

## Материалы занятия

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

Дисциплина: Основы программирования на Python

Тема занятия №15: Работа с комплексными файлами – excel, json, word.  
Библиотеки openpyxl, json, docx

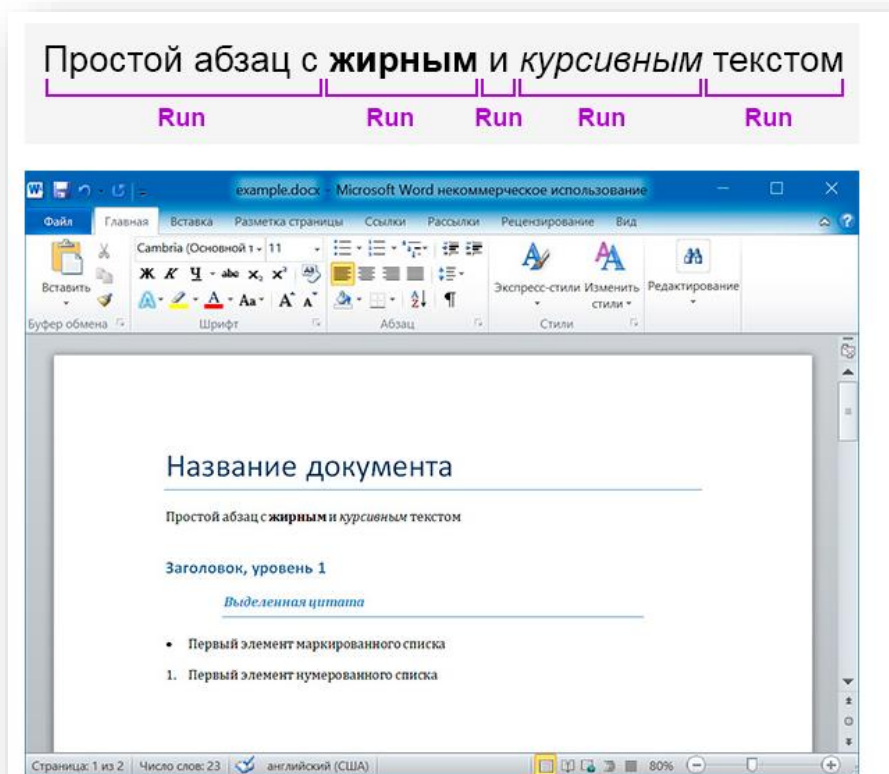
### 1. Работа с библиотекой word – открытие, чтение и запись

С помощью модуля python-docx можно создавать и изменять документы MS Word с расширением .docx. Чтобы установить этот модуль, выполняем команду:

```
$ pip install python-docx
```

#### Чтение документов MS Word

Файлы с расширением .docx обладают развитой внутренней структурой. В модуле python-docx эта структура представлена тремя различными типами данных. На самом верхнем уровне объект Document представляет собой весь документ. Объект Document содержит список объектов Paragraph, которые представляют собой абзацы документа. Каждый из абзацев содержит список, состоящий из одного или нескольких объектов Run, представляющих собой фрагменты текста с различными стилями форматирования.



```
import docx

doc = docx.Document('example.docx')

# количество абзацев в документе
print(len(doc.paragraphs))

# текст первого абзаца в документе
print(doc.paragraphs[0].text)

# текст второго абзаца в документе
print(doc.paragraphs[1].text)

# текст первого Run второго абзаца
print(doc.paragraphs[1].runs[0].text)
```

6

Название документа  
Простой абзац с жирным и курсивным текстом  
Простой абзац с

Получаем весь текст из документа:

```
text = []
for paragraph in doc.paragraphs:
    text.append(paragraph.text)
print('\n'.join(text))
```

Название документа  
Простой абзац с жирным и курсивным текстом  
Заголовок, уровень 1  
Выделенная цитата  
Первый элемент маркированного списка  
Первый элемент нумерованного списка

## Атрибуты объекта Run

Отдельные фрагменты текста, представленные объектами Run, могут подвергаться дополнительному форматированию с помощью атрибутов. Для каждого из этих атрибутов может быть задано одно из трех значений: True (атрибут активизирован), False (атрибут отключен) и None (применяется стиль, установленный для данного объекта Run).

