



# Информационные ресурсы в финансовом мониторинге

---

НИЯУ МИФИ, КАФЕДРА ФИНАНСОВОГО МОНИТОРИНГА

КУРС ЛЕКЦИЙ

В.Ю. РАДЫГИН. ЛЕКЦИЯ 4

# Часть 1

---

АВТОМАТИЧЕСКАЯ ОБРАБОТКА ТЕКСТОВЫХ ДОКУМЕНТОВ

# Основные форматы

---

Наиболее распространены в интернете следующие форматы представления текста:

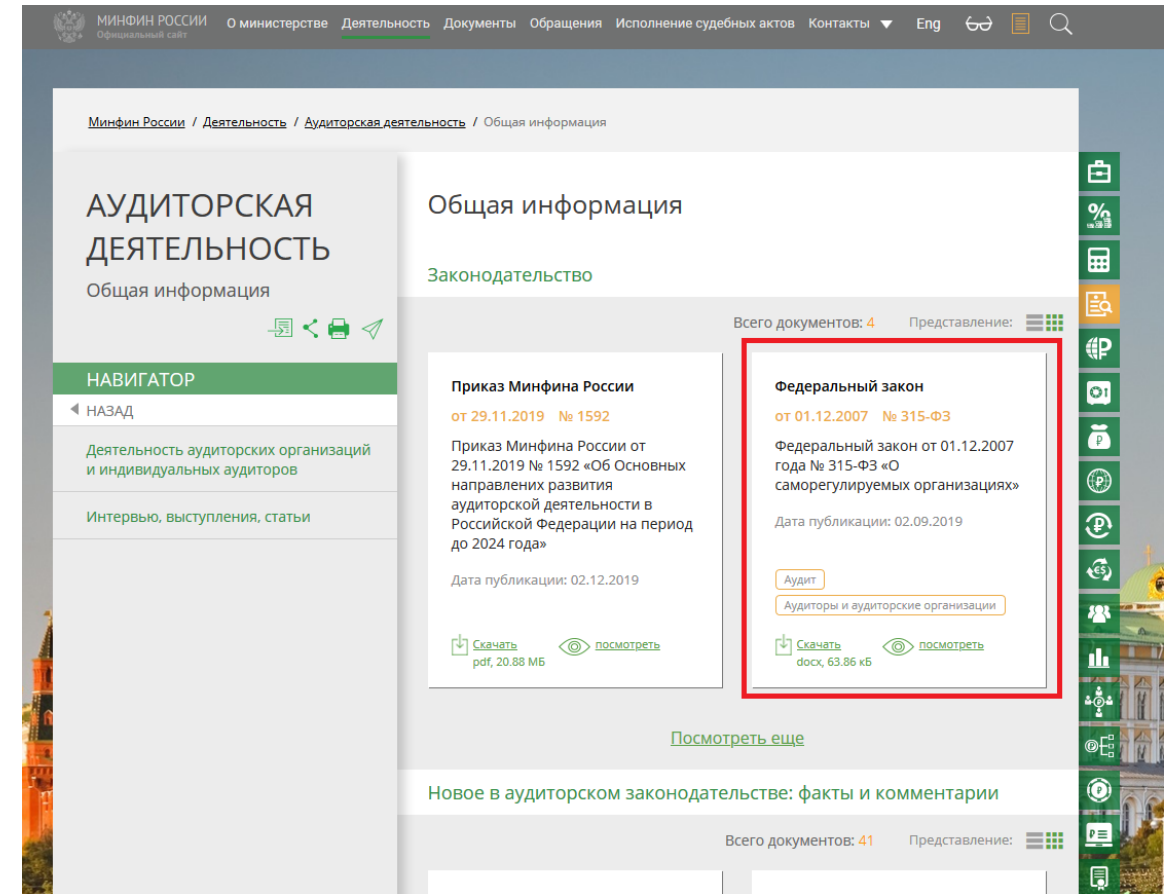
- rtf;
- doc;
- docx;
- pdf, в том числе:
  - с текстовым слоем;
  - без текстового слоя.

Для корректной работы с двумя первыми форматами нормальных средств нет. Можно либо использовать библиотеку `comtypes` и запускать для конвертации непосредственно Word, либо использовать `OpenOfficeAPI`, что тоже не очень приятно! К счастью, сегодня данные форматы уже не популярны, как раньше. Большинство документов доступны в DOCX и PDF. Разберём, как работать с ними.

# Задача 1

На сайте Министерства финансов РФ (<https://www.minfin.ru>) [1] выложен Федеральный закон от 01.12.2007 года № 315-ФЗ «О саморегулируемых организациях».

Задача: напишите программу, скачивающую данный закон, находящую и распечатывающую его 7 статью со всеми подстатьями (7.1 и т.д.) в виде обычного текста.



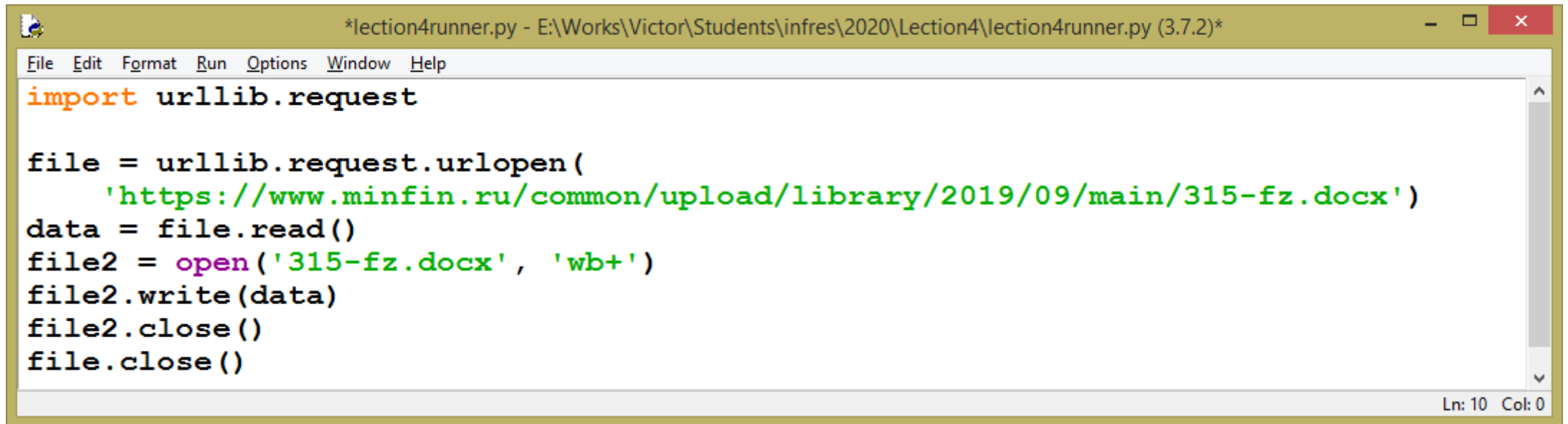
# Шаг 1: скачивание

---

Скачивать документы с сайтов мы уже учились в предыдущих лекциях. Для этого можно использовать библиотеку Urllib [2]. Посмотрим, всё ли у нас получится.

# Шаг 1: скачивание

---



The screenshot shows a Python IDE window titled '\*lection4runner.py - E:\Works\Victor\Students\infres\2020\Lecture4\lection4runner.py (3.7.2)\*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
import urllib.request

file = urllib.request.urlopen(
    'https://www.minfin.ru/common/upload/library/2019/09/main/315-fz.docx')
data = file.read()
file2 = open('315-fz.docx', 'wb+')
file2.write(data)
file2.close()
file.close()
```

The status bar at the bottom right indicates 'Ln: 10 Col: 0'.

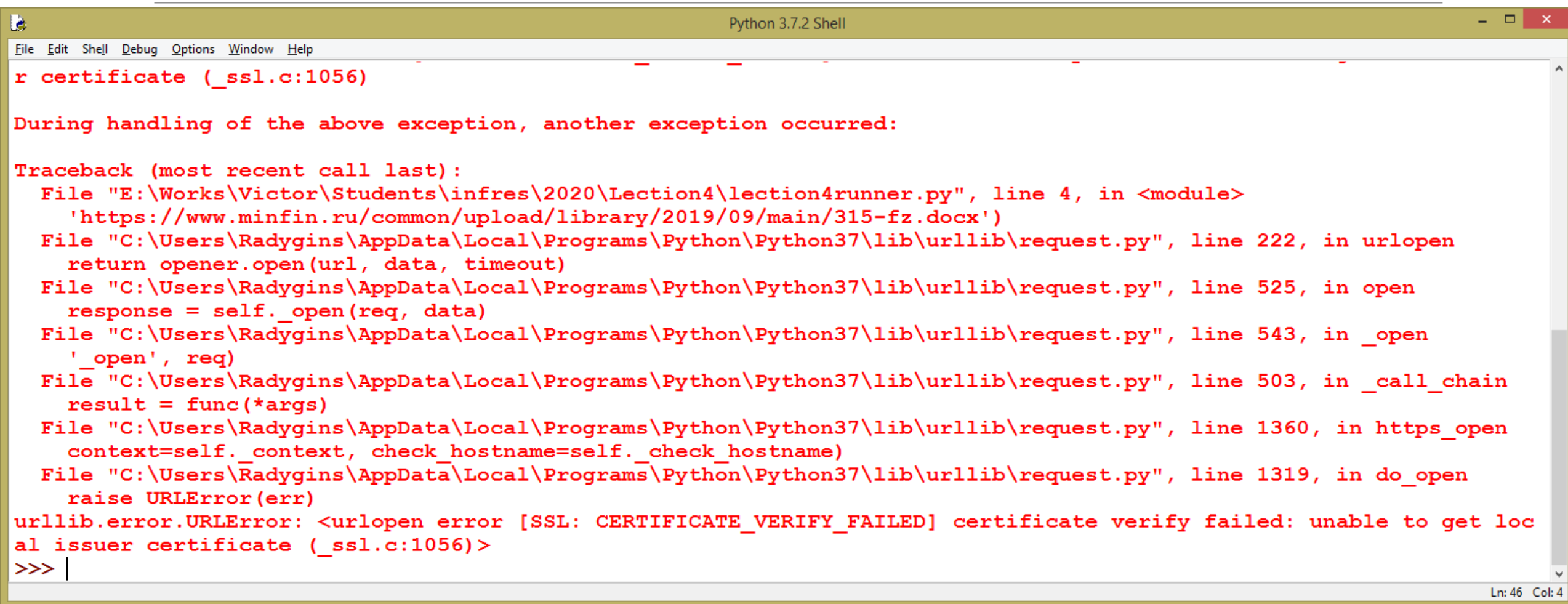
# Шаг 1: скачивание (текстом)

---

