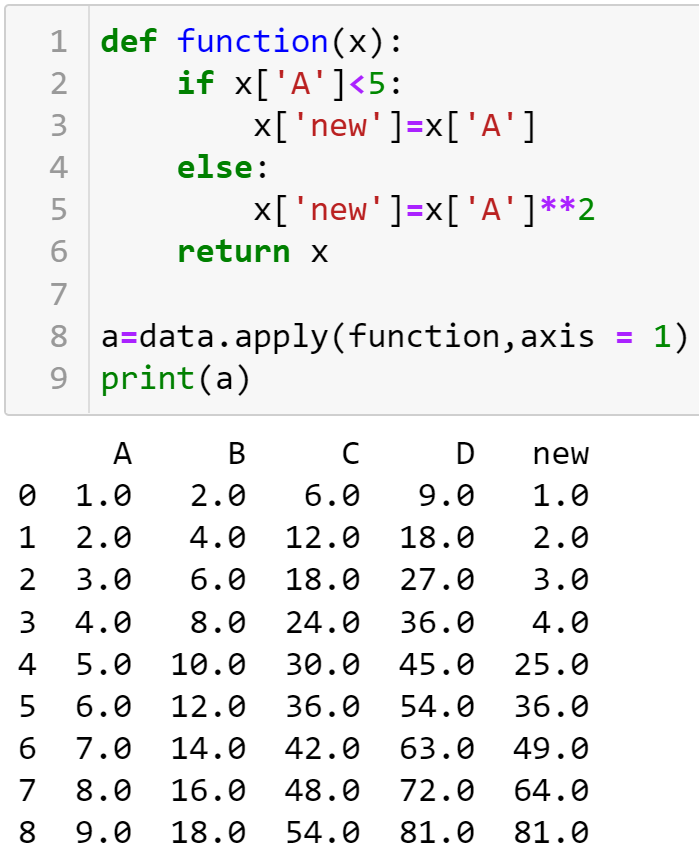
Метод loc – индексация на основе имени:



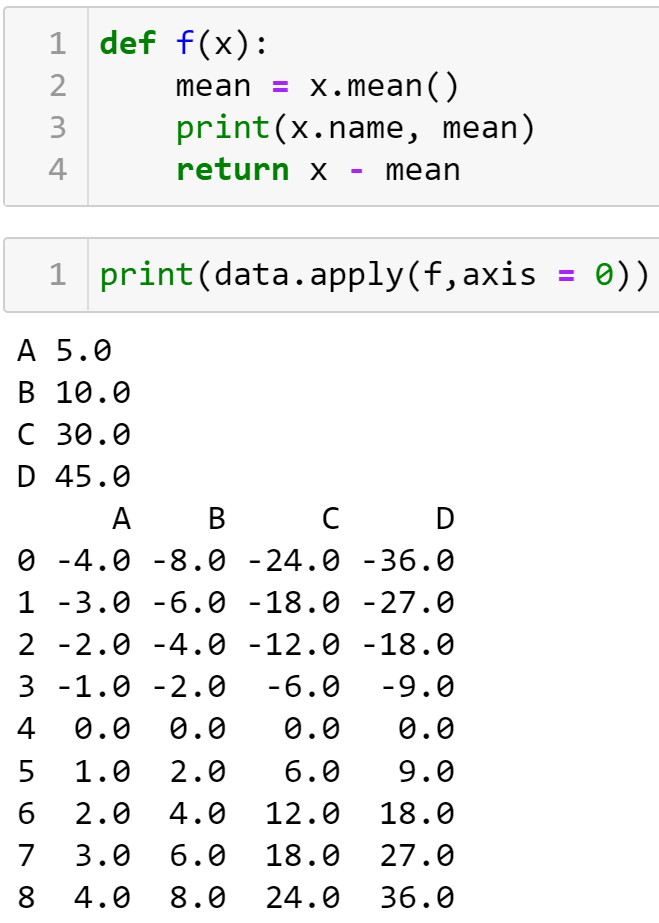
Так же с помощью loc можно фильтровать данные, например, выведем данные, где значение признака А > 3 и значение признака В < 16:



