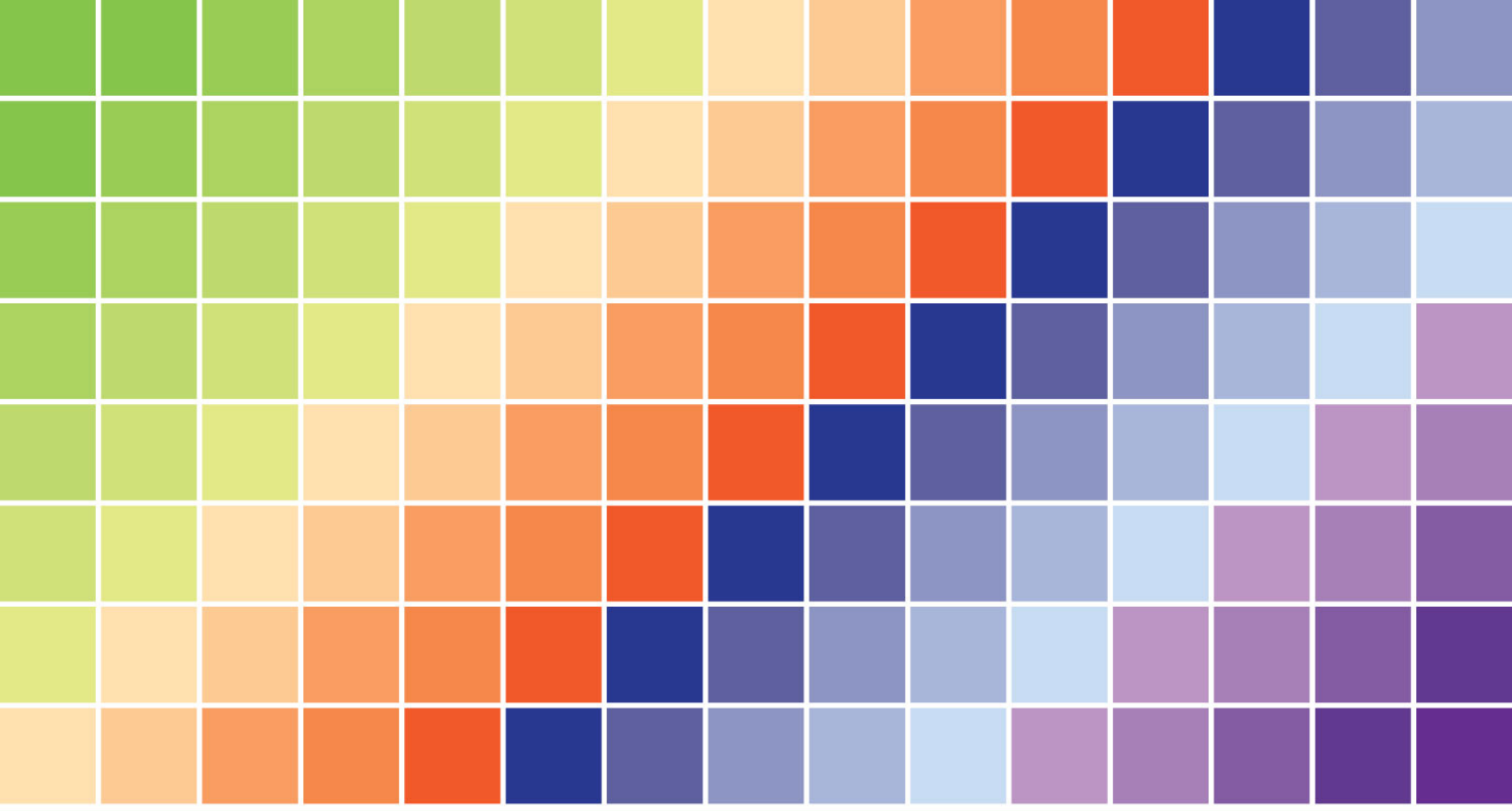


School of Computer Science

РАБОТА С ФОРМАТАМИ PDF И DOCX

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы

PDF-ДОКУМЕНТЫ
БИБЛИОТЕКА PYPDF2

- Устанавливается командой
  - pip install PyPDF2
- Импортируется так:
  - import PyPDF2
- Работает с документами PDF – Portable Document Format
- PDF-файлы поддерживаются на разных устройствах и операционных системах
- Документация: https://pythonhosted.org/PyPDF2

PDF-ДОКУМЕНТЫ
ИЗВЛЕЧЕНИЕ ТЕКСТА ИЗ PDF-ДОКУМЕНТА

- Импортируем модуль PyPDF2
- Открываем PDF в режиме чтения двоичных файлов
- Создаем объект PdfFileReader
- Создаем объект PageObject для заданного номера страницы
- Читаем текст этой страницы
- См. пример Shell-скрипта в pdf_fetch_text.txt

