



*School of  
Computer  
Science*

# РАБОТА С ФОРМАТОМ XLSX (EXCEL)

## ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ЧТО МОЖНО ДЕЛАТЬ ИЗ PYTHON

- Будем использовать модуль `openpyxl`, установите его командой:
  - `pip install openpyxl`
- Документация:  
<https://openpyxl.readthedocs.io/en/stable/>
- Можно делать следующее:
  - Копирование данных из одной таблицы в другую
  - Поиск среди многих тысяч строк
  - Внесение изменений по определенным критериям
  - ...



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ТЕРМИНОЛОГИЯ

- Документ электронной таблицы Excel называется *Рабочей книгой*
- Каждая книга может содержать любое количество *Листов*
- Лист состоит из столбцов, адресуемых с помощью букв (начиная с А), и строк, адресуемых с помощью чисел (начиная с единицы)



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ЧТЕНИЕ ДОКУМЕНТОВ EXCEL

– Работаем с примером `example.xlsx`:

Текущий рабочий каталог

```
>>> import openpyxl, os
>>> os.getcwd()
'C:\\Users\\slukashenko\\Office\\Лекции в IT-школе\\Lesson 21\\Scripts'
>>> os.chdir('C:/Users/slukashenko/Office/Лекции в IT-школе/
Lesson 21/Scripts')
>>>
>>> wb = openpyxl.load_workbook("example.xlsx")
>>> type(wb)
<class 'openpyxl.workbook.workbook.Workbook'>
```

Изменение рабочего каталога

Загрузка книги Excel



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ПОЛУЧЕНИЕ СПИСКА ЛИСТОВ

- Продолжаем работать с примером `example.xlsx`, обращаясь к листам:

```
>>> ...
>>> wb.sheetnames
['Sheet1', 'Sheet2', 'Sheet3']
>>> wb["Sheet1"]
<Worksheet "Sheet1">
>>> sheet3 = wb["Sheet3"]
>>> type(sheet3)
<class 'openpyxl.worksheet.worksheet.Worksheet'>
>>> sheet3.title
'Sheet3'
>>> activeSheet = wb.active
>>> activeSheet
<Worksheet "Sheet1">
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ПОЛУЧЕНИЕ ЯЧЕЕК РАБОЧИХ ЛИСТОВ

- Продолжаем работать с примером `example.xlsx`, обращаясь к ячейкам:

```
>>>
>>> ...
>>> activeSheet["A1"]
<Cell 'Sheet1'.A1>
>>> activeSheet["A1"].value
datetime.datetime(2015, 4, 5, 13, 34, 2)
>>> cell = activeSheet["B1"]
>>> cell.value
'Apples'
>>> "Строка " + str(cell.row) + ", Столбец " + cell.column + ": "
+ cell.value
'Строка 1, Столбец B: Apples'
>>> "Ячейка " + cell.coordinate + " : " + cell.value
'Ячейка B1 : Apples'
>>> activeSheet["C1"].value
73
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ПОЛУЧЕНИЕ ЯЧЕЕК ПО НОМЕРАМ

- Продолжаем работать с примером `example.xlsx`, обращаясь к ячейкам по ЧИСЛОВЫМ ССЫЛКАМ:

```
>>> ...  
>>> activeSheet.cell(row=1, column=2)  
<Cell 'Sheet1'.B1>  
>>> activeSheet.cell(row=1, column=2).value  
'Apples'  
>>> for i in range(1, 8, 2):  
        print(i, activeSheet.cell(row=i, column=2).value)
```

```
1 Apples  
3 Pears  
5 Apples  
7 Strawberries
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ПЕРЕБОР СТРОК

```
>>> tuple(activeSheet['A1':'C3'])
((<Cell 'Sheet1'.A1>, <Cell 'Sheet1'.B1>, <Cell 'Sheet1'.C1>),
(<Cell 'Sheet1'.A2>, <Cell 'Sheet1'.B2>, <Cell 'Sheet1'.C2>),
(<Cell 'Sheet1'.A3>, <Cell 'Sheet1'.B3>, <Cell 'Sheet1'.C3>))
>>>
>>> for rowOfCellObjects in activeSheet['A1':'C3']:
    for cellObj in rowOfCellObjects:
        print(cellObj.coordinate, cellObj.value)
    print('--- END OF THE ROW ---')
```