```
import urllib.request

file = urllib.request.urlopen(
    'https://www.minfin.ru/common/upload/library/2019/09/main/315-fz.docx')
data = file.read()
file2 = open('315-fz.docx', 'wb+')
file2.write(data)
file2.close()
file.close()
```

# Шаг 1: могут быть проблемы!



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
r certificate (_ssl.c:1056)

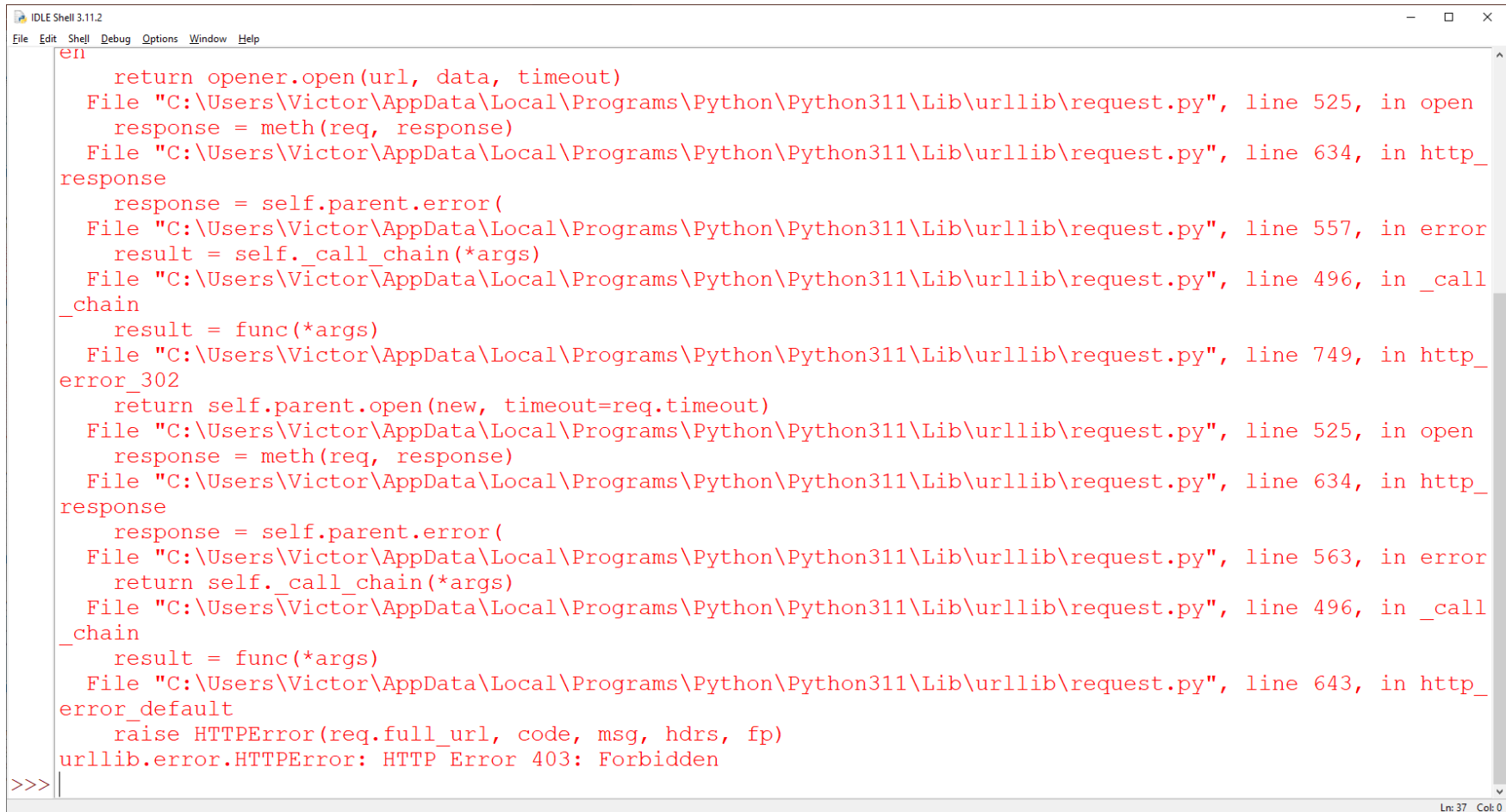
During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "E:\Works\Victor\Students\infres\2020\Lecton4\lection4runner.py", line 4, in <module>
    'https://www.minfin.ru/common/upload/library/2019/09/main/315-fz.docx')
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 222, in urlopen
    return opener.open(url, data, timeout)
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 525, in open
    response = self._open(req, data)
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 543, in _open
    '_open', req)
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 503, in _call_chain
    result = func(*args)
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 1360, in https_open
    context=self._context, check_hostname=self._check_hostname)
  File "C:\Users\Radygins\AppData\Local\Programs\Python\Python37\lib\urllib\request.py", line 1319, in do_open
    raise URLError(err)
urllib.error.URLError: <urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1056)>
>>> |
```

Ln: 46 Col: 4



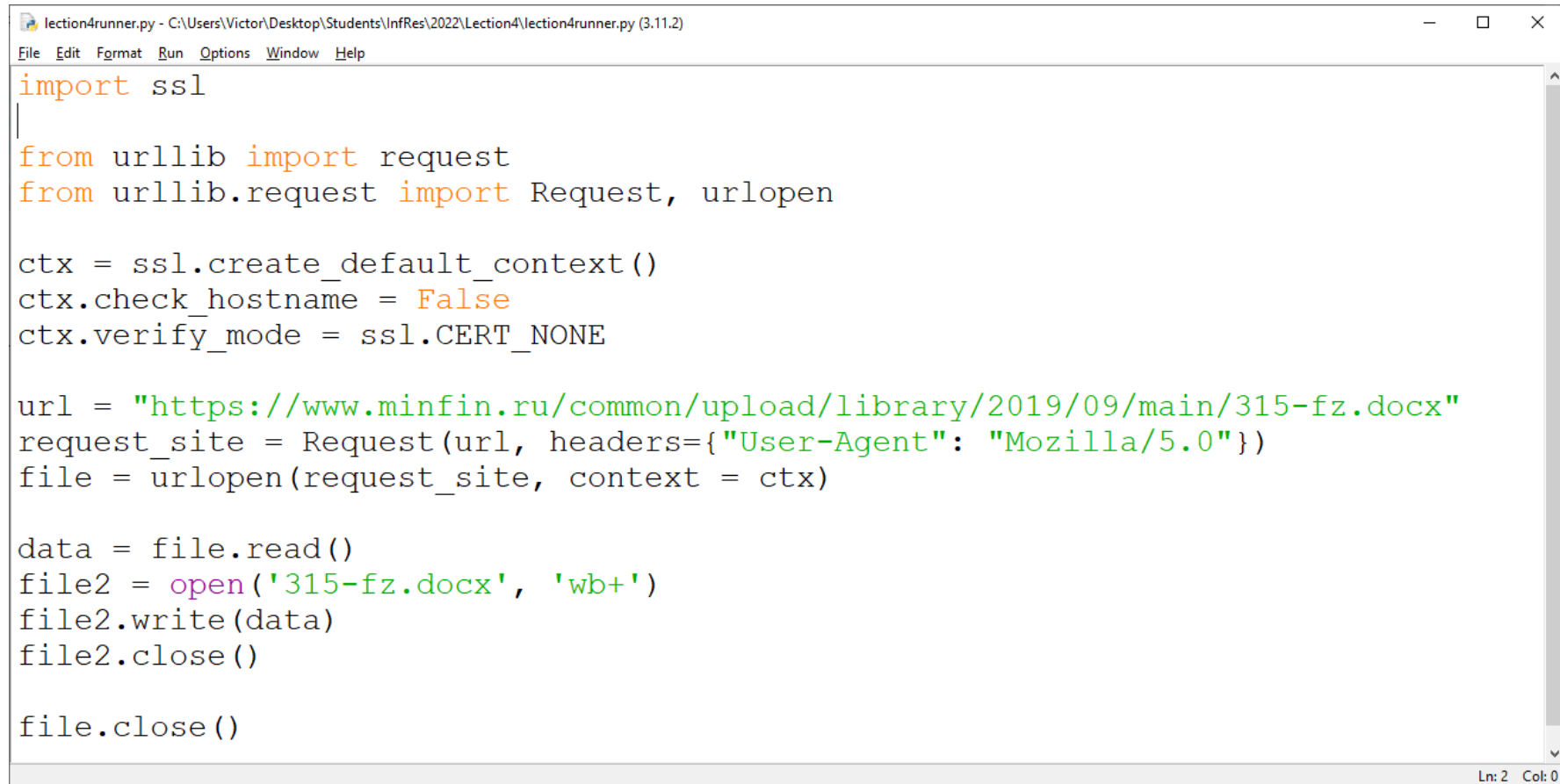
# Шаг 1: или могут быть проблемы!



```
en
    return opener.open(url, data, timeout)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 525, in open
    response = meth(req, response)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 634, in http_
response
    response = self.parent.error(
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 557, in error
    result = self._call_chain(*args)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 496, in _call
_chain
    result = func(*args)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 749, in http_
error_302
    return self.parent.open(new, timeout=req.timeout)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 525, in open
    response = meth(req, response)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 634, in http_
response
    response = self.parent.error(
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 563, in error
    return self._call_chain(*args)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 496, in _call
_chain
    result = func(*args)
File "C:\Users\Victor\AppData\Local\Programs\Python\Python311\Lib\urllib\request.py", line 643, in http_
error_default
    raise HTTPError(req.full_url, code, msg, hdrs, fp)
urllib.error.HTTPError: HTTP Error 403: Forbidden
>>>
```

Ln: 37 Col: 0

# Шаг 1: Исправим ситуацию



```
lection4runner.py - C:\Users\Victor\Desktop\Students\InfRes\2022\Lecture4\lection4runner.py (3.11.2)
File Edit Format Run Options Window Help
import ssl
|
from urllib import request
from urllib.request import Request, urlopen

ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE

url = "https://www.minfin.ru/common/upload/library/2019/09/main/315-fz.docx"
request_site = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(request_site, context = ctx)

data = file.read()
file2 = open('315-fz.docx', 'wb+')
file2.write(data)
file2.close()

file.close()
```

Ln: 2 Col: 0

# Шаг 1: Исправим ситуацию (текстом)

---

```
import ssl

from urllib import request

from urllib.request import Request,
urlopen

ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE
```

```
url = "https://www.minfin.ru/common/upload/library/2019/09/main/315-fz.docx"
request_site = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(request_site, context = ctx)

data = file.read()

file2 = open('315-fz.docx', 'wb+')
file2.write(data)
file2.close()

file.close()
```

## Шаг 2: превращение docx в текст

---

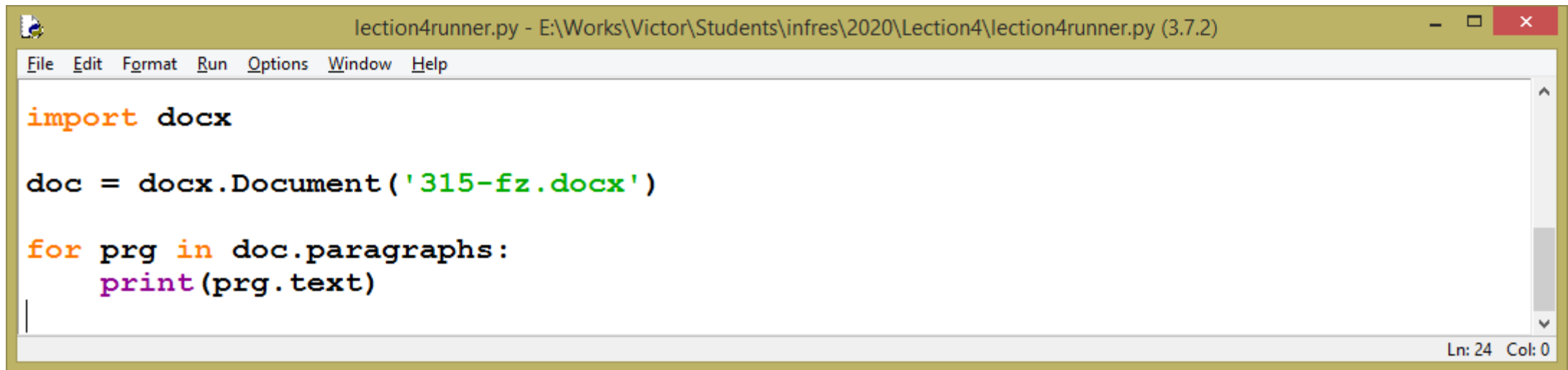
Теперь, когда мы добыли документ, нужно его превратить в Python-строки. Для этого нам потребуется библиотека, позволяющая выполнять разбор DOCX-документа. Для большинства языков характерна ситуация, когда для работы с DOCX-файлами существует две отдельных библиотеки: одна для «чтения» файлов, другая для создания файлов.

