

ХОЧУ ПОМОЧЬ ПРОЕКТУ

gb.ru

РЕКЛАМА · 16+

Бесплатный практикум для детей: Python и анимация

5,0 ★ Рейтинг организации ⓘ

Узнать больше

skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустроитесь Junior-специалистом.

Налоговый вычет 13% >

Спикеры-практики >

Трудоустройство >

Учитесь сейчас, платите потом >

Узнать больше

Работа с файлами XLSX при помощи модуля openpyxl

интуитивно понятный и удобный способ манипулирования большими наборами данных без какой-либо подготовки. По этому, это один из форматов, с которым, в какой-то момент времени, вам придется столкнуться. Решить стоящие задачи по извлечению каких-то данных из базы данных или файла логов в электронную таблицу Excel в какую-либо более удобную программную форму, с помощью openpyxl. openpyxl — это библиотека Python для чтения/записи форматов Office Open XML (файлов Excel 2010) с расширениями XLSX, XLSM, XLTM, XLTML. openpyxl работает с файлами формата Excel 2010 и новее, поэтому установка относительно проста.

если нет Python 3.7+ установить VirtualEnv

```
# активируем виртуальное окружение$ source .venv/bin/activate# ставим модуль openpyxl(VirtualEnv):~$ python3 -m pip install -U openpyxl
```

Основы работы с файлами Microsoft Excel на Python.

- Создание книги Excel.
 - Новый рабочий лист книги Excel.
 - Копирование рабочего листа книги Excel.
 - Удаление рабочего листа книги Excel.
- Доступ к ячейке электронной таблицы и ее значению.
- Доступ к диапазону ячеек листа электронной таблицы.
- Получение только значений ячеек листа.
- Добавление данных в ячейки списком.
- Сохранение созданной книги в файл Excel.
 - Сохранение данных книги в виде потока.
- Загрузка документа XLSX из файла.

Создание книги Excel.

Чтобы начать работу с модулем openpyxl, нет необходимости создавать файл электронной таблицы в файловой системе. Нужно просто импортировать класс Workbook и создать его экземпляр. Рабочая книга всегда создается как минимум с одним рабочим листом, его можно получить, используя свойство Workbook.active:

```
>>> from openpyxl import Workbook
# создаем книгу
>>> wb = Workbook()
# делаем единственный лист активным
>>> ws = wb.active
```

Новый рабочий лист книги Excel.

Новые рабочие листы можно создавать, используя метод Workbook.create_sheet():

```
# Вверх
>>> wb.create_sheet("Mysheet")
```

```
# вставить рабочий лист в первую позицию
>>> ws2 = wb.create_sheet("Mysheet", 0)
# вставить рабочий лист в предпоследнюю позицию
>>> ws3 = wb.create_sheet("Mysheet", -1)
```



S skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустройтесь Junior-специалистом.

- Налоговый вычет 13% >
- Спикеры-практики >
- Трудоустройство >
- Учитесь сейчас, платите потом >

Узнать больше

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Ссылка на объявление: [https://skillbox.ru/course/devops-engineer/](#)

Они имеют имя при создании. Они нумеруются последовательно (Sheet, Sheet1, Sheet2, ...). Эти листы можно назвать с помощью свойства Worksheet.title:

По умолчанию лист имеет цвет по умолчанию белый. Можно изменить этот цвет, указав цветовой код RRGGBB для атрибута Worksheet.tabColor:

```
ws2.tabColor = "1072BA"
```

Можно использовать имя листа, используя его имя в качестве ключа экземпляра созданной книги Excel:

Для копирования рабочего листа книги, необходимо использовать атрибут Workbook.sheetname. Также можно использовать имя листа в качестве ключа экземпляра созданной книги Excel.

```
wb.copy_worksheet(ws2, 'Mysheet')
```

Копирование рабочего листа книги Excel.

Для создания копии рабочих листов в одной книге, необходимо воспользоваться методом Workbook.copy_worksheet():

```
>>> source_page = wb.active
>>> target_page = wb.copy_worksheet(source_page)
```

Примечание. Копируются только ячейки (значения, стили, гиперссылки и комментарии) и определенные атрибуты рабочего листа (размеры, формат и свойства). Все остальные атрибуты книги/листа не копируются, например, изображения или диаграммы.

Поддерживается возможность копирования рабочих листов между книгами. Нельзя скопировать рабочий лист, если рабочая книга открыта в [режиме только для чтения](#) или только для записи.

Удаление рабочего листа книги Excel.

