

Работа с файлами Excel в Python: полное руководство

Работа с файлами Excel в Python осуществляется с помощью специализированных библиотек, которые обеспечивают чтение, запись и обработку данных в форматах .xlsx и .xls. Рассмотрим четыре основные библиотеки и их практическое применение.

Основные библиотеки для работы с Excel

openpyxl - универсальная библиотека для современного формата Excel (.xlsx). Поддерживает чтение, запись и редактирование файлов, включая форматирование ячеек, создание диаграмм и работу с формулами.

xlrd - специализированная библиотека для чтения старых файлов Excel (.xls). Оптимальна для работы с файлами Excel 2003 и более ранних версий.

pandas - мощная библиотека для анализа данных, которая упрощает работу с Excel-файлами через структуру данных DataFrame. Поддерживает оба формата и предоставляет богатый функционал для обработки данных.

xlsxwriter - библиотека, ориентированная на создание новых файлов Excel с расширенными возможностями форматирования и визуализации данных.

Установка библиотек

```
pip install openpyxl xlrd pandas xlsxwriter
```

Практические примеры работы

Чтение данных с помощью openpyxl

```
from openpyxl import load_workbook

# Загрузка файла Excel
wb = load_workbook('example.xlsx')
sheet = wb.active

# Итерация по всем строкам
for row in sheet.iter_rows(values_only=True):
    print(row)

# Чтение конкретной ячейки
cell_value = sheet['A1'].value
print(f"Значение ячейки A1: {cell_value}")
```

Чтение данных с помощью xlrd

```
import xlrd

# Открытие файла Excel
wb = xlrd.open_workbook('example.xls')
sheet = wb.sheet_by_index(0)

# Чтение всех строк
for row in range(sheet.nrows):
```

```
print(sheet.row_values(row))

# Чтение конкретной ячейки
cell_value = sheet.cell_value(0, 0)
print(f"Значение ячейки A1: {cell_value}")
```

Работа с данными через pandas

```
import pandas as pd

# Загрузка данных из Excel в DataFrame
df = pd.read_excel('example.xlsx')
print(df.head())

# Обработка данных
df_filtered = df[df['column_name'] > 100]

# Сохранение обработанных данных
df_filtered.to_excel('filtered_output.xlsx', index=False)

# Работа с несколькими листами
df_dict = pd.read_excel('example.xlsx', sheet_name=None)
```

Создание файлов с xlswriter

```
import xlswriter

# Создание нового файла Excel
workbook = xlswriter.Workbook('formatted_output.xlsx')
worksheet = workbook.add_worksheet()

# Добавление данных с форматированием
bold = workbook.add_format({'bold': True})
worksheet.write('A1', 'Заголовок', bold)
worksheet.write('A2', 'Данные')

# Создание диаграммы
chart = workbook.add_chart({'type': 'line'})
chart.add_series({'values': '=Sheet1!$A$2:$A$10'})
worksheet.insert_chart('C1', chart)

workbook.close()
```

Расширенные возможности работы с Excel

Создание и редактирование файлов с openpyxl

```
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill

# Создание нового файла
```

```

wb = Workbook()
ws = wb.active

# Добавление данных
ws['A1'] = 'Название'
ws['B1'] = 'Значение'
ws['A2'] = 'Продукт 1'
ws['B2'] = 100

# Форматирование ячеек
ws['A1'].font = Font(bold=True)
ws['A1'].fill = PatternFill(start_color='FFFF00',
                             end_color='FFFF00', fill_type='solid')

# Сохранение файла
wb.save('formatted_example.xlsx')

```

Обработка больших файлов с pandas

```

import pandas as pd

# Чтение файла по частям для больших данных
chunk_size = 1000
chunks = []

for chunk in pd.read_excel('large_file.xlsx',
                           chunksize=chunk_size):
    processed_chunk = chunk.groupby('category').sum()
    chunks.append(processed_chunk)

# Объединение обработанных частей
result = pd.concat(chunks, ignore_index=True)
result.to_excel('processed_large_file.xlsx', index=False)

```

Выбор подходящей библиотеки

Используйте `openpyxl`, если вам нужно:

- Работать с современными файлами Excel (.xlsx)
- Редактировать существующие файлы
- Применять форматирование и стили
- Работать с формулами

Выберите `xlrd` для:

