**Чтение и запись файлов Excel (XLSX) в Python**

Pandas можно использовать для чтения и записи файлов Excel с помощью Python. Это работает по аналогии с другими форматами. В этом материале рассмотрим, как это делается с помощью DataFrame.

Помимо чтения и записи рассмотрим, как записывать несколько DataFrame в Excel-файл, как считывать определенные строки и колонки из таблицы и как задавать имена для одной или нескольких таблиц в файле.

**Установка Pandas**

Для начала Pandas нужно установить. Проще всего это сделать с помощью pip.

Если у вас Windows, Linux или macOS:

pip install pandas # или pip3

В процессе можно столкнуться с ошибками ModuleNotFoundErrorили ImportError при попытке запустить этот код. Например:

ModuleNotFoundError: No module named 'openpyxl'

В таком случае нужно установить недостающие модули:

pip install openpyxl xlsxwriter xlrd # или pip3

**Запись в файл Excel с python**

Будем хранить информацию, которую нужно записать в файл Excel, в DataFrame. А с помощью встроенной функции to\_excel()ее можно будет записать в Excel.

Сначала импортируем модуль pandas. Потом используем словарь для заполнения DataFrame:

import pandas as pd

df = pd.DataFrame({'Name': ['Manchester City', 'Real Madrid', 'Liverpool',

'FC Bayern München', 'FC Barcelona', 'Juventus'],

'League': ['English Premier League (1)', 'Spain Primera Division (1)',

'English Premier League (1)', 'German 1. Bundesliga (1)',

'Spain Primera Division (1)', 'Italian Serie A (1)'],

'TransferBudget': [176000000, 188500000, 90000000,

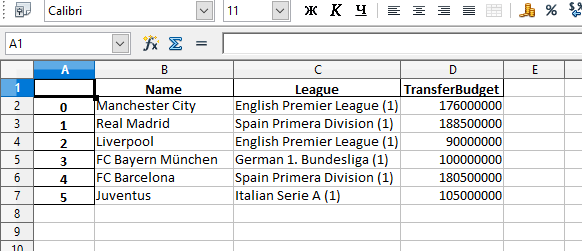
100000000, 180500000, 105000000]})

Ключи в словаре — это названия колонок. А значения станут строками с информацией.

Теперь можно использовать функцию to\_excel() для записи содержимого в файл. Единственный аргумент — это путь к файлу:

df.to\_excel('./teams.xlsx')

А вот и созданный файл Excel:

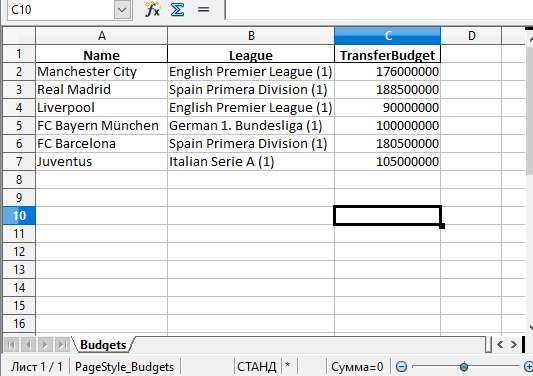


Стоит обратить внимание на то, что в этом примере не использовались параметры. Таким образом название листа в файле останется по умолчанию — «Sheet1». В файле может быть и дополнительная колонка с числами. Эти числа представляют собой индексы, которые взяты напрямую из DataFrame.

Поменять название листа можно, добавив параметр sheet\_name в вызов to\_excel():

df.to\_excel('./teams.xlsx', sheet\_name='Budgets', index=False)

Также можно добавили параметр index со значением False, чтобы избавиться от колонки с индексами. Теперь файл Excel будет выглядеть следующим образом:



**Запись нескольких DataFrame в файл Excel**

Также есть возможность записать несколько DataFrame в файл Excel. Для этого можно указать отдельный лист для каждого объекта:

salaries1 = pd.DataFrame({'Name': ['L. Messi', 'Cristiano Ronaldo', 'J. Oblak'],

'Salary': [560000, 220000, 125000]})

salaries2 = pd.DataFrame({'Name': ['K. De Bruyne', 'Neymar Jr', 'R. Lewandowski'],

'Salary': [370000, 270000, 240000]})

salaries3 = pd.DataFrame({'Name': ['Alisson', 'M. ter Stegen', 'M. Salah'],

'Salary': [160000, 260000, 250000]})

salary\_sheets = {'Group1': salaries1, 'Group2': salaries2, 'Group3': salaries3}

writer = pd.ExcelWriter('./salaries.xlsx', engine='xlsxwriter')

for sheet\_name in salary\_sheets.keys():

salary\_sheets[sheet\_name].to\_excel(writer, sheet\_name=sheet\_name, index=False)

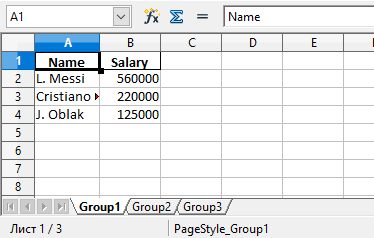
writer.save()

Здесь создаются 3 разных DataFrame с разными названиями, которые включают имена сотрудников, а также размер их зарплаты. Каждый объект заполняется соответствующим словарем.