- `bold` — полужирное начертание
- `underline` — подчеркнутый текст
- `italic` — курсивное начертание
- `strike` — зачеркнутый текст

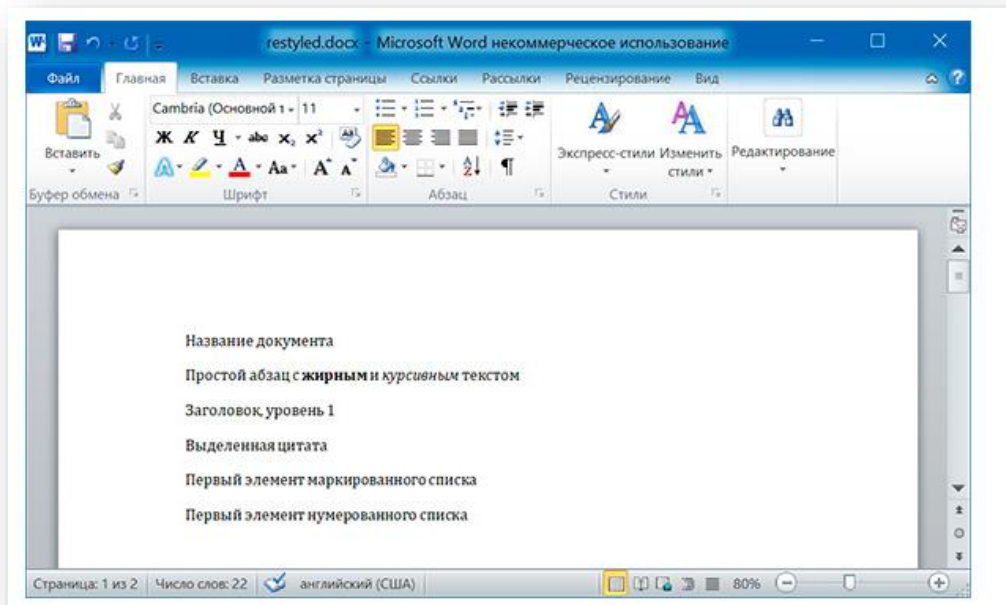
Изменим стили для всех параграфов нашего документа:

```
import docx
```

```
doc = docx.Document('example.docx')

# изменяем стили для всех параграфов
for paragraph in doc.paragraphs:
    paragraph.style = 'Normal'

doc.save('restyled.docx')
```



## Запись документов MS Word

Добавление абзацев осуществляется вызовом метода `add_paragraph()` объекта `Document`. Для добавления текста в конец существующего абзаца, надо вызвать метод `add_run()` объекта `Paragraph`:

```
import docx

doc = docx.Document()

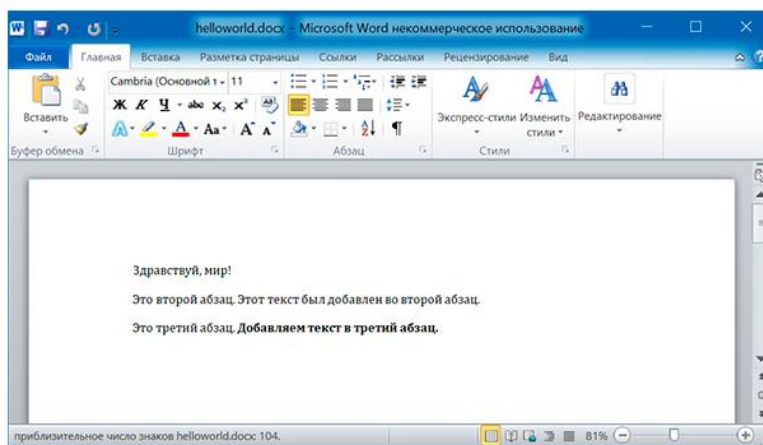
# добавляем первый параграф
doc.add_paragraph('Здравствуй, мир!')

# добавляем еще два параграфа
par1 = doc.add_paragraph('Это второй абзац.')
par2 = doc.add_paragraph('Это третий абзац.')

# добавляем текст во второй параграф
par1.add_run(' Этот текст был добавлен во второй абзац.')
```

```
# добавляем текст в третий параграф
par2.add_run(' Добавляем текст в третий абзац.').bold = True

doc.save('helloworld.docx')
```



Источник: <https://tokmakov.msk.ru/blog/item/78>