Метод apply() позволяет применить к серии или датафрейму некоторую функцию, например, создание нового признака, если значение в столбце A меньше 5 или больше:



axis = 1 – это проход по строкам, axis = 0 – это проход по столбцам, например:

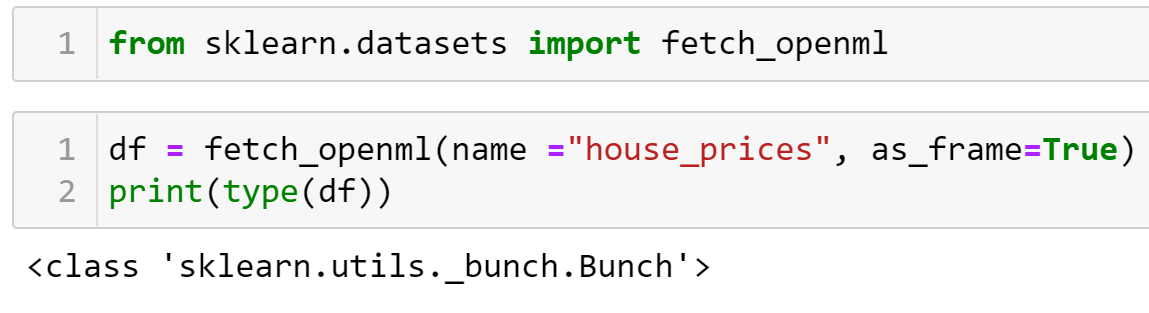


*Библиотека scikit-learn.*

scikit-learn – это библиотека для эффективного предиктивного анализа данных. Она содержит в себе различные алгоритмы машинного обучения, инструменты для предобработки данных, готовые наборы данных для исследований и тд.

Готовые датасеты можно взять в модуле dataset. У каждого датасета есть функция для его загрузки.

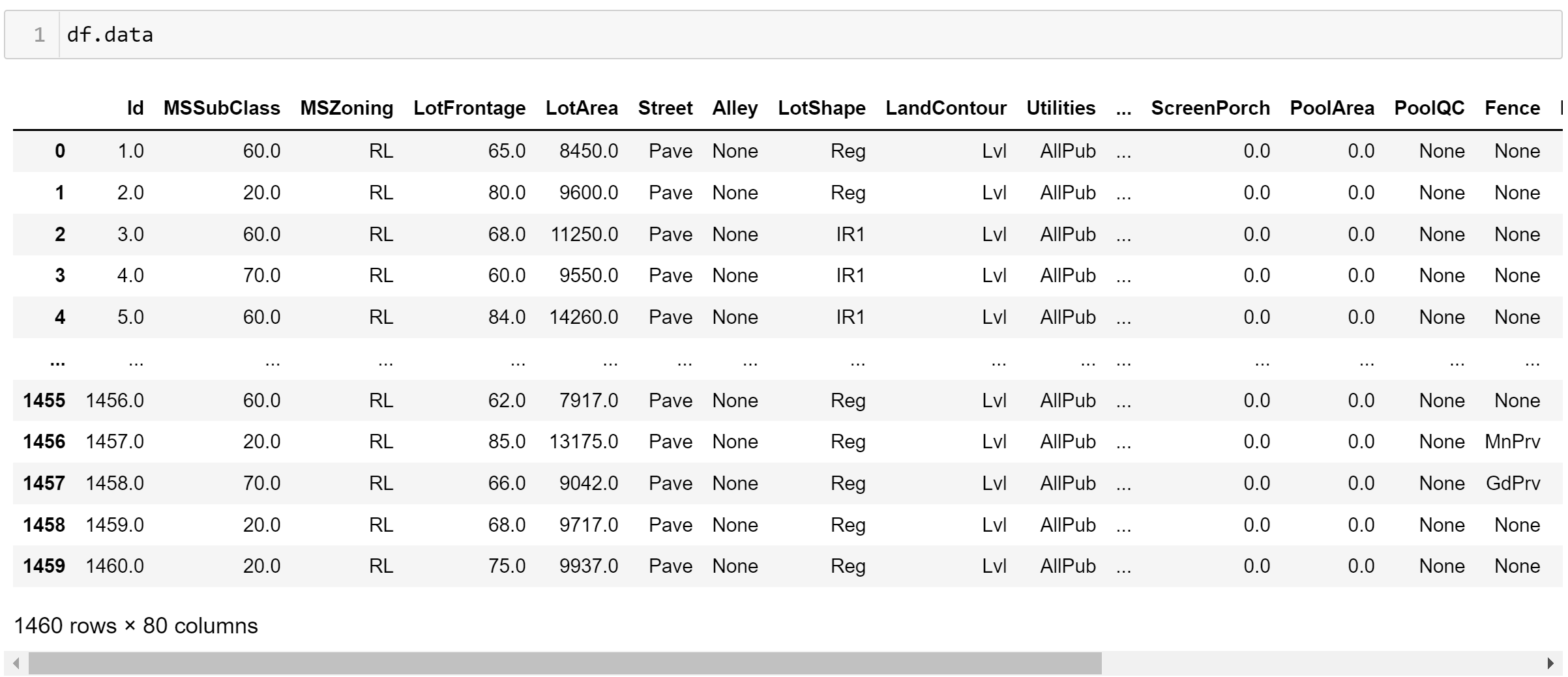
Рассмотрим пример загрузки датасета Ames housing:



