

[Главная](#) [Уроки](#) [Чтение и запись файлов Excel \(XLSX\) в Python](#)

# 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 `ModuleNotFoundError` или `ImportError` при попытке запустить этот код. Например:

```
ModuleNotFoundError: No module named 'openpyxl'
```

В таком случае нужно установить недостающие модули:

```
pip install openpyxl xlswriter xlrd # или pip3
```

## Запись в файл Excel с python

[КОДЕ](#)[Установка Pandas](#)[Запись в файл Excel](#)[Запись нескольких файлов Excel](#)[Чтение файлов Excel](#)[Чтение определенных файлов Excel](#)[Выводы](#) [Пройди тест и получи](#)[НОВО](#)[Нахождение делителей](#)[Лямбда-функции и анон](#)[Когда стоит использовать](#)[Как извлечь кубический](#)

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`:

КОПИРОВАТЬ

```
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 [
                           'English Premier League (1)', 'German 1. Bunde
                           'Spain Primera Division (1)', 'Italian Serie A
                  '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:

КОПИРОВАТЬ

```
df.to_excel('./teams.xlsx')
```

РЕКЛАМА

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()` :

[КОПИРОВАТЬ](#)

```
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:

КОПИРОВАТЬ

```
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()
```

РЕКЛАМА

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 `xlsxwriter` для 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 `xlsxwriter`, 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`). Подробности о модуле `xlsxwriter` можно найти в [официальной документации](#).

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()`:

КОПИРОВАТЬ

```
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`:

КОПИРОВАТЬ

```
top_players = pd.read_excel('./top_players.xlsx', index_col='Name')
top_players.head()
```

Результат будет следующим:

РЕКЛАМА

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:

	<b>Name</b>	<b>Overall</b>	<b>Potential</b>
<b>0</b>	L. Messi	93	93
<b>1</b>	Cristiano Ronaldo	92	92
<b>2</b>	J. Oblak	91	93
<b>3</b>	K. De Bruyne	91	91
<b>4</b>	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.

Рекомендую изучить `DataFrame` в моих [уроках по Pandas](#).

## Выводы

В этом материале были рассмотрены функции `read_excel()` и `to_excel()` из библиотеки `Pandas`. С их помощью можно считывать данные из файлов Excel и выполнять запись в них. С помощью различных параметров есть возможность менять поведение функций, создавая нужные файлы, не просто копируя содержимое из объекта `DataFrame`.

## Обучение с трудоустройством

| Программирование

Python-разработчик

→

145 440 Р

4 692 Р/мес.

| Data Science

Data Scientist

→

172 800 Р

5 574 Р/мес.

| Программирование

Fullstack-разработчик на Python

→

145 440 Р

4 692 Р/мес.

| Data Science

Data Scientist: анализ данных

→

95 040 Р

3 960 Р/мес.

теги

pandas

Facebook

Telegram

Twitter

VK

Максим

Я создал этот блог в 2018 году, чтобы распространять полезные учебные материалы, документации и уроки на русском. На сайте опубликовано множество статей по основам python и библиотекам, уроков для начинающих и примеров написания программ.  
Мои контакты: [Почта](#)

Статьи по теме

Запуск Django-приложения в Docker контейнере

Алгоритм классификации Random Forest на Python

Алгоритмы K-ближайших соседей и K-средних на Python

Полное руководство по линейной регрессии в Scikit-Learn

Создаем API блога на Django REST Framework

Настройка и подключение статических файлов в Django