```
A1 2015-04-05 13:34:02
B1 Apples
C1 73
--- END OF THE ROW ---
A2 2015-04-05 03:41:23
B2 Cherries
C2 85
--- END OF THE ROW ---
A3 2015-04-06 12:46:51
B3 Pears
C3 14
--- END OF THE ROW ---
```





# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## РАБОЧИЕ КНИГИ, ЛИСТЫ И ЯЧЕЙКИ

- Алгоритм чтения ячеек:
  - Импортируем модуль `openpyxl`
  - Вызываем метод `openpyxl.load_workbook()`
  - Получаем объект класса `Workbook`
  - Обращаемся к свойству `active` этого объекта или находим рабочий лист, указывая его в качестве ключа
  - Получаем объект класса `Worksheet`
  - Используем индексирование для этого объекта или метод `cell()`
  - Получаем объект класса `Cell`
  - Читаем свойство `value` этого объекта



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ИДЕИ ПРОГРАММ ПО ЧТЕНИЮ ДАННЫХ

- Можно решать следующие задачи:
  - Сравнение данных, хранящихся в нескольких строках таблицы
  - Открытие нескольких файлов Excel и сравнение хранящихся в них данных
  - Поиск пустых строк или недопустимых данных в ячейках электронной таблицы
  - Чтение данных из электронной таблицы и их использование в качестве входных данных для программы на языке Python



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## СОЗДАНИЕ И СОХРАНЕНИЕ ДОКУМЕНТОВ

```
>>> wb_new = openpyxl.Workbook()
>>> wb_new.sheetnames
['Sheet']
>>> sheet = wb_new.active
>>> sheet.title
'Sheet'
>>> sheet.title = "Spam Spam Spam"
>>> wb_new.sheetnames
['Spam Spam Spam']
```

```
>>> wb = openpyxl.load_workbook("example.xlsx")
>>> sheet = wb.active
>>> sheet.title = "Spam Spam Spam"
>>> wb.save("example_copy.xlsx")
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## СОЗДАНИЕ РАБОЧИХ ЛИСТОВ

- Продолжаем предыдущий скрипт:

```
>>> import openpyxl
>>> wb = openpyxl.Workbook()
>>> wb.sheetnames
['Sheet']
>>> wb.create_sheet()
<Worksheet "Sheet1">
>>> wb.sheetnames
['Sheet', 'Sheet1']
>>> wb.create_sheet(index=0, title="Первый лист")
<Worksheet "Первый лист">
>>> wb.sheetnames
['Первый лист', 'Sheet', 'Sheet1']
>>> wb.create_sheet(index=2, title="Средний лист")
<Worksheet "Средний лист">
>>> wb.sheetnames
['Первый лист', 'Sheet', 'Средний лист', 'Sheet1']
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## УДАЛЕНИЕ РАБОЧИХ ЛИСТОВ

– Продолжаем предыдущий скрипт:

```
>>>
>>> wb.sheetnames
['Первый лист', 'Sheet', 'Средний лист', 'Sheet1']
>>> wb.remove(wb["Средний лист"])
>>> wb.remove(wb["Sheet1"])
>>> wb.sheetnames
['Первый лист', 'Sheet']
```



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## ЗАПИСЬ ЗНАЧЕНИЙ В ЯЧЕЙКИ

- Сохранение данных в ячейке напоминает запись значений в ключи словаря:

```
>>> import openpyxl
>>> wb = openpyxl.Workbook()
>>> sheet = wb.active
>>> sheet["A1"] = "Здравствуй, мир!"
>>> sheet["A1"].value
'Здравствуй, мир!'
```



# ПРАКТИЧЕСКИЙ ПРИМЕР С EXCEL

## ОБНОВЛЕНИЕ ДАННЫХ В ТАБЛИЦЕ

- В файле `produceSales.xlsx` реестр проданных товаров с ценами
- Нужно исправить цены у нескольких товаров
- Скрипт `update_produce.py` делает следующее:
  - Пробегает по всем строкам данных
  - Если товар в строке находится в списке для изменения – меняет цену



# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## РАБОТА С ФОРМУЛАМИ

- Ячейка, которая начинается со знака “=” задает формулу, которая рассчитывает значение на основании других ячеек – см. [formula.py](#)
- Листы с формулами могут считываться в 2-ух режимах:
  1. чтение формул
  2. чтение данных
- Примеры в [load\\_formula.py](#)





# ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

## РЕГУЛИРОВАНИЕ ЯЧЕЕК

- Установка высоты строки и ширины колонки: [tall\\_wide.py](#)
- Слияние ячеек – [merge.py](#):
- Разъединение ячеек – [unmerge.py](#)
- «Замораживание» строк – [freeze.py](#)



# ПРАКТИКА С ТАБЛИЦЕЙ EXCEL

## ТАБЛИЦА УМНОЖЕНИЯ

- Создайте программу, которая принимает число  $N$  из командной строки и создает таблицу умножения  $N \times N$  в таблице Excel
- Пример результата для  $N=6$ :

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	8	10	12
3	3	6	9	12	15	18
4	4	8	12	16	20	24
5	5	10	15	20	25	30
6	6	12	18	24	30	36

**СПАСИБО ЗА ВНИМАНИЕ !**  
**ВОПРОСЫ ?**



*School of  
Computer  
Science*