- Чтения старых файлов Excel (.xls)
- Простого извлечения данных
- Работы с унаследованными системами

Применяйте pandas при необходимости:

- Анализа и обработки данных
- Работы с большими объемами информации
- Интеграции с другими источниками данных
- Статистической обработки

Используйте xlswriter для:

- Создания новых файлов с нуля
- Генерации отчетов с диаграммами
- Создания файлов с расширенным форматированием
- Автоматизации создания документов

Обработка ошибок и оптимизация

```
import pandas as pd
from openpyxl import load_workbook

try:
    # Безопасное чтение файла
    df = pd.read_excel('data.xlsx')

    # Проверка наличия данных
    if df.empty:
        print("Файл пуст")
    else:
        print(f"Загружено {len(df)} строк")

except FileNotFoundError:
    print("Файл не найден")
except Exception as e:
    print(f"Ошибка при чтении файла: {e}")
```

Вот подробная таблица методов основных библиотек для работы с Excel в Python:

pandas (для работы с Excel)

Метод	Описание	Пример использования
pd.read_excel()	Чтение Excel файла	df = pd.read_excel('file.xlsx')
df.to_excel()	Запись в Excel файл	df.to_excel('output.xlsx')
pd.ExcelFile()	Создание объекта Excel файла	xls = pd.ExcelFile('file.xlsx')

Метод	Описание	Пример использования
<code>xls.sheet_names</code>	Получение списка листов	<code>sheets = xls.sheet_names</code>
<code>xls.parse()</code>	Чтение конкретного листа	<code>df = xls.parse('Sheet1')</code>

openpyxl (основные классы и методы)

Workbook (Рабочая книга)

Метод	Описание	Пример использования
<code>Workbook()</code>	Создание новой книги	<code>wb = Workbook()</code>
<code>load_workbook()</code>	Загрузка существующей книги	<code>wb = load_workbook('file.xlsx')</code>
<code>wb.save()</code>	Сохранение книги	<code>wb.save('file.xlsx')</code>
<code>wb.create_sheet()</code>	Создание нового листа	<code>ws = wb.create_sheet('NewSheet')</code>
<code>wb.remove()</code>	Удаление листа	<code>wb.remove(wb['Sheet1'])</code>
<code>wb.sheetnames</code>	Список имен листов	<code>names = wb.sheetnames</code>
<code>wb.active</code>	Активный лист	<code>ws = wb.active</code>
<code>wb.copy_worksheet()</code>	Копирование листа	<code>wb.copy_worksheet(ws)</code>

Worksheet (Рабочий лист)

Метод	Описание	Пример использования
<code>ws.cell()</code>	Получение/установка значения ячейки	<code>ws.cell(row=1, column=1, value='Hello')</code>
<code>ws.append()</code>	Добавление строки	<code>ws.append(['A', 'B', 'C'])</code>

Метод	Описание	Пример использования
<code>ws.insert_rows()</code>	Вставка строк	<code>ws.insert_rows(1, 3)</code>
<code>ws.insert_cols()</code>	Вставка столбцов	<code>ws.insert_cols(1, 2)</code>
<code>ws.delete_rows()</code>	Удаление строк	<code>ws.delete_rows(1, 2)</code>
<code>ws.delete_cols()</code>	Удаление столбцов	<code>ws.delete_cols(1, 2)</code>
<code>ws.max_row</code>	Максимальная строка с данными	<code>max_r = ws.max_row</code>
<code>ws.max_column</code>	Максимальный столбец с данными	<code>max_c = ws.max_column</code>
<code>ws.iter_rows()</code>	Итерация по строкам	<code>for row in ws.iter_rows():</code>
<code>ws.iter_cols()</code>	Итерация по столбцам	<code>for col in ws.iter_cols():</code>
<code>ws.merge_cells()</code>	Объединение ячеек	<code>ws.merge_cells('A1:B2')</code>
<code>ws.unmerge_cells()</code>	Разъединение ячеек	<code>ws.unmerge_cells('A1:B2')</code>

Cell (Ячейка)

Свойство/Метод	Описание	Пример использования
<code>cell.value</code>	Значение ячейки	<code>val = cell.value</code>
<code>cell.coordinate</code>	Координаты ячейки	<code>coord = cell.coordinate</code>
<code>cell.row</code>	Номер строки	<code>row = cell.row</code>
<code>cell.column</code>	Номер столбца	<code>col = cell.column</code>