В результате df – это объект Bunch, который представляет собой аналог словаря в библиотеке Scikit-Learn. Соответственно, у него есть ключи.



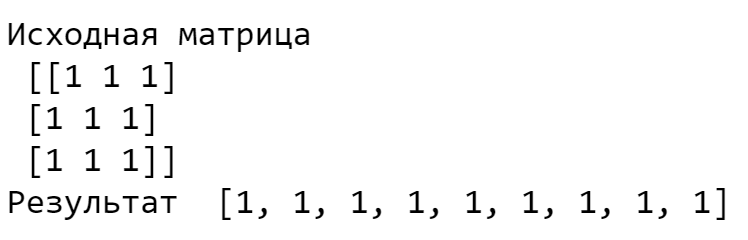
data – это таблица признаков. target – это значение целевой переменной. Этот набор данных используется для прогнозирования стоимости дома.



Подробнее данную библиотеку рассмотрим в следующих практических работах.

**Практическая работа**

1. Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.
2. Напишите программу, которая выводит последовательность чисел, длинною N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – print(\*list).
3. Матрицу произвольного размера вытянуть в один вектор, не применяя встроенные методы Python. Пример:



Для создания матрицы можно использовать np.random.rand(кол-во строк, кол-во столбцов).

1. Даны два списка:

А = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2]

В = [‘a’, ’b’, ’c’, ’c’, ’c’, ’b’, ’a’, ’c’, ’a’, ’a’, ’b’, ’c’, ’b’, ’a’]

Создать словарь, в котором ключи – это содержимое списка В, а значения для ключей словаря – это сумма всех элементов списка А в соответствии с буквой, содержащийся на той же позиции в списке В. Пример результата программы: {‘a’ : 10, ‘b’ : 15, ‘c’ : 6}.

1. Скачать и загрузить данные о стоимости домов в калифорнии, используя библиотеку sklearn.



1. Используя метод pd.concat([датафрейм1, датафрейм2], axis = 1), добавить к данным столбец, содержащий информацию о медианной стоимости дома (.target и .target\_names).
2. Использовать метод info().
3. Узнать, есть ли пропущенные значения, используя isna().sum().
4. Вывести записи, где средний возраст домов в районе более 50 лет и население более 2500 человек, используя метод loc().
5. Узнать максимальное и минимальное значения медианной стоимости дома (max(), min()).
6. Используя метод apply(), вывести на экран название признака и его среднее значение.
7. Составить отчет о проделанной работе. В отчете должен быть представлен код и результаты его выполнения с выводами.