Так как нам нужно только «прочитать» DOCX-файл, то мы используем библиотеку `python-docx` [3]. Для её установки в Windows используйте в cmd, запущенном от имени администратора, команду:

**`/путь к питону/python.exe -m pip install python-docx`**

## Шаг 2: превращение docx в текст

---



The screenshot shows a Python IDE window titled "lection4runner.py - E:\Works\Victor\Students\infres\2020\Lecture4\lection4runner.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
import docx

doc = docx.Document('315-fz.docx')

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

The status bar at the bottom right indicates "Ln: 24 Col: 0".

## Шаг 2: превращение docx в текст (текстом)

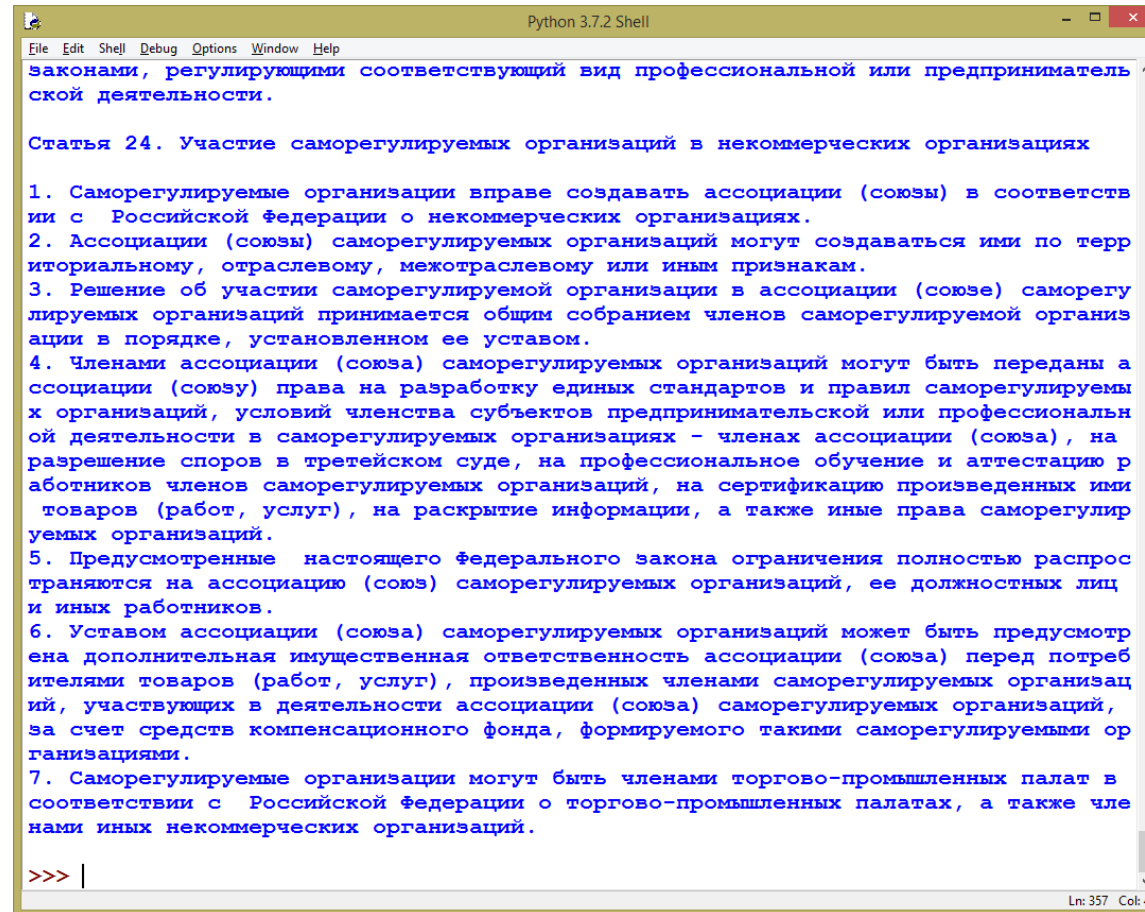
---

```
import docx

doc = docx.Document('315-fz.docx')

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

# Шаг 2: превращение docx в текст



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
законами, регулирующими соответствующий вид профессиональной или предпринимательской деятельности.

Статья 24. Участие саморегулируемых организаций в некоммерческих организациях

1. Саморегулируемые организации вправе создавать ассоциации (союзы) в соответствии с Российской Федерации о некоммерческих организациях.
2. Ассоциации (союзы) саморегулируемых организаций могут создаваться ими по территориальному, отраслевому, межотраслевому или иным признакам.
3. Решение об участии саморегулируемой организации в ассоциации (союзе) саморегулируемых организаций принимается общим собранием членов саморегулируемой организации в порядке, установленном ее уставом.
4. Членами ассоциации (союза) саморегулируемых организаций могут быть переданы ассоциации (союзу) права на разработку единых стандартов и правил саморегулируемых организаций, условий членства субъектов предпринимательской или профессиональной деятельности в саморегулируемых организациях - членах ассоциации (союза), на разрешение споров в третейском суде, на профессиональное обучение и аттестацию работников членов саморегулируемых организаций, на сертификацию произведенных ими товаров (работ, услуг), на раскрытие информации, а также иные права саморегулируемых организаций.
5. Предусмотренные настоящего федерального закона ограничения полностью распространяются на ассоциацию (союз) саморегулируемых организаций, ее должностных лиц и иных работников.
6. Уставом ассоциации (союза) саморегулируемых организаций может быть предусмотрена дополнительная имущественная ответственность ассоциации (союза) перед потребителями товаров (работ, услуг), произведенных членами саморегулируемых организаций, участвующих в деятельности ассоциации (союза) саморегулируемых организаций, за счет средств компенсационного фонда, формируемого такими саморегулируемыми организациями.
7. Саморегулируемые организации могут быть членами торгово-промышленных палат в соответствии с Российской Федерации о торгово-промышленных палатах, а также членами иных некоммерческих организаций.

>>> |
```

Ln: 357 Col: 4

## Шаг 3: поиск нужной статьи

---

На шаге 2 мы научились получать текст из DOCX-файла. Библиотека `python-docx` представляет DOCX-файл в виде набора параграфов, у каждого из которых при помощи атрибута `text` можно посмотреть текстовое содержимое.

Но мы не знаем с какого параграфа начинается статья 7. И не знаем сколько она занимает. Нам только известно, что данная статья начинается с фразы «Статья 7», а следующая статья начинается с фразы «Статья 8».

Используем для поиска регулярные выражения.



# Регулярные выражения в Python

---

На 2 курсе мы проходили понятие регулярных выражений и то, как ими пользоваться в Oracle SQL. Данный эффективный механизм есть в большинстве языков программирования. Не исключением является и язык Python.

Для подключения регулярных выражений в Python используется модуль `re`:

```
import re
```

# ОСНОВНЫЕ МЕТОДЫ

---

`re.match(выражение, строка, флаги)`

`re.search(выражение, строка, флаги)`

`re.findall(выражение, строка, флаги)`

`re.split(выражение, строка, число разбиений, флаги)`

`re.sub(выражение, замена, строка, счетчик, флаги)`

# Пример

```
*lection4runner.py - E:\Works\Victor\Students\infres\2020\Lecture4\lection4runn...
File Edit Format Run Options Window Help

import re

txt = """This is text!
This is very simple text!"""
res = re.match(r'[Tt][A-Za-z]*', txt)
print(res)
res = re.match(r't[a-z]*', txt, re.I)
print(res)
res = re.match(r't[a-z]*', txt)
print(res)
res = re.search(r't[a-z]*', txt)
print(res)
res = re.findall(r't[a-z]*', txt, re.I)
print(res)
res = re.split(r't[a-z]*', txt, 0, re.I)
print(res)
res = re.sub(r't[a-z]*', 'T-word', txt, 0, re.I)
print(res)
```

Ln: 42 Col: 10

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: E:\Works\Victor\Students\infres\2020\Lecture4\lection4runner.py ==
<re.Match object; span=(0, 4), match='This'>
<re.Match object; span=(0, 4), match='This'>
None
<re.Match object; span=(8, 12), match='text'>
['This', 'text', 'This', 'text']
['', ' is ', '!\n', ' is very simple ', '!!']
T-word is T-word!
T-word is very simple T-word!
>>> |
```

Ln: 13 Col: 4

# Пример (текстом)

---

```
import re

txt = """This is text!
This is very simple text!"""

res = re.match(r'[Tt][A-Za-z]*', txt)
print(res)

res = re.match(r't[a-z]*', txt, re.I)
print(res)

res = re.match(r't[a-z]*', txt)
print(res)
```

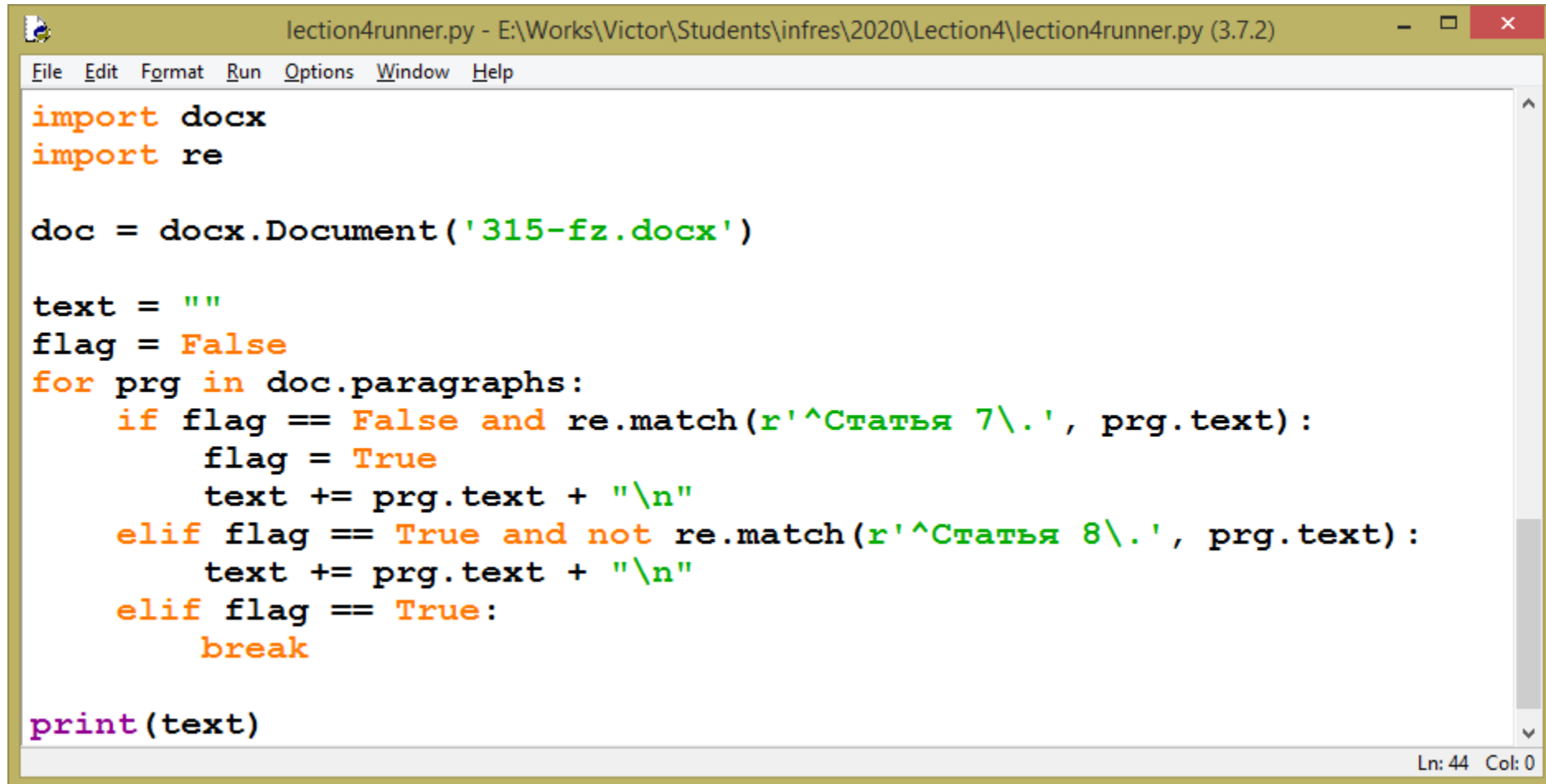
```
res = re.search(r't[a-z]*', txt)
print(res)

