

# Работа с документами MS Excel.

## Модуль openpyxl

---

### ✓ Задания 1 на openpyxl (для работы с Excel)

Задание 1.1: Создай новую книгу Excel Условие: Создай новый файл Excel с именем my\_first\_table.xlsx. Добавь лист с названием "Продажи". В ячейку A1 запиши "Товар", в B1 - "Количество", в C1 - "Цена". Сохрани файл.

Задание 1.2: Заполни таблицу данными Условие: Возьми файл из предыдущего задания. Добавь три строки с данными о продажах (например, Яблоки, 100, 50; Груши, 75, 70; Сливы, 50, 90). Сохрани изменения.

Задание 1.3: Прочитай данные из файла Условие: Напиши программу, которая откроет созданный файл my\_first\_table.xlsx, прочитает и выведет на экран все данные из колонки "Товар"

### Решения на занятии

```
import openpyxl
wb = openpyxl.Workbook()
wb.create_sheet(title = 'Продажи', index = 0)
sheet = wb.active
sheet.append(['Товар', 'Количество', 'Цена'])
wb.save('my_first_table_ksp.xlsx')
```

```
wb = openpyxl.Workbook()
wb.create_sheet(title = 'Продажи', index = 0)
sheet = wb.active
sheet['A1'] = 'Товар'
sheet['B1'] = 'Количество'
sheet['C1'] = 'Цена'
wb.save('my_first_table_ksp.xlsx')
```

```
wb = openpyxl.Workbook()
sheet = wb.active
```

```

sheet_title = sheet.title
sheet_title = "Продажи"
a1 = sheet.cell(row = 1, column = 1)
a1.value = "Товар"
b1 = sheet.cell(row= 1 , column = 2)
b1.value = "Количество"
c1 = sheet.cell(row = 1, column = 3)
c1.value = "Цена"
wb.save("Analytics_2.xlsx")

```

```

from openpyxl import load_workbook

```

```

file_path = "Analytics_2.xlsx"
wb = load_workbook(file_path)

```

```

# не рабочий вариант: ❌ ws = wb["Продажи"]

```

```

# Замени на: ✅

```

```

ws = wb.create_sheet("Продажи")

```

```

ws.append(["Яблоки", 100, 50])
ws.append(["Груши", 75, 70])
ws.append(["Сливы", 50, 90])

```

```

wb.save(file_path)

```

```

print("Данные успешно добавлены в файл:", file_path)

```

Данные успешно добавлены в файл: Analytics\_2.xlsx

```

from openpyxl import load_workbook

```

```

file_path = "my_first_table_ksp.xlsx"
wb = load_workbook(file_path)

```

```

ws = wb["Продажи"]

```

```

ws.append(["Яблоки", 100, 50])
ws.append(["Груши", 75, 70])
ws.append(["Сливы", 50, 90])

```

```

wb.save(file_path)

```

```

print("Данные успешно добавлены в файл:", file_path)

```

Данные успешно добавлены в файл: my\_first\_table\_ksp.xlsx

```
wb = openpyxl.load_workbook('my_first_table_ksp.xlsx')
sheet = wb.active
rows = sheet.max_row
cols = sheet.max_column
for i in range(1, rows + 1):
    for j in range(1, cols + 1):
        cell = sheet.cell(row = i, column = j)
        if cell.value == "Товар":
            print(*[p.value for p in sheet[openpyxl.utils.get_column_id(j)]])
            break
```



Товар Яблоки Груши Сливы

```
wb = openpyxl.load_workbook('my_first_table_ksp.xlsx')
ws = wb['Продажи']
print('Список товаров')
for row in range(2, ws.max_row+1):
    product_name = ws[f'A{row}'].value
    print(f' - {product_name}')
```

Список товаров

- Яблоки
- Груши
- Сливы

```
wb = openpyxl.load_workbook('my_first_table_ksp.xlsx')
sheet = wb.active
for row in sheet['A2':'A4']:
    string = ''
    for cell in row:
        string = string + str(cell.value) + ' '
    print(string)
```