Очевидно, что встает необходимость удалить лист электронной таблицы, который уже существует. Модуль openpyxl дает возможность удалить лист по его имени. Следовательно, сначала необходимо выяснить, какие листы присутствуют в книге, а потом удалить ненужный. За удаление листов книги отвечает метод Workbook.remove().

Смотрим пример:

```
# выясним, названия листов присутствуют в книге
>>> name_list = wb.sheetnames
>>> name_list
# ['Mysheet1', 'NewPage', 'Mysheet2', 'Mysheet', 'Mysheet1 Copy']

# допустим, что нам не нужны первый и последний
# удаляем первый лист по его имени с проверкой
# существования такого имени в книге
>>> if 'Mysheet1' in wb.sheetnames:
# Если лист с именем `Mysheet1` присутствует
# в списке листов экземпляра книги, то удаляем
...     wb.remove(wb['Mysheet1'])
...
>>> wb.sheetnames
# ['NewPage', 'Mysheet2', 'Mysheet', 'Mysheet1 Copy']
```

```
# удаляем последний лист через оператор
# `del`, имя листа извлечем по индексу
# полученного списка `name_list`
>>> del wb[name_list[-1]]
```



S

skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустроитесь Junior-специалистом.

Налоговый вычет 13%

Спикеры-практики

Трудоустройство

Учитесь сейчас, платите потом

Узнать больше

```
et']
```

значению.

ст, можно начинать изменять содержимое ячеек. К ячейкам можно обращаться непосредственно по примеру `ws['A4']`. Это вернет ячейку на A4 или создаст ее, если она еще не существует. Рассмотрим теперь, как изменить значение ячейки напрямую:

Существует также метод `worksheet.cell()`. Он обеспечивает доступ к ячейкам с непосредственным указанием значений строк и столбцов:

```
>>> d = ws.cell(row=4, column=2, value=10)
>>> d
# <Cell 'NewPage'.B4>
>>> d.value = 3.14
>>> print(d.value)
# 3.14
```

Примечание. При создании рабочего листа в памяти, он не содержит ячеек. Ячейки создаются при первом доступе к ним.

Важно! Из-за такого поведения, простой перебор ячеек в цикле, **создаст объекты этих ячеек в памяти**, даже если не присваивать им значения.

Не запускайте этот пример, поверьте на слово:

```
# создаст в памяти 100x100=10000 пустых объектов
# ячеек, просто так израсходовав оперативную память.
>>> for x in range(1,101):
...     for y in range(1,101):
...         ws.cell(row=x, column=y)
```

Доступ к диапазону ячеек листа электронной таблицы.

Диапазон с ячейками активного листа электронной таблицы можно получить с помощью простых срезов. Эти срезы будут возвращать итераторы объектов ячеек.

```
>>> cell_range = ws['A1':'C2']
>>> cell_range
# ((<Cell 'NewPage'.A1>, <Cell 'NewPage'.B1>, <Cell 'NewPage'.C1>),
# (<Cell 'NewPage'.A2>, <Cell 'NewPage'.B2>, <Cell 'NewPage'.C2>))
```


Аналогично можно получить диапазоны имеющихся строк или столбцов на листе:

```
# Все доступные ячейки в колонке `C`
>>> colC = ws['C']
# Все доступные ячейки в диапазоне колонок `C:D`
>>> range = ws['C:D']
# Все доступные ячейки в строке 10
```

```
>>> row10 = ws[10]
# Все доступные ячейки в диапазоне строк `5:10`
>>> row_range = ws[5:10]
```

Можно также использовать метод `Worksheet.iter_rows()`:

РЕКЛАМА • 16+



S skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустроитесь Junior-специалистом.

Налоговый вычет 13% >

Спикеры-практики >

Трудоустройство >

Учитесь сейчас, платите потом >

Узнать больше

```
for row in ws.iter_rows(max_col=3, max_row=2):
```

`Worksheet.iter_cols()` будет возвращать столбцы:

```
for col in ws.iter_cols(max_col=3, max_row=2):
```

Из соображений производительности метод `Worksheet.iter_cols()` недоступен в [режиме только для чтения](#).