res = re.findall(r't[a-z]*', txt, re.I)
print(res)

res = re.split(r't[a-z]*', txt, 0, re.I)
print(res)

res = re.sub(r't[a-z]*', 'T-word', txt, 0, re.I)
print(res)
```

# Шаг 3: поиск нужной статьи



```
lection4runner.py - E:\Works\Victor\Students\infres\2020\Lecture4\lection4runner.py (3.7.2)
File Edit Format Run Options Window Help

import docx
import re

doc = docx.Document('315-fz.docx')

text = ""
flag = False
for prg in doc.paragraphs:
    if flag == False and re.match(r'^Статья 7\.', prg.text):
        flag = True
        text += prg.text + "\n"
    elif flag == True and not re.match(r'^Статья 8\.', prg.text):
        text += prg.text + "\n"
    elif flag == True:
        break

print(text)
```

Ln: 44 Col: 0

## Шаг 3: поиск нужной статьи (текстом)

---

```
import docx
import re

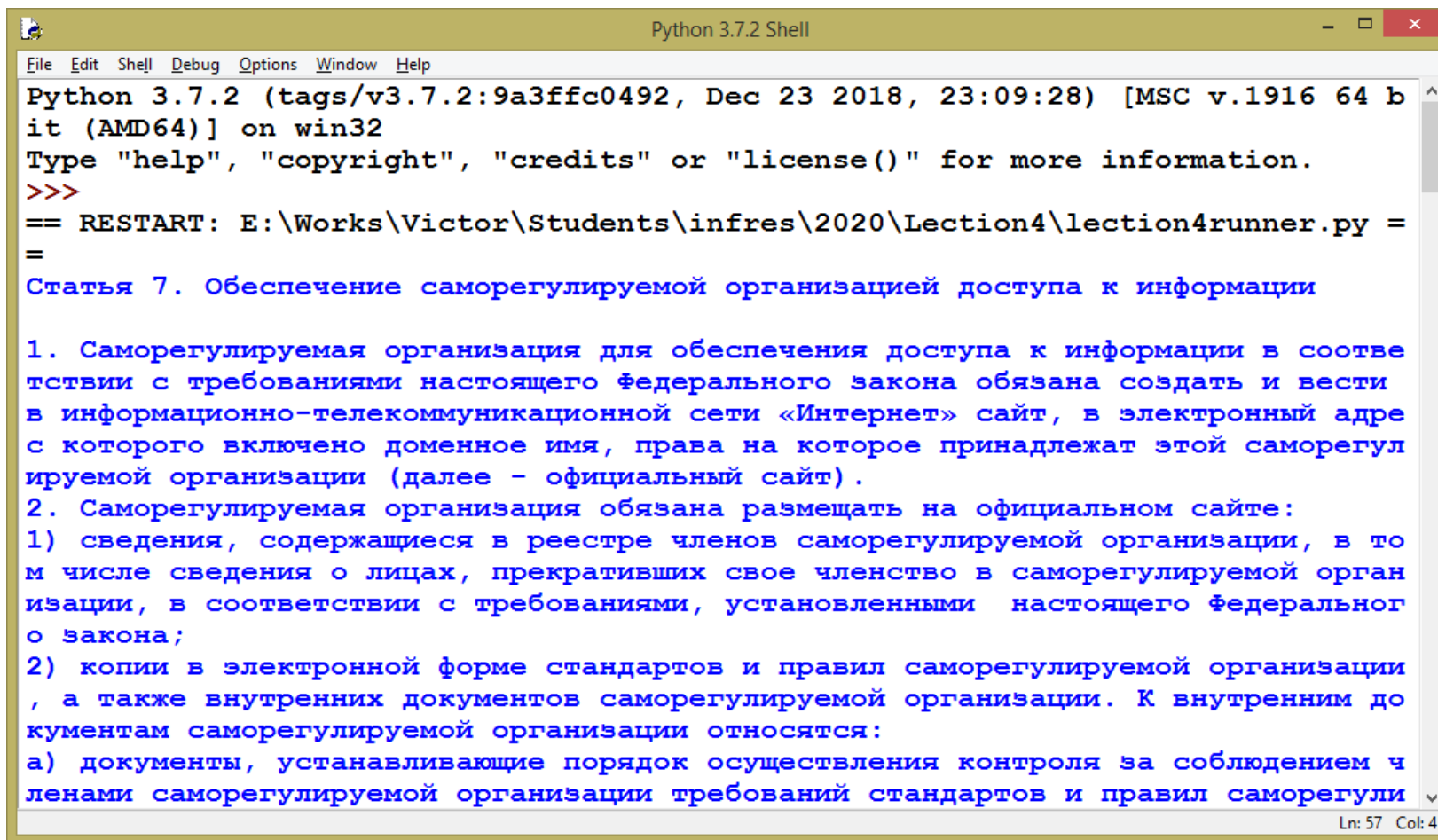
doc = docx.Document('315-fz.docx')

text = ""
flag = False
for prg in doc.paragraphs:
```

```
    if flag == False and re.match(r'^Статья 7\.', prg.text):
        flag = True
        text += prg.text + "\n"
    elif flag == True and not re.match(r'^Статья 8\.', prg.text):
        text += prg.text + "\n"
    elif flag == True:
        break

print(text)
```

# Шаг 3: поиск нужной статьи



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: E:\Works\Victor\Students\infres\2020\Lecture4\lecture4runner.py =
=
Статья 7. Обеспечение саморегулируемой организацией доступа к информации

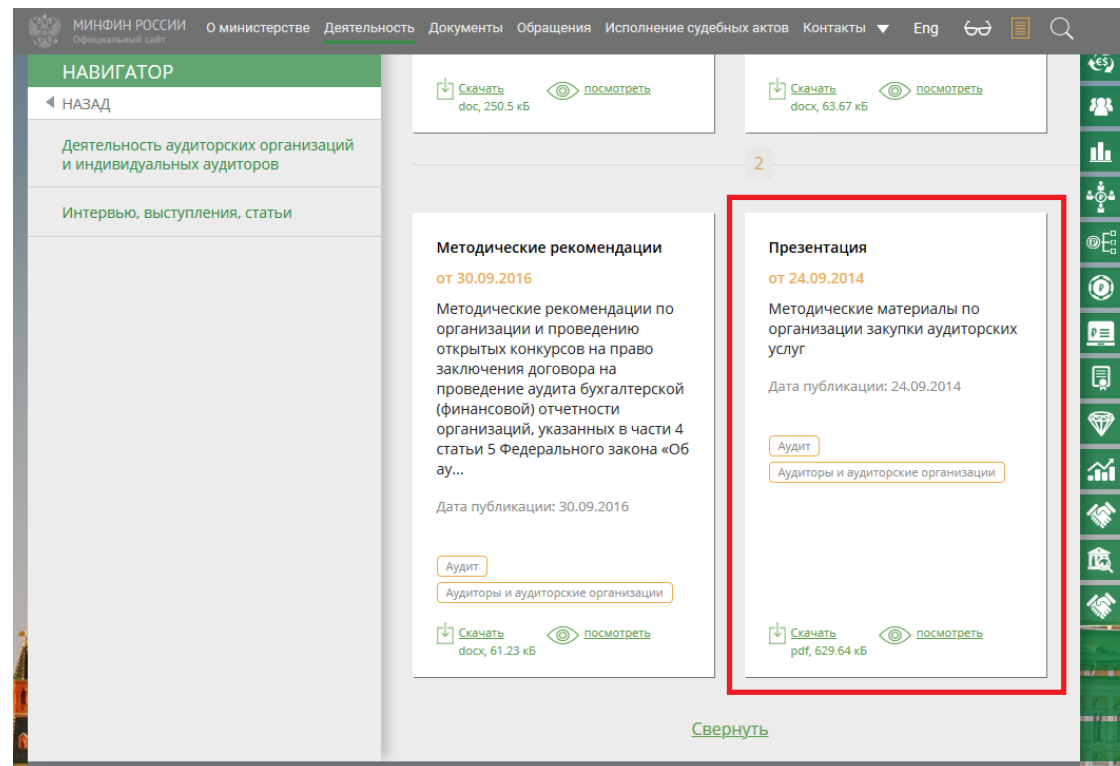
1. Саморегулируемая организация для обеспечения доступа к информации в соотве
тствии с требованиями настоящего федерального закона обязана создать и вести
в информационно-телекоммуникационной сети «Интернет» сайт, в электронный адре
с которого включено доменное имя, права на которое принадлежат этой саморегул
ируемой организации (далее – официальный сайт) .
2. Саморегулируемая организация обязана размещать на официальном сайте:
1) сведения, содержащиеся в реестре членов саморегулируемой организации, в то
м числе сведения о лицах, прекративших свое членство в саморегулируемой орган
изации, в соответствии с требованиями, установленными настоящего федеральног
о закона;
2) копии в электронной форме стандартов и правил саморегулируемой организации
, а также внутренних документов саморегулируемой организации. К внутренним до
кументам саморегулируемой организации относятся:
а) документы, устанавливающие порядок осуществления контроля за соблюдением ч
ленами саморегулируемой организации требований стандартов и правил саморегули
Ln: 57 Col: 4
```

# Задача 2

На сайте Министерства финансов РФ (<https://www.minfin.ru>) [2] выложена презентация «Методические материалы по организации закупки аудиторских услуг».

Задача: напишите программу, скачивающую данную презентацию и распечатывающую 4 слайд в виде обычного текста.

