

# Lendo documentos

Arquivos PDF

# Lendo documentos

## Arquivos PDF

Ao realizar um scraping você pode ter a necessidade de tratar arquivos PDF.

O **PDF (Portable Document Format)** é um formato de arquivo, desenvolvido pela Adobe Systems em 1993, para representar documentos de maneira independente do aplicativo, do *hardware* e do sistema operacional usados para criá-los. Um arquivo PDF pode descrever documentos que contenham texto, gráficos e imagens num formato independente de dispositivo e resolução.

# Lendo documentos

## Arquivos PDF

O PDF é um padrão aberto, e qualquer pessoa pode escrever aplicativos que leiam ou escrevam neste padrão. Há aplicativos gratuitos para Microsoft Windows, Mac e Linux, alguns deles distribuídos pela própria Adobe e há diversos aplicativos sob licenças livres.

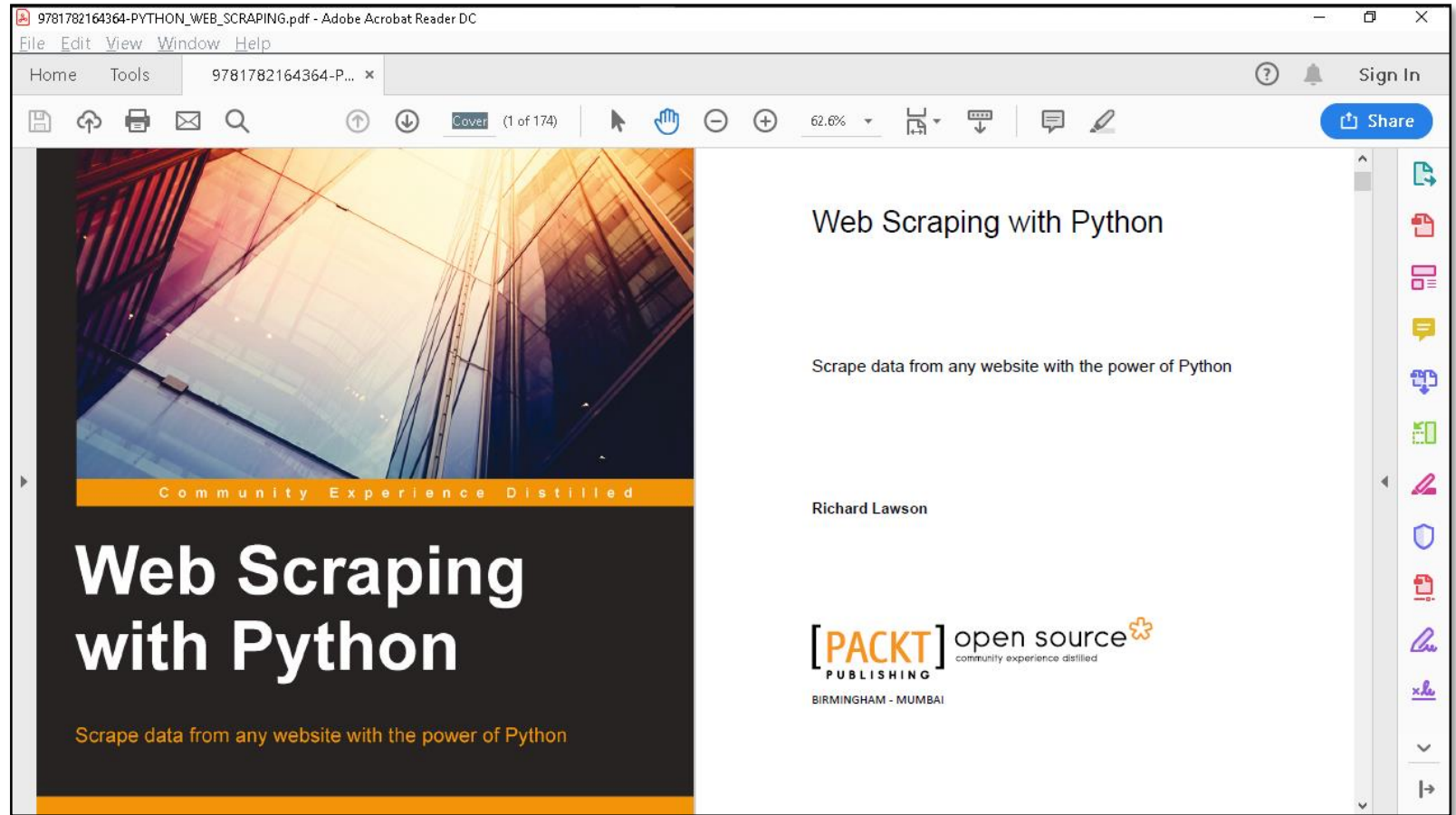
Fonte: [https://pt.wikipedia.org/wiki/Portable Document Format](https://pt.wikipedia.org/wiki/Portable_Document_Format)



