

Материалы занятия

Курс: Разработка Web-приложений на Python, с применением
Фреймворка Django

Дисциплина: Основы программирования на Python

Тема занятия №13: Работа с комплексными файлами - excel, json, word.
Библиотеки `openpyxl`, `json`, `docx`

1. Работа с библиотекой `openpyxl` – открытие, чтение, запись и форматирование файлов

Электронные таблицы Excel — это интуитивно понятный и удобный способ манипулирования большими наборами данных без какой-либо предварительной технической подготовки. Поэтому, это один из форматов, с которым, в какой-то момент времени, вам придется иметь дело. Часто будут стоять задачи по извлечению каких-то данных из базы данных или файла логов в электронную таблицу Excel, или наоборот, преобразовывать электронную таблицу Excel в какую-либо более удобную программную форму, примеров этому масса.

Модуль `openpyxl` — это библиотека Python для чтения/записи форматов Office Open XML (файлов Excel 2010) с расширениями `xlsx/xlsm/xltx/xltm`.

Установка модуля `openpyxl` в виртуальное окружение

Модуль `openpyxl` размещен на PyPI, поэтому установка относительно проста.

Вариант для Linux и Mac:

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

Вариант для Windows:

```
$ python3 -m venv venv
$ source venv\Scripts\activate
$ pip install openpyxl
```

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()`:

```

# вставить рабочий лист в конец (по умолчанию)
ws1 = 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)

```

Доступ к ячейке и ее значению

После того как выбран рабочий лист, можно начинать изменять содержимое ячеек. К ячейкам можно обращаться непосредственно как к ключам рабочего листа, например `ws['A4']`. Это вернет ячейку на A4 или создаст ее, если она еще не существует. Значения могут быть присвоены напрямую:

```

ws['A4'] = 5

print(ws['A4'])
# <Cell 'NewPage'.A4>
print(ws['A4'].value) # 5
print(ws['A4'].column) # 1
print(ws['A4'].row) # 4

```

Если объект ячейки присвоить переменной, то этой переменной, также можно присваивать значение:

```

c = ws['A4']
c.value = c.value * 2
print(c.value) # 10

```

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']

print(cell_range)
# ((<Cell 'NewPage'.A1>, <Cell 'NewPage'.B1>, <Cell 'NewPage'.C1>),
# (<Cell 'NewPage'.A2>, <Cell 'NewPage'.B2>, <Cell 'NewPage'.C2>))

```

Сохранение созданной книги в файл Excel

Самый простой и безопасный способ сохранить книгу, это использовать метод `Workbook.save()` объекта `Workbook`:

```
wb = Workbook()  
wb.save('test.xlsx')
```

Внимание. Эта операция перезапишет существующий файл без предупреждения!!!

После сохранения, можно открыть полученный файл в Excel и посмотреть данные, выбрав лист с именем `NewPage`.

Примечание. Расширение имени файла не обязательно должно быть `xlsx` или `xlsm`, хотя могут возникнуть проблемы с его открытием непосредственно в другом приложении. Поскольку файлы OOXML в основном представляют собой ZIP-файлы, их также можете открыть с помощью своего любимого менеджера ZIP-архивов.

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']
```

Источник: <https://docs-python.ru/packages/modul-openpyxl/>