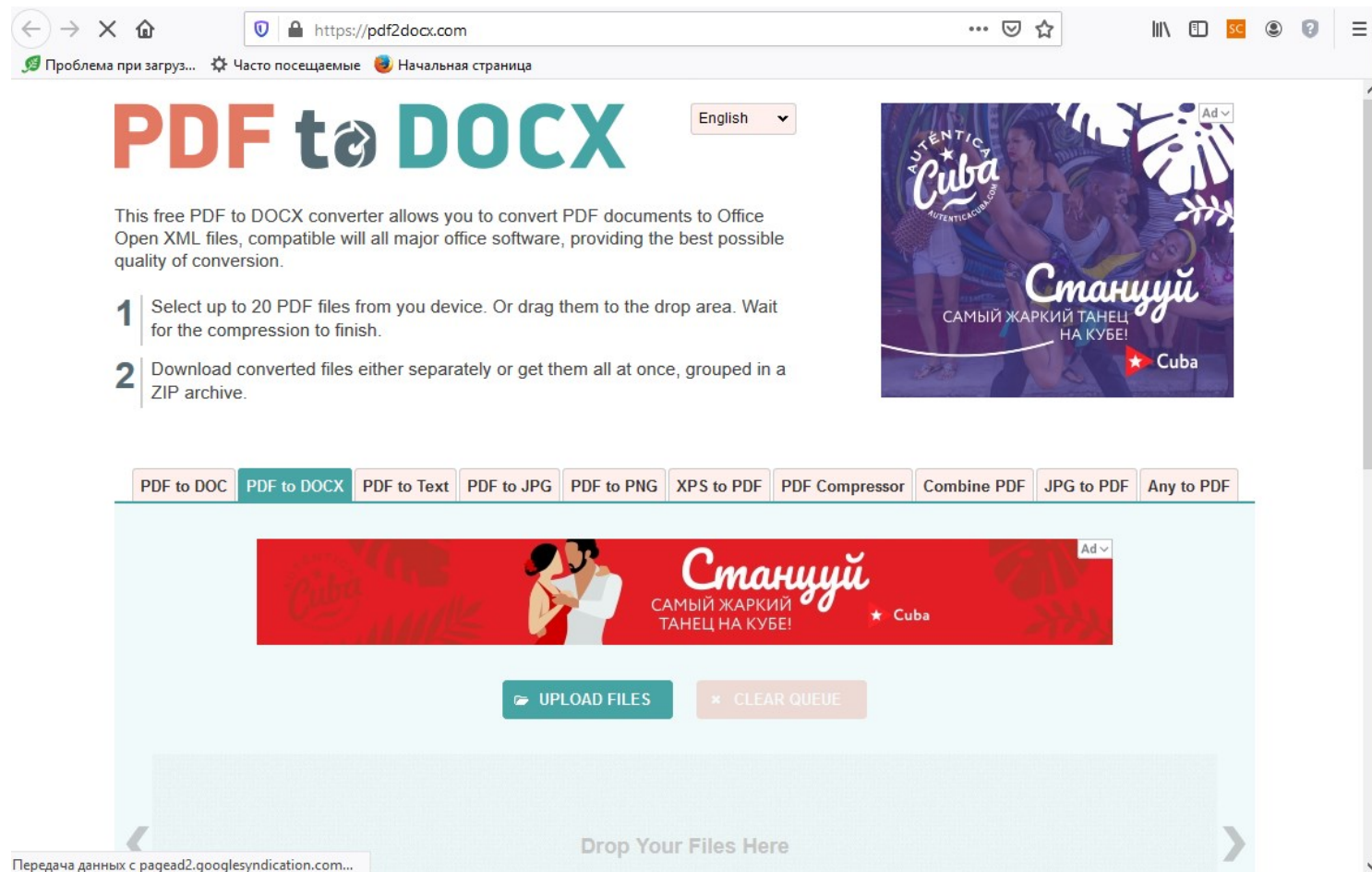
Проблема в том, что файл в формате PDF и старый подход не поможет.





# Если это нужно сделать один раз

Если Вам нужно добыть текст из PDF один раз, то не надо писать программы. Используйте бесплатные онлайн-средства:



# Шаг 1: как в задаче 1



```
lection4runner.py - C:\Users\Victor\Desktop\Students\InfRes\2022\Lesson4\lection4runner.py (3.11.2)
File Edit Format Run Options Window Help

import ssl

from urllib import request
from urllib.request import Request, urlopen

ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE

url = "https://minfin.gov.ru/common/upload/library/2018/11/main/Stuk_071118.pdf"
request_site = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(request_site, context = ctx)

data = file.read()
file2 = open('pres.pdf', 'wb+')
file2.write(data)
file2.close()

file.close()
```

Ln: 19 Col: 12

# Шаг 1: как в задаче 1 (текстом)

---

```
import ssl

from urllib import request

from urllib.request import Request,
urlopen

ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE

url = "https://minfin.gov.ru/common/upload/library/2018/11/main/Stuk_071118.pdf"
request_site = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(request_site, context = ctx)

data = file.read()

file2 = open('pres.pdf', 'wb+')
file2.write(data)
file2.close()

file.close()
```

## Шаг 2: превращение pdf в текст

---

Теперь, когда мы добыли документ, нужно его превратить в Python-строки. Для этого нам потребуется библиотека, позволяющая выполнять разбор PDF-документа. Мы используем библиотеку pdfminer [5]. Для её установки в Windows используйте в cmd, запущенном от имени администратора, команду:

**`/путь к питону/python.exe -m pip install pdfminer.six`**

# Шаг 2: превращение pdf целиком в текстовый файл

```
Lesson4runner.py - C:\Users\Victor\Desktop\Students\InfRes\2022\Lecture4\Lecture4runner.py (3.11.2)
File Edit Format Run Options Window Help

pdf_document = "pres.pdf"

from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage

file_txt = open('pres.txt', 'wb+')

resource_manager = PDFResourceManager()
converter = TextConverter(resource_manager, file_txt)
page_interpreter = PDFPageInterpreter(resource_manager, converter)

with open(pdf_document, 'rb') as fh:
    for page in PDFPage.get_pages(fh,
                                  caching=True,
                                  check_extractable=True):
        page_interpreter.process_page(page)

converter.close()
file_txt.close()
```

Ln: 28 Col: 34

# Шаг 2: превращение pdf целиком в текстовый файл (текстом)

```
pdf_document = "pres.pdf"
```

```
from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage
```

```
file_txt = open('pres.txt', 'wb+')
```

```
resource_manager = PDFResourceManager()
converter = TextConverter(resource_manager, file_txt)
page_interpreter = PDFPageInterpreter(resource_manager,
converter)
```

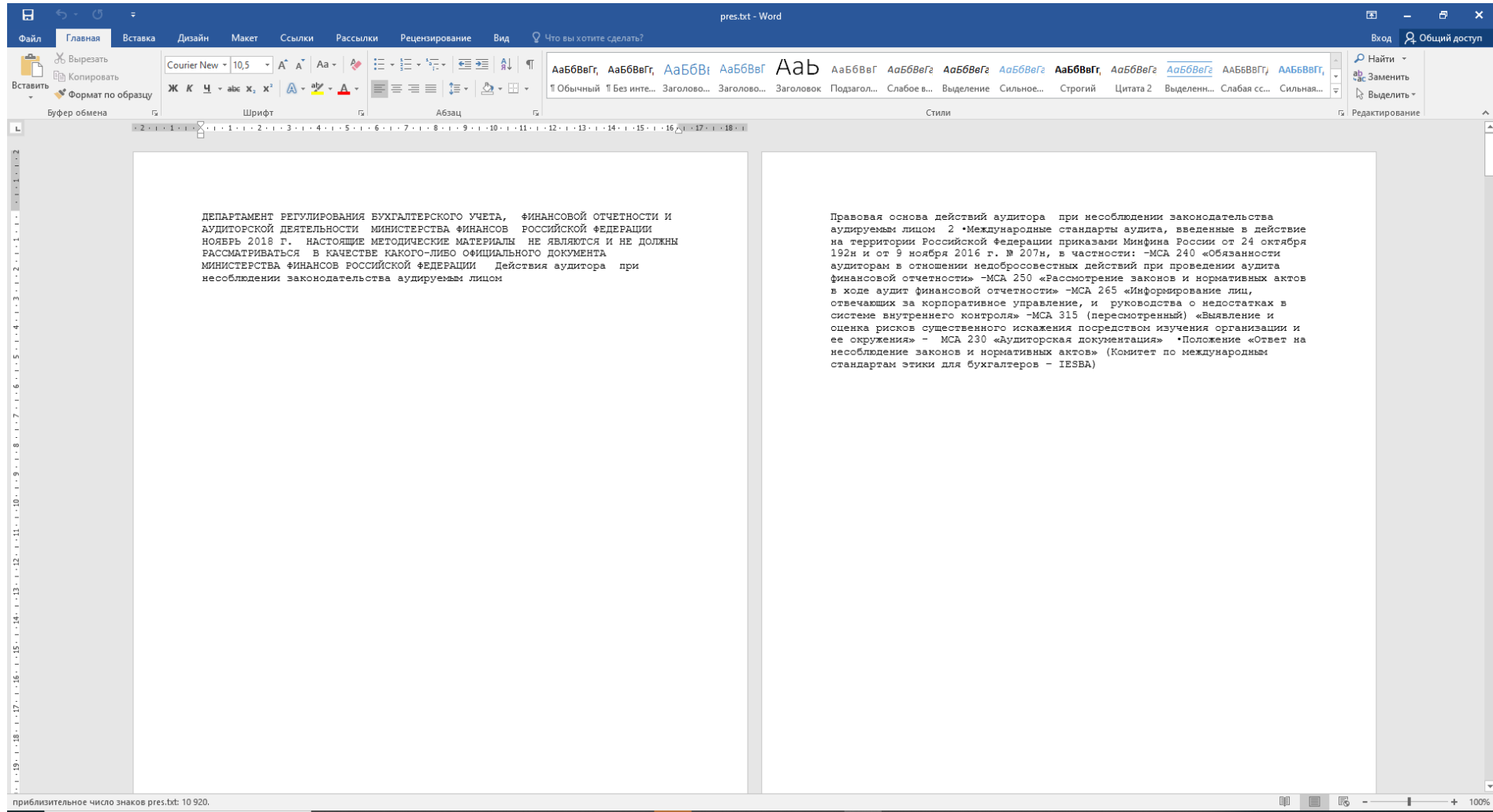
```
with open(pdf_document, 'rb') as fh:
```

```
    for page in PDFPage.get_pages(fh,
                                    caching=True,
                                    check_extractable=True):
        page_interpreter.process_page(page)
```