Яблоки  
Груши

Сливы

## Вариант решения от преподавателя с комментариями для разбора

```
from openpyxl import Workbook

# Создаем новую книгу
wb = Workbook()

# Активируем лист (по умолчанию он уже создан)
ws = wb.active
ws.title = "Продажи" # Меняем название листа

# Заполняем шапку таблицы
ws['A1'] = 'Товар'
ws['B1'] = 'Количество'
ws['C1'] = 'Цена'

# Сохраняем файл
wb.save('my_first_table.xlsx')

from openpyxl import load_workbook

# Открываем существующий файл
wb = load_workbook('my_first_table.xlsx')
ws = wb['Продажи'] # Выбираем лист по имени

# Данные для добавления (список списков)
data = [
    ['Яблоки', 100, 50],
    ['Груши', 75, 70],
    ['Сливы', 50, 90]
]

# Записываем данные, начиная со второй строки (после шапки)
for i, row_data in enumerate(data, start=2):
    ws[f'A{i}'] = row_data[0]
    ws[f'B{i}'] = row_data[1]
    ws[f'C{i}'] = row_data[2]

# Сохраняем изменения
wb.save('my_first_table.xlsx')
```

```

from openpyxl import load_workbook

# Открываем файл
wb = load_workbook('my_first_table.xlsx')
ws = wb['Продажи']

# Читаем все значения из колонки А, начиная со второй строки (чтобы пр
print("Список товаров:")
for row in range(2, ws.max_row + 1):
    product_name = ws[f'A{row}'].value
    print(f"- {product_name}")

```

Список товаров:

- Яблоки
- Груши
- Сливы

## Задание 2 на docx (для работы с Word)

Задание 2.1: Создай новый документ Условие: Создай новый документ Word с именем my\_document.docx. Добавь в него заголовок "Мой первый документ" и параграф с текстом "Привет, мир! Это мой первый документ, созданный с помощью Python." Сохрани файл.

Задание 2.2: Измени форматирование Условие: Открой документ из предыдущего задания. Сделай заголовок жирным и выровняй его по центру. Текст в параграфе сделай курсивным. Сохрани изменения.

Задание 2.3: Прочитай документ Условие: Напиши программу, которая откроет документ my\_document.docx и выведет весь его текст в консоль.

### Решения на занятии

```

import docx
new_doc = docx.Document()
new_doc.add_heading('Мой первый документ', 0)
new_doc.add_paragraph(text = 'Привет, мир! Это мой первый документ, со
new_doc.save('my_first_doc_ksp.docx')

```

```
from docx import Document
from docx.shared import Pt
from docx.enum.text import WD_ALIGN_PARAGRAPH
from docx.xml.ns import qn
```

```
doc = Document('my_first_doc_ksp.docx')
if len(doc.paragraphs) > 0:
    heading = doc.paragraphs[0]
    for run in heading.runs:
        run.bold = True
    heading.alignment = WD_ALIGN_PARAGRAPH.CENTER
if len(doc.paragraphs) > 1:
    paragraph = doc.paragraphs[1]
    for run in paragraph.runs:
        run.italic = True
doc.save('my_first_doc_ksp.docx')
```

```
from docx import Document
```

```
doc = Document("my_first_doc_ksp.docx")
```

```
for para in doc.paragraphs:
    print(para.text)
```

Мой первый документ

Привет, мир! Это мой первый документ, созданный с помощью Python

## Вариант решения от преподавателя с комментариями для разбора

```
from docx import Document
```

```
# Создаем новый документ
doc = Document()
```

```
# Добавляем заголовок (уровень 0 - самый крупный)
doc.add_heading('Мой первый документ', 0)
```

```
# Добавляем параграф с текстом
doc.add_paragraph('Привет, мир! Это мой первый документ, созданный с п
```

```

# Сохраняем документ
doc.save('my_document.docx')

from docx import Document
from docx.enum.text import WD_ALIGN_PARAGRAPH
from docx.shared import Pt

# Открываем существующий документ
doc = Document('my_document.docx')

# Первый параграф - это заголовок
heading = doc.paragraphs[0]
heading.alignment = WD_ALIGN_PARAGRAPH.CENTER # Выравнивание по центр

# Второй параграф - это основной текст
main_text = doc.paragraphs[1]

# Меняем форматирование для всего параграфа
for run in main_text.runs:
    run.italic = True # Делаем курсивом

# Сохраняем изменения
doc.save('my_document.docx')

```



```

from docx import Document

# Открываем документ
doc = Document('my_document.docx')

# Читаем и выводим весь текст
full_text = []
for paragraph in doc.paragraphs:
    full_text.append(paragraph.text)

# Объединяем все параграфы в одну строку с переносами
print('\n'.join(full_text))

```

Мой первый документ  
Привет, мир! Это мой первый документ, созданный с помощью Python.