Объединим все три в переменной salary\_sheets, где каждый ключ будет названием листа, а значение — объектом DataFrame.

Дальше используем движок xlsxwriterдля создания объекта writer. Он и передается функции to\_excel().

Перед записью пройдемся по ключам salary\_sheets и для каждого ключа запишем содержимое в лист с соответствующим именем. Вот сгенерированный файл:



Можно увидеть, что в этом файле Excel есть три листа: Group1*,*Group2 и Group3. Каждый из этих листов содержит имена сотрудников и их зарплаты в соответствии с данными в трех DataFrame из кода.

Параметр движка в функции to\_excel() используется для определения модуля, который задействуется библиотекой Pandas для создания файла Excel. В этом случае использовался xslswriter, который нужен для работы с классом ExcelWriter. Разные движка можно определять в соответствии с их функциями.

В зависимости от установленных в системе модулей Python другими параметрами для движка могут быть openpyxl*(*для xlsx или xlsm) и xlwt (для xls).

Наконец, в коде была строка writer.save(), которая нужна для сохранения файла на диске.

**Чтение файлов Excel с python**

По аналогии с записью объектов DataFrame в файл Excel, эти файлы можно и читать, сохраняя данные в объект DataFrame. Для этого достаточно воспользоваться функцией read\_excel():

top\_players = pd.read\_excel('./top\_players.xlsx')

top\_players.head()

Содержимое финального объекта можно посмотреть с помощью функции head().

**Примечание:**

*Этот способ самый простой, но он и способен прочесть лишь содержимое первого листа.*

Посмотрим на вывод функции head():

|  | **Name** | **Age** | **Overall** | **Potential** | **Positions** | **Club** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | L. Messi | 33 | 93 | 93 | RW,ST,CF | FC Barcelona |
| **1** | Cristiano Ronaldo | 35 | 92 | 92 | ST,LW | Juventus |
| **2** | J. Oblak | 27 | 91 | 93 | GK | Atlético Madrid |
| **3** | K. De Bruyne | 29 | 91 | 91 | CAM,CM | Manchester City |
| **4** | Neymar Jr | 28 | 91 | 91 | LW,CAM | Paris Saint-Germain |

Pandas присваивает метку строки или числовой индекс объекту DataFrame по умолчанию при использовании функции read\_excel().

Это поведение можно переписать, передав одну из колонок из файла в качестве параметра index\_col:

top\_players = pd.read\_excel('./top\_players.xlsx', index\_col='Name')

top\_players.head()

Результат будет следующим:

| **Name** | **Age** | **Overall** | **Potential** | **Positions** | **Club** |
| --- | --- | --- | --- | --- | --- |
| L. Messi | 33 | 93 | 93 | RW,ST,CF | FC Barcelona |
| Cristiano Ronaldo | 35 | 92 | 92 | ST,LW | Juventus |
| J. Oblak | 27 | 91 | 93 | GK | Atlético Madrid |
| K. De Bruyne | 29 | 91 | 91 | CAM,CM | Manchester City |
| Neymar Jr | 28 | 91 | 91 | LW,CAM | Paris Saint-Germain |

В этом примере индекс по умолчанию был заменен на колонку «Name» из файла. Однако этот способ стоит использовать только при наличии колонки со значениями, которые могут стать заменой для индексов.

**Чтение определенных колонок из файла Excel**

Иногда удобно прочитать содержимое файла целиком, но бывают случаи, когда требуется получить доступ к определенному элементу. Например, нужно считать значение элемента и присвоить его полю объекта.

Это делается с помощью функции read\_excel() и параметра usecols. Например, можно ограничить функцию, чтобы она читала только определенные колонки. Добавим параметр, чтобы он читал колонки, которые соответствуют значениям «Name», «Overall» и «Potential».

Для этого укажем числовой индекс каждой колонки:

КОПИРОВАТЬ

cols = [0, 2, 3]

top\_players = pd.read\_excel('./top\_players.xlsx', usecols=cols)

top\_players.head()

Вот что выдаст этот код:

|  | **Name** | **Overall** | **Potential** |
| --- | --- | --- | --- |
| **0** | L. Messi | 93 | 93 |
| **1** | Cristiano Ronaldo | 92 | 92 |
| **2** | J. Oblak | 91 | 93 |
| **3** | K. De Bruyne | 91 | 91 |
| **4** | Neymar Jr | 91 | 91 |

Таким образом возвращаются лишь колонки из списка cols.

В DataFrame много встроенных возможностей. Легко изменять, добавлять и агрегировать данные. Даже можно строить сводные таблицы. И все это сохраняется в Excel одной строкой кода.