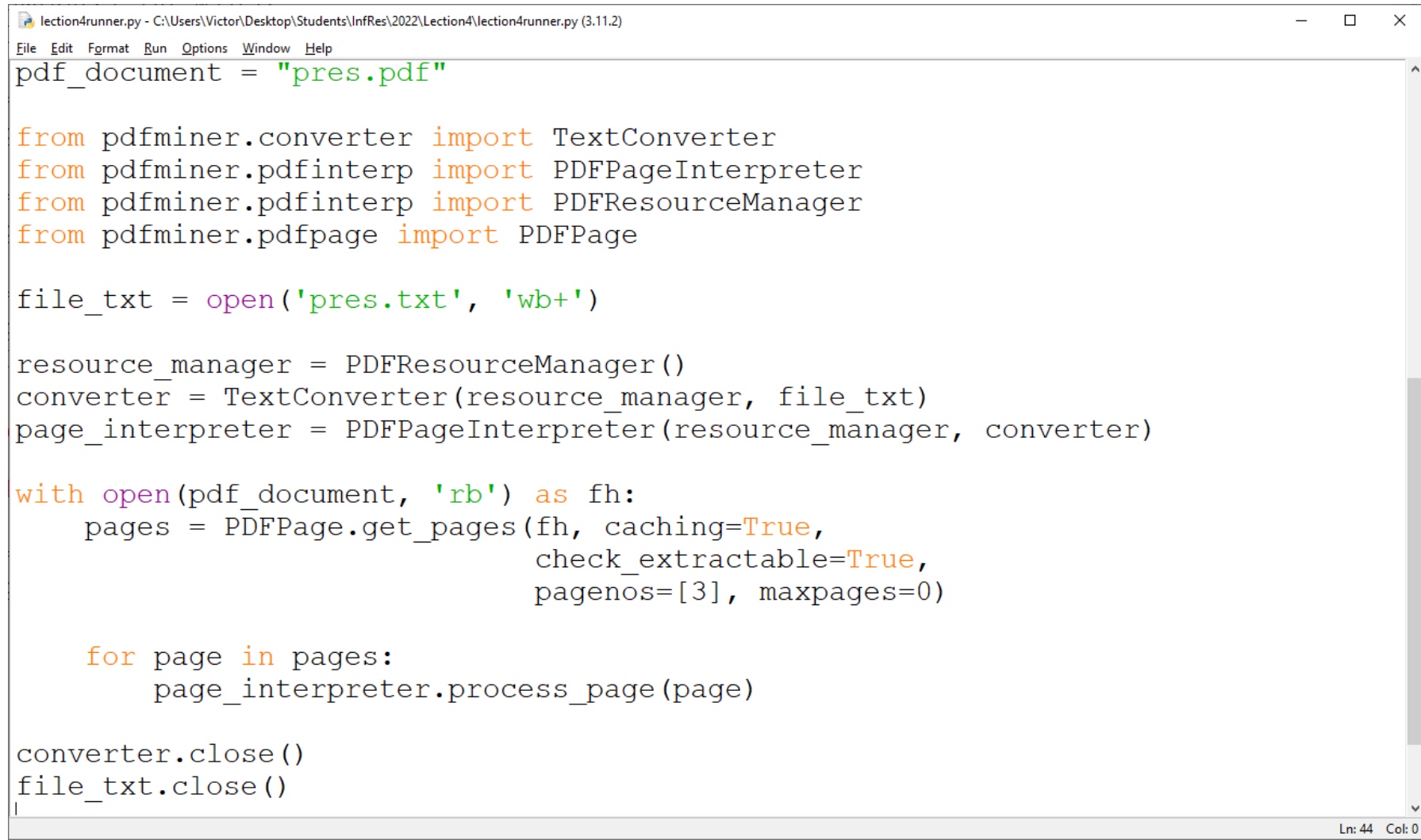
```
converter.close()
```

```
file_txt.close()
```

# Шаг 2: превращение pdf целиком в текстовый файл



## Шаг 2: только 4 страница

A screenshot of a Python script editor window. The title bar shows the file path: 'C:\Users\Victor\Desktop\Students\InfRes\2022\Lecture4\Lecture4runner.py (3.11.2)'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The script content is as follows:

```
pdf_document = "pres.pdf"

from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage

file_txt = open('pres.txt', 'wb+')

resource_manager = PDFResourceManager()
converter = TextConverter(resource_manager, file_txt)
page_interpreter = PDFPageInterpreter(resource_manager, converter)

with open(pdf_document, 'rb') as fh:
    pages = PDFPage.get_pages(fh, caching=True,
                              check_extractable=True,
                              pagenos=[3], maxpages=0)

    for page in pages:
        page_interpreter.process_page(page)

converter.close()
file_txt.close()
```

The status bar at the bottom right indicates 'Ln: 44 Col: 0'.



## Шаг 2: только 4 страница (текстом)

```
pdf_document = "pres.pdf"
```

```
from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage
```

```
file_txt = open('pres.txt', 'wb+')
```

```
resource_manager = PDFResourceManager()
converter = TextConverter(resource_manager, file_txt)
page_interpreter = PDFPageInterpreter(resource_manager,
converter)
```

```
with open(pdf_document, 'rb') as fh:
```

```
    pages = PDFPage.get_pages(fh, caching=True,
                                check_extractable=True,
                                pagenos=[3], maxpages=0)
```

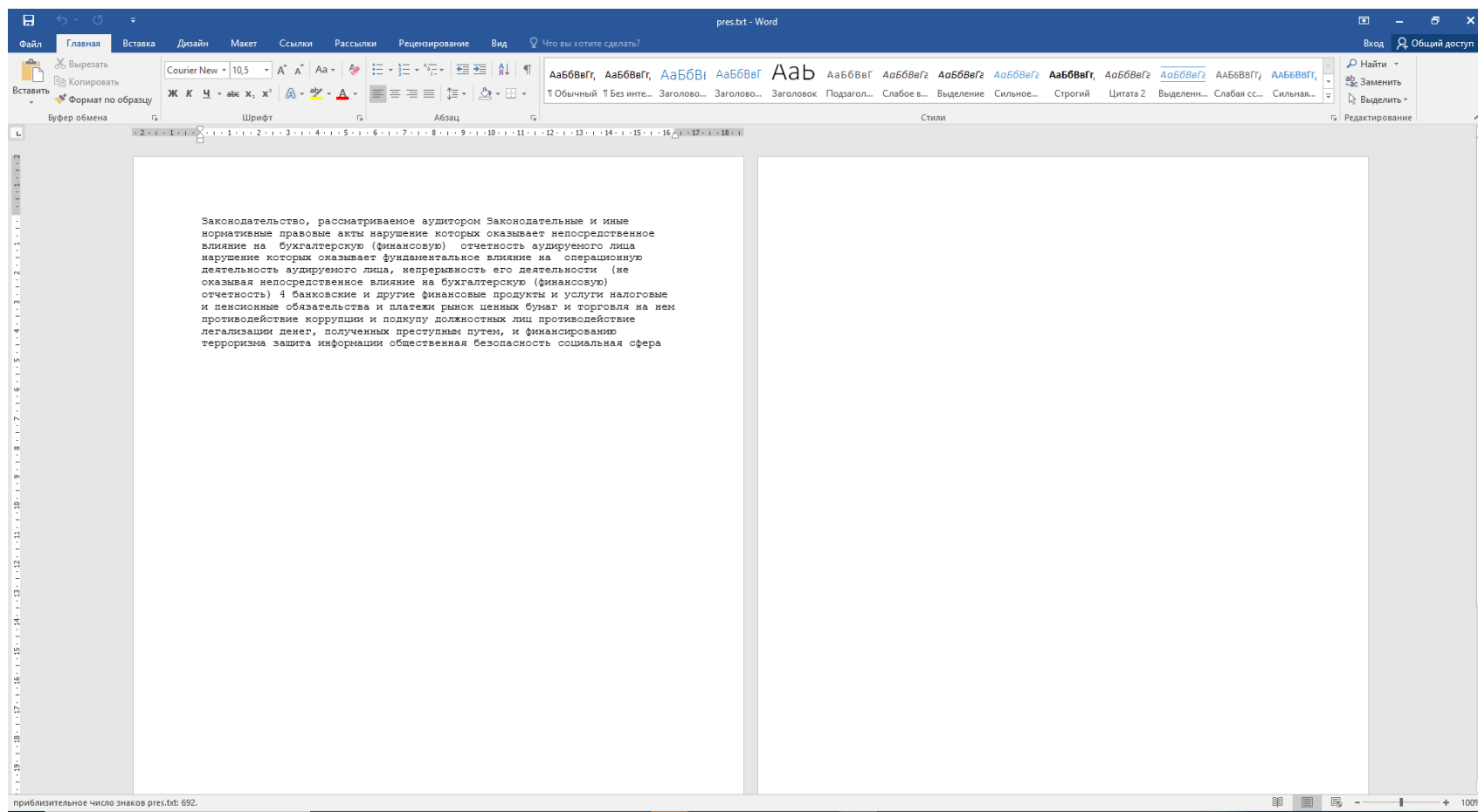
```
    for page in pages:
```

```
        page_interpreter.process_page(page)
```

```
converter.close()
```

```
file_txt.close()
```

# Шаг 2: только 4 страница



# Шаг 2: только 4 страница сразу на экран

```
lecture4runner.py - C:\Users\Victor\Desktop\Students\InfRes\2022\Lecture4\lecture4runner.py (3.11.2)
File Edit Format Run Options Window Help

pdf_document = "pres.pdf"

from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage
import io

string = io.StringIO()
resource_manager = PDFResourceManager()
converter = TextConverter(resource_manager, string)

page_interpreter = PDFPageInterpreter(resource_manager, converter)

with open(pdf_document, 'rb') as fh:
    pages = PDFPage.get_pages(fh, caching=True,
                              check_extractable=True,
                              pagenos=[3], maxpages=0)

    for page in pages:
        page_interpreter.process_page(page)

text = string.getvalue()
print(text)
converter.close()
string.close()
```

Ln: 47 Col: 0

# Шаг 2: только 4 страница сразу на экран (ТЕКСТОМ)

```
pdf_document = "pres.pdf"
```

```
from pdfminer.converter import TextConverter
```

```
from pdfminer.pdfinterp import PDFPageInterpreter
```

```
from pdfminer.pdfinterp import PDFResourceManager
```

```
from pdfminer.pdfpage import PDFPage
```

```
import io
```

```
string = io.StringIO()
```

```
resource_manager = PDFResourceManager()
```

```
converter = TextConverter(resource_manager, string)
```

```
page_interpreter = PDFPageInterpreter(resource_manager, converter)
```

```
with open(pdf_document, 'rb') as fh:
```

```
    pages = PDFPage.get_pages(fh, caching=True,
```

```
                               check_extractable=True,
```

```
                               pagenos=[3], maxpages=0)
```

```
    for page in pages:
```

```
        page_interpreter.process_page(page)
```