### Задание 3. Анализ и модификация таблицы продаж

Условие: На основе файла my\_first\_table.xlsx выполни следующие действия:

Добавь новую колонку "Сумма" (D1), которая будет рассчитываться как Количество \* Цена

Заполни эту колонку для всех товаров

Добавь итоговую строку "Итого" в конце таблицы, которая посчитает общую сумму продаж

Подсчитай и выведи в консоль среднюю цену товаров

Сохрани изменения в новый файл sales\_report.xlsx

## В НАЧАЛЕ ПОПРОБОВАТЬ РЕШИТЬ САМОСТОЯТЕЛЬНО

```
from openpyxl import load_workbook
from openpyxl.styles import Font

# Открываем существующий файл
wb = load_workbook('my_first_table_ksp.xlsx')
ws = wb['Продажи']

# 1. Добавляем заголовок для новой колонки
ws['D1'] = 'Сумма'

# 2. Заполняем колонку "Сумма" для всех товаров
total_sum = 0
total_price = 0
item_count = 0

for row in range(2, ws.max_row + 1):
    quantity = ws[f'B{row}'].value
    price = ws[f'C{row}'].value

    # Проверяем, что значения не пустые
    if quantity is not None and price is not None:
        amount = quantity * price
        ws[f'D{row}'] = amount
        total_sum += amount
        total_price += price
        item_count += 1

# 3. Добавляем итоговую строку
```



```

next_row = ws.max_row + 1
ws[f'A{next_row}'] = 'Итого'
ws[f'D{next_row}'] = total_sum

# 4. Подсчитываем и выводим среднюю цену
if item_count > 0:
    average_price = total_price / item_count
    print(f"Средняя цена товара: {average_price:.2f} руб.")
else:
    print("В таблице нет данных для анализа")

# 5. Сохраняем в новый файл
wb.save('sales_report_ksp.xlsx')

```

Средняя цена товара: 70.00 руб.

## Задание 4. Дополнительное (поиск и фильтрация)

Условие: Напиши программу, которая:

Откроет файл sales\_report.xlsx

Найдет товар с максимальной суммой продаж

Найдет товар с минимальной ценой

Найдет сколько всего товаров в отчете

Выведет эту информацию в консоль

**В НАЧАЛЕ ПОПРОБОВАТЬ РЕШИТЬ САМОСТОЯТЕЛЬНО**

```

from openpyxl import load_workbook

def analyze_sales():
    wb = load_workbook('sales_report_ksp.xlsx')
    ws = wb['Продажи']

    max_amount = 0
    max_amount_product = ""
    min_price_product = ""
    min_price = float('inf')

```

```

# Пропускаем заголовок и итоговую строку
for row in range(2, ws.max_row):
    product = ws[f'A{row}'].value
    price = ws[f'C{row}'].value
    amount = ws[f'D{row}'].value

    if product and price is not None and amount is not None:
        # Поиск товара с максимальной суммой
        if amount > max_amount:
            max_amount = amount
            max_amount_product = product

        # Поиск товара с минимальной ценой
        if price < min_price:
            min_price = price
            min_price_product = product

print("Анализ продаж:")
print(f"Товар с максимальной суммой продаж: {max_amount_product} (")
print(f"Товар с минимальной ценой: {min_price_product} ({min_price

# Дополнительная статистика
total_items = ws.max_row - 2 # без заголовка и итогов
print(f"Всего товаров в отчете: {total_items}")

# Запускаем анализ
analyze_sales()

```

Анализ продаж:

Товар с максимальной суммой продаж: Груши (5250 руб.)

Товар с минимальной ценой: Яблоки (50 руб. за шт.)

Всего товаров в отчете: 3