Если нужно перебирать строки или столбцы файла, то можно использовать свойство `Worksheet.rows`:

```
>>> ws = wb.active
>>> ws['C9'] = 'hello world'
>>> tuple(ws.rows)
# ((<Cell Sheet.A1>, <Cell Sheet.B1>, <Cell Sheet.C1>),
# (<Cell Sheet.A2>, <Cell Sheet.B2>, <Cell Sheet.C2>),
# (<Cell Sheet.A3>, <Cell Sheet.B3>, <Cell Sheet.C3>),
# ...
# (<Cell Sheet.A7>, <Cell Sheet.B7>, <Cell Sheet.C7>),
# (<Cell Sheet.A8>, <Cell Sheet.B8>, <Cell Sheet.C8>),
# (<Cell Sheet.A9>, <Cell Sheet.B9>, <Cell Sheet.C9>))
```

или свойство `Worksheet.columns`:

```
>>> tuple(ws.columns)
# ((<Cell Sheet.A1>,
# <Cell Sheet.A2>,
# ...
# <Cell Sheet.B8>,
# <Cell Sheet.B9>),
# (<Cell Sheet.C1>,
# <Cell Sheet.C2>,
# ...
# <Cell Sheet.C8>,
# <Cell Sheet.C9>))
```

Примечание. Из соображений производительности свойство `Worksheet.columns` недоступно в [режиме только для чтения](#).

Получение только значений ячеек активного листа.

Если просто нужны значения из рабочего листа, то можно использовать свойство активного листа `Worksheet.values`. Это свойство перебирает все строки на листе, но возвращает только значения ячеек:

```
for row in ws.values:
    for value in row:
        print(value)
```


Для возврата только значения ячейки, методы `Worksheet.iter_rows()` и `Worksheet.iter_cols()`, представленные выше, могут принимать дополнительный аргумент `values_only`:

Вверх

Сохранение данных книги в виде потока.

Если необходимо сохранить файл в поток, например, при использовании веб-приложения, такого как [Flask](#) или Django, то можно просто предоставить [tempfile.NamedTemporaryFile\(\)](#):

РЕКЛАМА · 16+



skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустройтесь Junior-специалистом.


Налоговый вычет 13% >

Список

Трудоустройство

Учитель


РЕКЛАМА · 16+ я



Углубленное изучение JavaScript и Node.js.

→

РЕКЛАМА



Python для Школьников Бесплатно. +10 баллов к ЕГЭ

→

Примечание. Атрибут `wb.template` по умолчанию имеет значение `False`, это означает - сохранить как документ.

Внимание. Следующее не удастся:

```
>>> from openpyxl import load_workbook
>>> wb = load_workbook('test.xlsx')
# Необходимо сохранить с расширением *.xlsx
>>> wb.save('new_test.xlsm') # MS Excel не может открыть документ

# Нужно указать атрибут `keep_vba=True`
>>> wb = load_workbook('test.xlsm')
>>> wb.save('new_test.xlsm')

>>> wb = load_workbook('test.xltn', keep_vba=True)
# Если нужен шаблон документа, то необходимо указать расширение *.xltn.
>>> wb.save('new_test.xlsm') # MS Excel не может открыть документ
```

Загрузка документа XLSX из файла.

Чтобы открыть существующую книгу Excel необходимо использовать функцию `openpyxl.load_workbook()`:

```
>>> from openpyxl import load_workbook
>>> wb2 = load_workbook('test.xlsx')
>>> print(wb2.sheetnames)
# ['Mysheet1', 'NewPage', 'Mysheet2', 'Mysheet']
```

Есть несколько флагов, которые можно использовать в функции `openpyxl.load_workbook()`.

- `data_only`: определяет, будут ли содержать ячейки с формулами - формулу (по умолчанию) или только значение, сохраненное/посчитанное при последнем чтении листа Excel.
- `keep_vba` определяет, сохраняются ли какие-либо элементы Visual Basic (по умолчанию). Если они сохранены, то они не могут изменяться/редактироваться.

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Встроенные в Excel формулы и модуль openpyxl](#)
- [Как установить формат в ячейку Excel модулем openpyxl](#)



S skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!

Получите поддержку от Центра Карьеры и трудоустроитесь Junior-специалистом.

- Налоговый вычет 13% >
- Спикеры-практики >
- Трудоустройство >
- Учитесь сейчас, платите потом >

Узнать больше

[Модуль openpyxl](#)
[Модуль openpyxl](#)
[, модулем openpyxl](#)
[да модулем openpyxl](#)
[щипку, модулем openpyxl](#)
[нта в Excel модулем openpyxl](#)
[, перемещение ячеек, модуль openpyxl](#)
[улем openpyxl](#)
[, модуль openpyxl](#)
[еек, модуль openpyxl](#)
[уле openpyxl](#)
[openpyxl](#)
[дуля openpyxl](#)
[умент, модуль openpyxl](#)
[openpyxl](#)
[ля openpyxl](#)
[l](#)
[ля openpyxl](#)

[DOCS-Python.ru™](#), 2023 г.

(Внимание! При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)