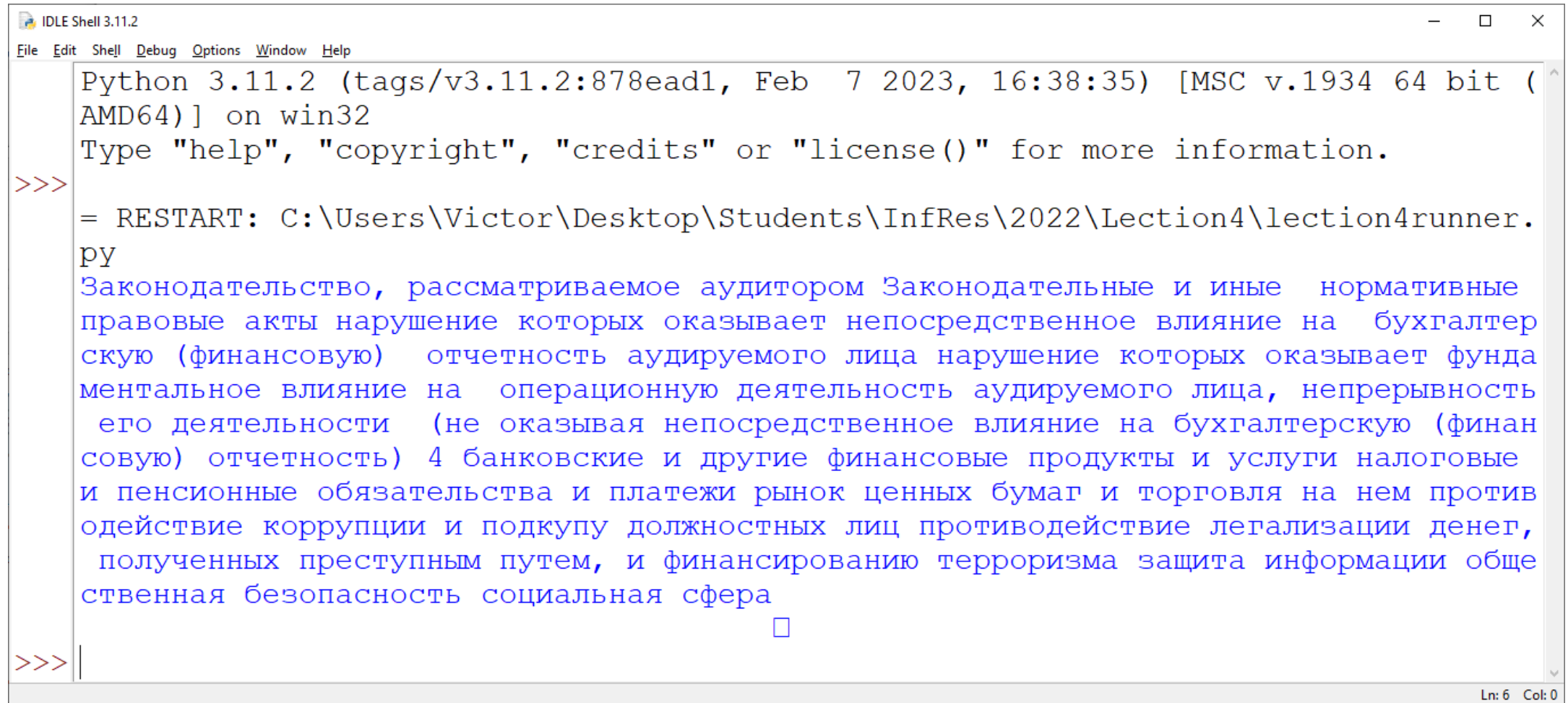
```
text = string.getvalue()
```

```
print(text)
```

```
converter.close()
```

```
string.close()
```

# Шаг 2: только 4 страница сразу на экран



```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Victor\Desktop\Students\InfRes\2022\Lecture4\lecture4runner.py
Законодательство, рассматриваемое аудитором Законодательные и иные нормативные
правовые акты нарушение которых оказывает непосредственное влияние на бухгалтер
скую (финансовую) отчетность аудируемого лица нарушение которых оказывает фунда
ментальное влияние на операционную деятельность аудируемого лица, непрерывность
его деятельности (не оказывая непосредственное влияние на бухгалтерскую (финан
совую) отчетность) 4 банковские и другие финансовые продукты и услуги налоговые
и пенсионные обязательства и платежи рынок ценных бумаг и торговля на нем против
одействие коррупции и подкупу должностных лиц противодействие легализации денег,
полученных преступным путем, и финансированию терроризма защита информации обще
ственная безопасность социальная сфера
>>>
```

Ln: 6 Col: 0

# Часть 2

---

РАСПОЗНАВАНИЕ ТЕКСТА

# Работа со скан-копиями

---

Многие документы размещаются в сети интернет в виде скан-копий. В этом случае для получение их содержимого в машинно-обрабатываемом виде необходимо применять программы для распознавания текста.

Наиболее известной программой для распознавания текста является ABBYY FineReader. Но у данной технологии есть ряд недостатков, начиная от стоимости и заканчивая отсутствием многоплатформенности и закрытостью кода.

Хорошей альтернативой ABBYY FineReader является свободная технология Tesseract [6].

Посмотрим, как её можно использовать в Python.

# Установка Tesseract

---

Tesseract – это не пакет для Python, а самостоятельный продукт. Поэтому для работы с Tesseract в Python надо установить 3 вещи:

- 1) Tesseract;
- 2) Русский словарь для Tesseract;
- 3) Python-пакет для Tesseract.



# Установка Tesseract для Windows

---

Полноценно работает на данный момент под Windows Tesseract версии 3. Инсталляционный пакет для Windows можно скачать из разных источников. Например, из [7].

После установки данного пакета потребуется добавить Русский словарь (если он не установился вместе с пакетом). Словари для Tesseract также доступны из многих источников, например [8].

Русские словари надо разместить в папке tessdata. Данная папка размещена в папке, в которой установился Tesseract (обычно, C:\Program Files (x86)\Tesseract-OCR).

# Библиотека Tesseract для Python

---

Для Python есть несколько библиотек Tesseract. В данной лекции я рекомендую pytesseract [9].

Для её установки в Windows используйте в cmd, запущенном от имени администратора, команду:

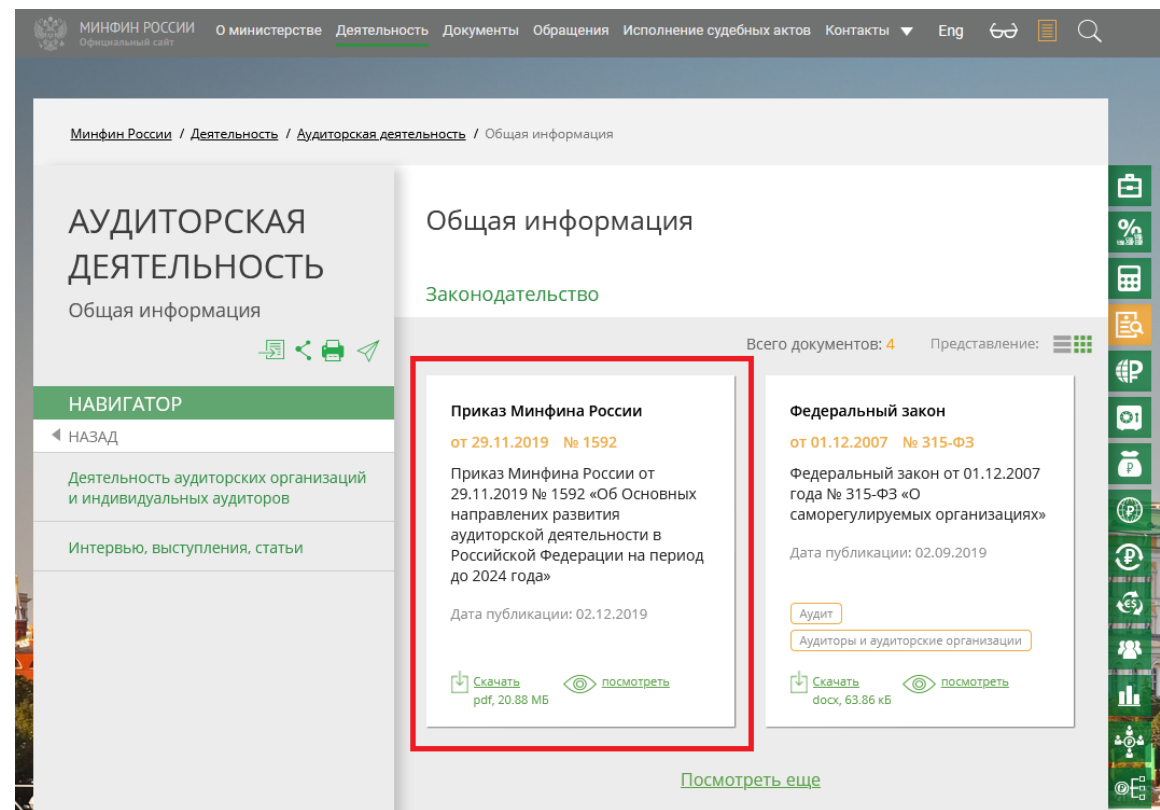
**/путь к питону/python.exe -m pip install pytesseract**

# Задача 3

На сайте Министерства финансов РФ (<https://www.minfin.ru>) [1] выложен Приказ Минфина России от 29.11.2019 № 1592 «Об основных направлениях развития аудиторской деятельности в Российской Федерации на период до 2024 года».

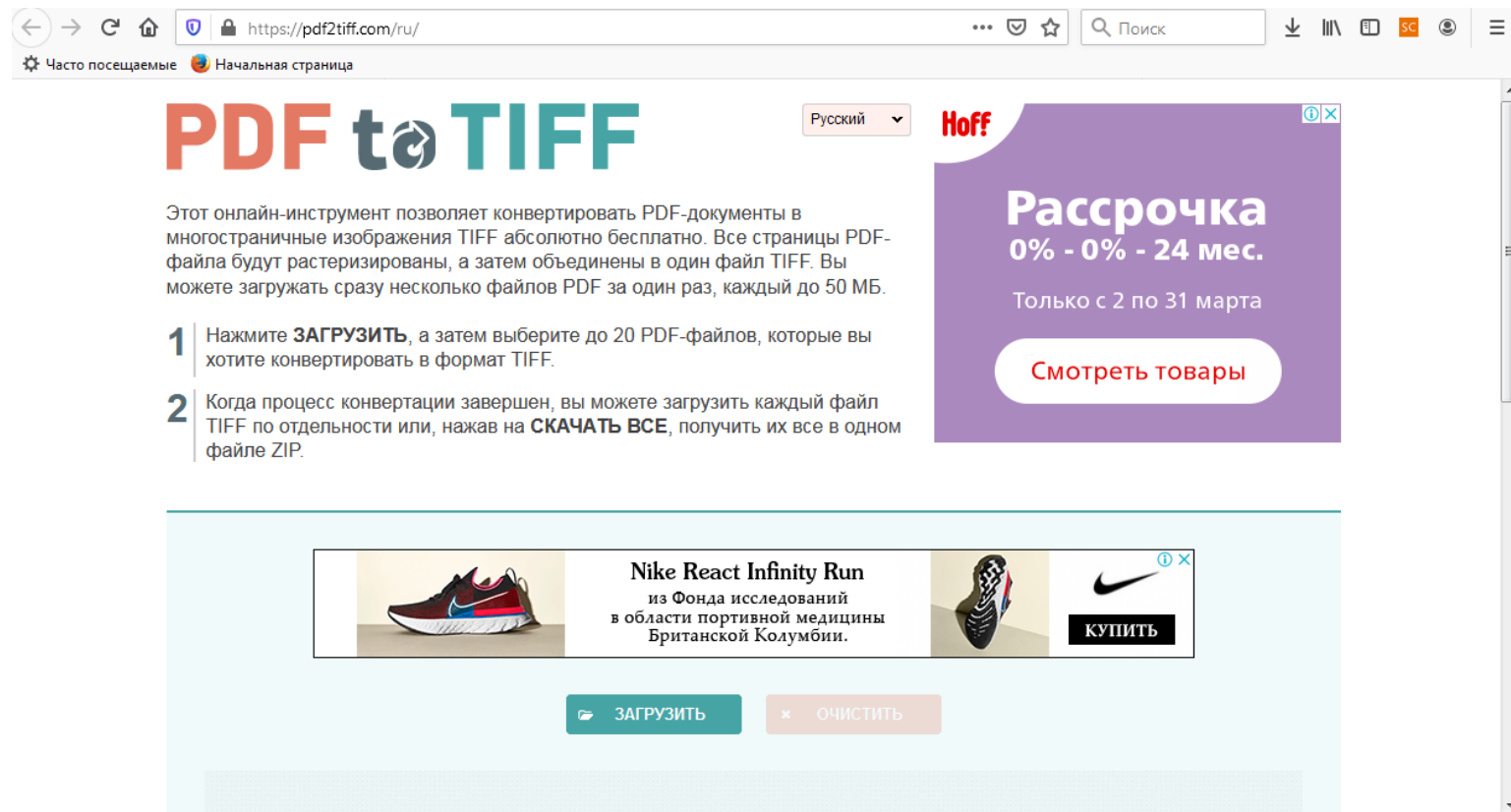
Предположим, что мы скачали приказ и сохранили его первую страницу в виде tiff-изображения в файле 29.11.2019\_1592.tiff.