PDF-ДОКУМЕНТЫ
СОЗДАНИЕ PDF-ДОКУМЕНТА СЛИЯНИЕМ

- Объект PdfFileWriter можно использовать для создания PDF
- Можно соединять несколько PDF-документов в один
- Метод write() объекта PdfFileWriter принимает файл, открытый для записи в бинарном режиме
- См. пример Shell-скрипта в pdf_combine.txt

PDF-ДОКУМЕНТЫ
УПРАЖНЕНИЕ «КОНСТИТУЦИЯ»

- Вы работаете программистом в Гос. думе РФ
- Председатель комитета по законодательству Павел Крашенинников попросил вас соединить два PDF-файла:
  - Конституция РФ – constitution.pdf
  - Дополнения к Конституции – constitution_adjustment.pdf
- Руководствуясь правилом «*Я ж программист*» вы решаете написать универсальную функцию
- Напишите такую функцию, которая:
  - Принимает список имен PDF-файлов любой длины
  - Сохраняет результат слияния файлов из списка в новый PDF-файл с заданным именем
- Проверьте эту функцию с текстом основного закона РФ и дополнений, которые к нему предлагаются

ДОКУМЕНТЫ WORD
БИБЛИОТЕКА DOCX

- Устанавливается командой
  - pip install python-docx
- Импортируется так:
  - import docx
- Помогает создавать и читать документы MS Word в формате docx
- Документация: https://pypi.python.org/pypi/python-docx

ДОКУМЕНТЫ WORD
СТРУКТУРА

- Текст состоит из параграфов
- Параграф (абзац) – состоит из текста с разным оформлением:

- Группированные элементы оформления параграфа образуют стиль документа
- Каждой смене стиля в строке соответствует свой объект Run

ДОКУМЕНТЫ WORD
ЧТЕНИЕ ДОКУМЕНТА

```
>>> import docx, os
>>> os.chdir("C:\\Users\\slukashenko\\Office\\Лекции в IT-школе\\Lesson 21\\
Scripts")
>>>
>>> doc = docx.Document('demo.docx')
>>> len(doc.paragraphs)
7
>>> doc.paragraphs[0].text
'Document Title'
>>> doc.paragraphs[1].text
'A plain paragraph with some bold and some italic'
>>> len(doc.paragraphs[1].runs)
5
>>> doc.paragraphs[1].runs[0].text
'A plain paragraph with'
>>> doc.paragraphs[1].runs[1].text
' some '
>>> doc.paragraphs[1].runs[2].text
'bold'
>>> doc.paragraphs[1].runs[3].text
' and some '
>>> doc.paragraphs[1].runs[4].text
'italic'
```

ПОЛУЧЕНИЕ ВСЕГО ТЕКСТА ИЗ ДОКУМЕНТА

- Смотрите функцию get_text() в модуле read_docx.py:
  - Открываем документ Word
  - В цикле просматриваем все абзацы и добавляем их текст в список
  - Объединяем текст всех абзацев из списка, разделяя их переводом строки
- Пример использования:

```
>>> import read_docx
>>> read_docx.get_text('demo.docx')
'Document Title\nA plain paragraph with some bold and some italic
\nHeading, level 1\nIntense quote\nfirst item in unordered list\n
first item in ordered list\n\n'
```

ДОКУМЕНТЫ WORD
УПРАВЛЕНИЕ СТИЛЯМИ

- Стили могут применяться к абзацам, символам и фрагментам текста (Runs)
- В Python с помощью библиотеки docx можно управлять стилями, которые уже существуют в документе
- У каждого из объектов для этого определен набор логических свойств
- Пример использования:

```
>>> doc = docx.Document("demo.docx")
>>> doc.paragraphs[0].style = 'Normal'
>>> doc.paragraphs[1].runs[1].underline = True
>>> doc.paragraphs[1].runs[3].underline = True
>>> doc.save("restyled.docx")
```

ДОКУМЕНТЫ WORD
ЗАПИСЬ ДОКУМЕНТОВ

```
>>> doc = docx.Document()
>>> doc.add_paragraph('Здравствуй, Word!')
<docx.text.paragraph.Paragraph object at 0x030516F0>
>>> doc.save('HelloWord.docx')
```

```
>>> doc = docx.Document()
>>> doc.add_paragraph('Здравствуй, Word!')
<docx.text.paragraph.Paragraph object at 0x03074650>
>>> paraObj1 = doc.add_paragraph('Это второй абзац')
>>> paraObj2 = doc.add_paragraph('Это еще один абзац')
>>> paraObj1.add_run('Этот текст добавлен во 2-ой абзац')
<docx.text.run.Run object at 0x030746D0>
>>> doc.save('MultipleParagraphs.docx')
```