Свойство/Метод	Описание	Пример использования
cell.column_letter	Буква столбца	letter = cell.column_letter
cell.font	Шрифт ячейки	cell.font = Font(bold=True)
cell.fill	Заливка ячейки	cell.fill = PatternFill(fill_type='solid')
cell.border	Границы ячейки	cell.border = Border(left=Side())
cell.alignment	Выравнивание	cell.alignment = Alignment(horizontal='center')
cell.number_format	Формат числа	cell.number_format = '0.00'

xlswriter (основные методы)

Workbook

Метод	Описание	Пример использования
xlswriter.Workbook()	Создание новой книги	wb = xlswriter.Workbook('file.xlsx')
wb.add_worksheet()	Добавление листа	ws = wb.add_worksheet()
wb.add_format()	Создание формата	fmt = wb.add_format({'bold': True})
wb.close()	Закрытие и сохранение	wb.close()

Worksheet

Метод	Описание	Пример использования
ws.write()	Запись данных	ws.write(0, 0, 'Hello')

Метод	Описание	Пример использования
<code>ws.write_row()</code>	Запись строки	<code>ws.write_row(0, 0, ['A', 'B', 'C'])</code>
<code>ws.write_column()</code>	Запись столбца	<code>ws.write_column(0, 0, [1, 2, 3])</code>
<code>ws.write_formula()</code>	Запись формулы	<code>ws.write_formula(0, 0, '=A1+B1')</code>
<code>ws.insert_image()</code>	Вставка изображения	<code>ws.insert_image('A1', 'image.png')</code>
<code>ws.add_chart()</code>	Добавление графика	<code>chart = wb.add_chart({'type': 'column'})</code>
<code>ws.set_column()</code>	Настройка столбца	<code>ws.set_column('A:A', 20)</code>
<code>ws.set_row()</code>	Настройка строки	<code>ws.set_row(0, 30)</code>
<code>ws.merge_range()</code>	Объединение ячеек	<code>ws.merge_range('A1:B2', 'Text')</code>
<code>ws.freeze_panes()</code>	Закрепление панелей	<code>ws.freeze_panes(1, 0)</code>
<code>ws.autofilter()</code>	Автофильтр	<code>ws.autofilter('A1:D10')</code>

`xlrd` (для чтения старых файлов `.xls`)

Workbook

Метод	Описание	Пример использования
<code>xlrd.open_workbook()</code>	Открытие файла	<code>wb = xlrd.open_workbook('file.xls')</code>
<code>wb.sheet_names()</code>	Имена листов	<code>names = wb.sheet_names()</code>
<code>wb.sheet_by_index()</code>	Лист по индексу	<code>ws = wb.sheet_by_index(0)</code>

Метод	Описание	Пример использования
<code>wb.sheet_by_name()</code>	Лист по имени	<code>ws = wb.sheet_by_name('Sheet1')</code>
<code>wb.nsheets</code>	Количество листов	<code>count = wb.nsheets</code>

Worksheet

Метод	Описание	Пример использования
<code>ws.cell_value()</code>	Значение ячейки	<code>val = ws.cell_value(0, 0)</code>
<code>ws.cell_type()</code>	Тип ячейки	<code>type = ws.cell_type(0, 0)</code>
<code>ws.nrows</code>	Количество строк	<code>rows = ws.nrows</code>
<code>ws.ncols</code>	Количество столбцов	<code>cols = ws.ncols</code>
<code>ws.row_values()</code>	Значения строки	<code>row = ws.row_values(0)</code>
<code>ws.col_values()</code>	Значения столбца	<code>col = ws.col_values(0)</code>

xlwt (для записи в .xls)

Workbook

Метод	Описание	Пример использования
<code>xlwt.Workbook()</code>	Создание книги	<code>wb = xlwt.Workbook()</code>
<code>wb.add_sheet()</code>	Добавление листа	<code>ws = wb.add_sheet('Sheet1')</code>
<code>wb.save()</code>	Сохранение	<code>wb.save('file.xls')</code>

Worksheet

Метод	Описание	Пример использования
<code>ws.write()</code>	Запись данных	<code>ws.write(0, 0, 'Hello')</code>
<code>ws.write_merge()</code>	Запись с объединением	<code>ws.write_merge(0, 1, 0, 1, 'Text')</code>