Задача: напишите программу, распознающую содержимое данной страницы приказа и печатающую его на экран в виде обычного текста.



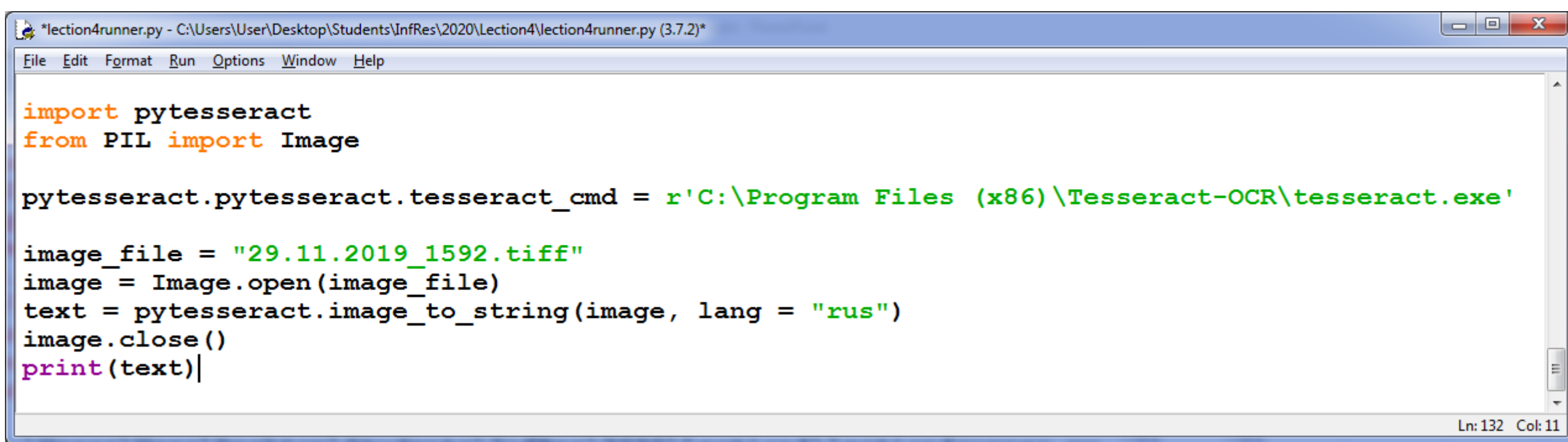
# Задача 3. Подготовка

Так как предполагается, что первую страницу мы как-то достали вручную и преобразовали её в tiff-изображение, то это придётся как-то сделать без программы. Например, с помощью онлайн-средств.



# Задача 3: программа

---



The image shows a screenshot of a Python IDE window titled "\*lection4runner.py - C:\Users\User\Desktop\Students\InfRes\2020\Lecture4\lection4runner.py (3.7.2)\*". The window has a menu bar with File, Edit, Format, Run, Options, Window, and Help. The main text area contains the following Python code:

```
import pytesseract
from PIL import Image

pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files (x86)\Tesseract-OCR\tesseract.exe'

image_file = "29.11.2019_1592.tiff"
image = Image.open(image_file)
text = pytesseract.image_to_string(image, lang = "rus")
image.close()
print(text)|
```

The status bar at the bottom right of the window indicates "Ln: 132 Col: 11".

# Задача 3: программа

---

```
import pytesseract
from PIL import Image

pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

image_file = "29.11.2019_1592.tiff"
image = Image.open(image_file)
text = pytesseract.image_to_string(image, lang = "rus")
image.close()
print(text)
```

# Результат

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\User\Desktop\Students\InfRes\2020\Lecture4\lecture4runner.py
МИНИСТЕРСТВО ФИНАНСОВ РОССИЙСКОЙ ФЕДЕРАЦИИ

>- .

(минфин РОССИИ)

ПРИКАЗ

№ , „599 3

Москва

Об Основных направлениях развития аудиторской деятельности в
Российской Федерации на период до 2024 года

В целях определения приоритетных направлений дальнейшего развития
аудиторской деятельности в Российской Федерации п р и к а з ы в а ю:

1. Утвердить прилагаемые Основные направления развития аудиторской
деятельности в Российской Федерации на период до 2024 года, одобренные
решением Совета по аудиторской деятельности от 20 ноября 2019 года № 50.

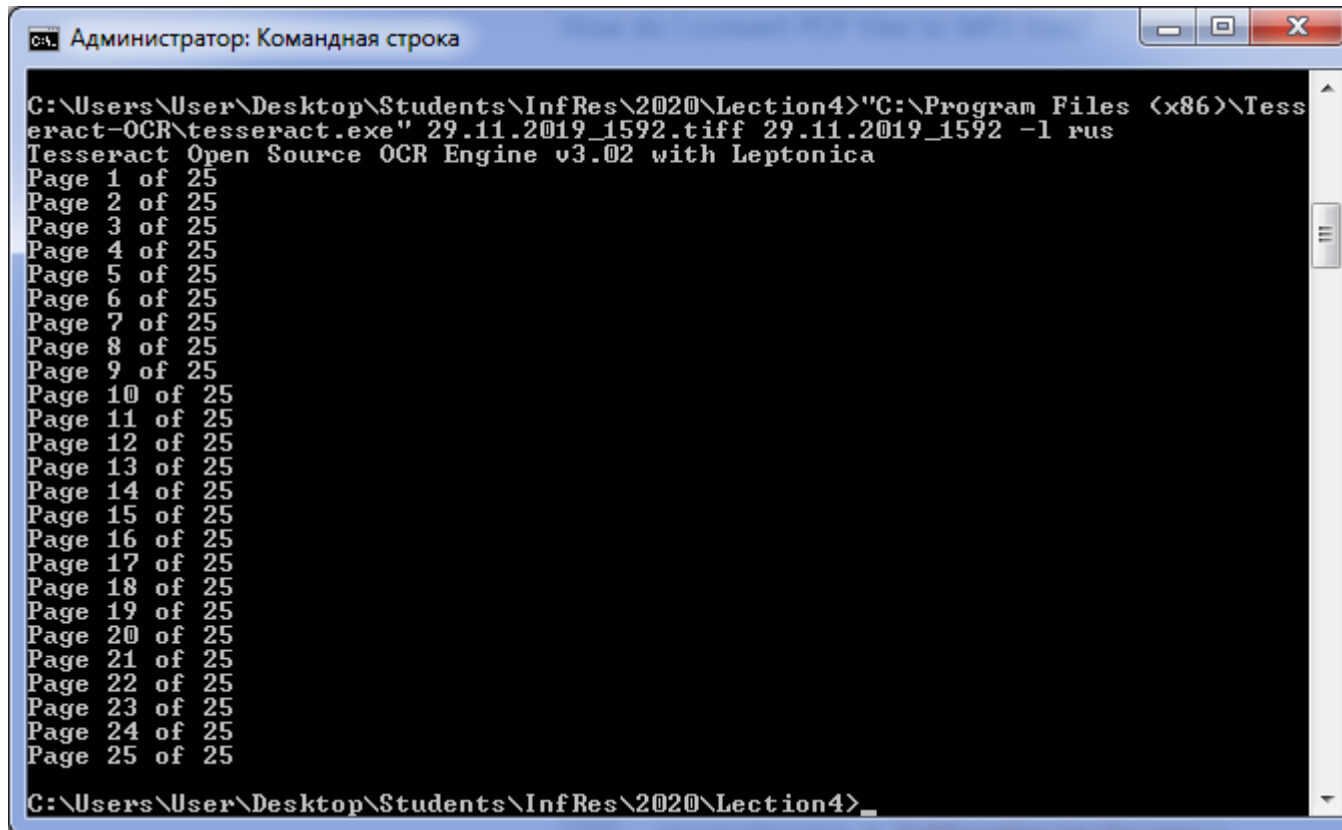
2. Департаменту регулирования бухгалтерского учета, финансовой
отчетности и аудиторской деятельности не позднее 31 марта 2020 года
разработать с участием заинтересованных организаций и профессионального
сообщества и представить на утверждение План мероприятий («дорожную
карту») по реализации Основных направлений развития аудиторской
деятельности на период до 2024 года.

Первый заместитель Председателя
Правительства Российской Федерации —
Министр финансов

Российской Федерации А.Г. Силуанов
>>>
```

# Tesseract можно использовать и без Python

Tesseract отлично распознает файлы и сам по себе. Достаточно запустить его в терминале cmd указав три аргумента: tiff-изображение, txt-файл для результата и с ключом -l язык.

A screenshot of a Windows Command Prompt window titled "Администратор: Командная строка". The window shows the execution of the Tesseract OCR command. The command entered is: `C:\Users\User\Desktop\Students\InfRes\2020\Lecture4>"C:\Program Files (x86)\Tesseract-OCR\tesseract.exe" 29.11.2019_1592.tiff 29.11.2019_1592 -l rus`. The output shows the Tesseract version and language, followed by a list of 25 pages. The command prompt ends with the same directory path: `C:\Users\User\Desktop\Students\InfRes\2020\Lecture4>`.

```
Администратор: Командная строка
C:\Users\User\Desktop\Students\InfRes\2020\Lecture4>"C:\Program Files (x86)\Tesseract-OCR\tesseract.exe" 29.11.2019_1592.tiff 29.11.2019_1592 -l rus
Tesseract Open Source OCR Engine v3.02 with Leptonica
Page 1 of 25
Page 2 of 25
Page 3 of 25
Page 4 of 25
Page 5 of 25
Page 6 of 25
Page 7 of 25
Page 8 of 25
Page 9 of 25
Page 10 of 25
Page 11 of 25
Page 12 of 25
Page 13 of 25
Page 14 of 25
Page 15 of 25
Page 16 of 25
Page 17 of 25
Page 18 of 25
Page 19 of 25
Page 20 of 25
Page 21 of 25
Page 22 of 25
Page 23 of 25
Page 24 of 25
Page 25 of 25
C:\Users\User\Desktop\Students\InfRes\2020\Lecture4>
```



# Полезные ссылки

---

1. <https://www.minfin.ru/ru/performance/audit/basics/>
2. [https://minfin.gov.ru/common/upload/library/2018/11/main/Stuk\\_071118.pdf](https://minfin.gov.ru/common/upload/library/2018/11/main/Stuk_071118.pdf)
3. <https://pypi.org/project/urllib3/>
4. <https://pypi.org/project/python-docx/>
5. <https://tproger.ru/translations/regular-expression-python/>
6. <https://pypi.org/project/pdfminer.six/>
7. <https://tesseract-ocr.github.io/tessdoc/>
8. <https://www.youwindowsworld.com/en/downloads/software/tools/tesseract-ocr/download-tesseract-ocr>
9. <https://tesseract.patagames.com/langs/>
10. <https://pypi.org/project/pytesseract/>