ДОКУМЕНТЫ WORD
ДОБАВЛЕНИЕ ЗАГОЛОВКОВ

```
>>> doc = docx.Document()
>>> doc.add_heading('Header 0', 0)
<docx.text.paragraph.Paragraph object at 0x031CDD30>
>>> doc.add_heading('Header 1', 1)
<docx.text.paragraph.Paragraph object at 0x031CD0F0>
>>> doc.add_heading('Header 2', 2)
<docx.text.paragraph.Paragraph object at 0x031CDDB0>
>>> doc.add_heading('Header 3', 3)
<docx.text.paragraph.Paragraph object at 0x031CDD30>
>>> doc.add_heading('Header 4', 4)
<docx.text.paragraph.Paragraph object at 0x031CD0F0>
>>> doc.save('Headings.docx')
```

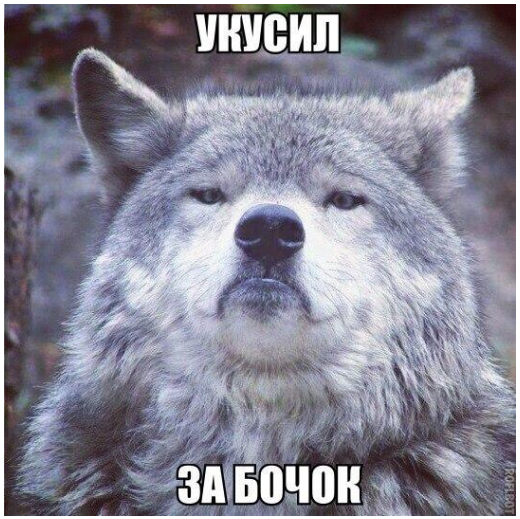
ДОКУМЕНТЫ WORD
РАЗРЫВЫ СТРОК И СТРАНИЦ

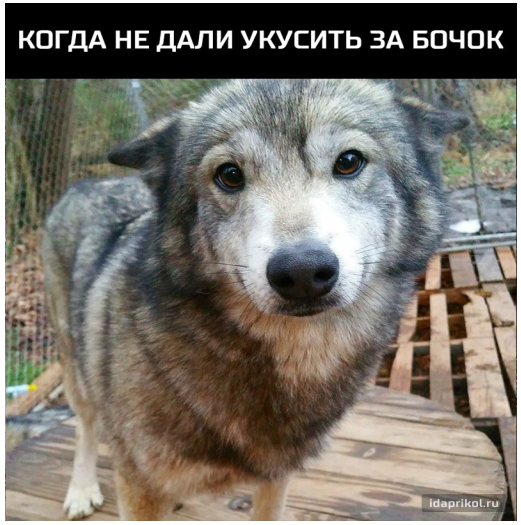
– Чтобы добавить разрыв строки, не начиная новый абзац, или вставить разрыв страницы, используется метод add_break() объекта Run:

```
>>> from docx.enum.text import WD_BREAK
>>> doc = docx.Document()
>>> doc.add_paragraph('Этот абзац на первой странице.')
<docx.text.paragraph.Paragraph object at 0x031D2FB0>
>>> doc.paragraphs[0].runs[0].add_break(WD_BREAK.PAGE)
>>> doc.add_paragraph('Этот абзац на второй странице.')
<docx.text.paragraph.Paragraph object at 0x030519D0>
>>> doc.save("TwoPages.docx")
```

ДОКУМЕНТЫ WORD
ВСТАВКА КАРТИНОК

```
>>> from docx.shared import Inches
>>> doc = docx.Document()
>>> doc.add_picture('dog_1.jpg', width=Inches(6))
<docx.shape.InlineShape object at 0x031D25D0>
>>> doc.paragraphs[0].runs[0].add_break(WD_BREAK.PAGE)
>>> doc.add_picture('dog_2.jpg', width=Inches(6))
<docx.shape.InlineShape object at 0x031D2E50>
>>> doc.save("Dogs.docx")
```





ДОКУМЕНТЫ WORD
УПРАЖНЕНИЕ «ПРОГУЛОЧНЫЕ СОБАКИ»

- Вы работаете программистом в кинологическом клубе города Москва
- В связи с самоизоляцией, в Апреле-Мае 2020, люди могут гулять около дома только с собаками
- Ваше руководство решило отдавать своих собак в аренду для прогулок на час
- Изображения собачек собраны в каталоге Dogs
- Необходимо подготовить файл в формате Word для публикации на сайте объявлений
- Напишите функцию, которая принимает список с именами картинок и сохраняет их в docx-файл
- В полученном файле MS Word каждая собачка должна находиться на отдельной странице

СПАСИБО ЗА ВНИМАНИЕ !

ВОПРОСЫ ?

School of Computer Science