# Lendo documentos

## Arquivos PDF

Um arquivo PDF aberto no Adobe Acrobat Reader, que é um programa gratuito para leitura de arquivos PDF. Nas minhas aulas tenho disponibilizado materiais em PDF.



# Lendo documentos

## Arquivos PDF

Para trabalhar com arquivos PDF podemos usar a biblioteca PDFMiner3k.

O PDFMiner3k é uma implementação do pdfminer para Python. O PDFMiner é uma ferramenta para extrair informações de documentos PDF.

Ao contrário de outras ferramentas relacionadas a PDF, ele se concentra inteiramente em obter e analisar dados de texto. O PDFMiner permite obter a localização exata dos textos em uma página, bem como outras informações, como fontes ou linhas. Inclui um conversor de PDF que pode transformar arquivos PDF em outros formatos de texto (como HTML).

# Lendo documentos

## Arquivos PDF

Podemos instalar esta biblioteca utilizando o pip.

*\$ pip install pdfminer3k*

```
evaldowolkers@evaldo ~/aula/pasta7 $ pip install pdfminer3k
Collecting pdfminer3k
  Downloading https://files.pythonhosted.org/packages/8c/87/cee0aa24f95c287020df7e3936cb51d32b34b05b430759bac15f89ea5ac2/pdfminer3k-1.3.1.tar.gz (4.1MB)
    2% |          | 102kB 382kB/s eta 0:00:1
    2% |          | 112kB 510kB/s eta 0:00:0
    2% |          | 122kB 480kB/s eta 0:00:0
    3% |          | 133kB 469kB/s eta 0:00:0
    3% |          | 143kB 623kB/s eta 0:00:0
    3% |          | 153kB 623kB/s eta 0:00:0
    3% |          | 163kB 538kB/s eta 0:00:0
    4% |          | 174kB 619kB/s eta 0:00:0
    4% |          | 184kB 625kB/s eta 0:00:0
    4% |          | 194kB 626kB/s eta 0:00:0
```

```
C:\Users\evaldo>pip install pdfminer3k
Collecting pdfminer3k
  Downloading https://files.pythonhosted.org/packages/8c/87/cee0aa24f95c287020df7e3936cb51d32b34b05b430759bac15f89ea5ac2/pdfminer3k-1.3.1.tar.gz (4.1MB)
    100% |          | 4.1MB 610kB/s
Collecting pytest>=2.0 (from pdfminer3k)
  Downloading https://files.pythonhosted.org/packages/70/0b/c577e79496be9698ca118afe0c1dafd4878dec73337b21570b0d28bacc2/pytest-3.7.3-py2.py3-none-any.whl (204kB)
    100% |          | 204kB 614kB/s
Collecting ply>=3.4 (from pdfminer3k)
  Downloading https://files.pythonhosted.org/packages/a3/58/35da89ee790598a0700ea49b2a66594140f44dec458c07e8e3d4979137fc/ply-3.11-py2.py3-none-any.whl (49kB)
    100% |          | 51kB 544kB/s
Collecting py>=1.5.0 (from pytest>=2.0->pdfminer3k)
  Downloading https://files.pythonhosted.org/packages/c8/47/d179b80ab1dc1bfd46
```

# Lendo documentos

## Arquivos PDF

# Realizando as importações necessárias.

```
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
# Para abrir um PDF localmente
from io import open
# Para abrir um PDF online
from urllib.request import urlopen
```

# Lendo documentos

## Arquivos PDF

`PDFResourceManager`

Repositório de recursos compartilhados.

`ResourceManager` facilita a reutilização de recursos compartilhados tais como fontes e imagens, para que objetos não sejam alocados várias vezes ocupando muito espaço em memória.



# Lendo documentos

## Arquivos PDF

LAParams

Define os parâmetros que serão passados para a função TextConverter.

line\_overlap=0.5 (sobreposição de linha)

char\_margin=2.0 (margem do caracter)

line\_margin=0.5 (margem da linha)

word\_margin=0.1 (margem da palavra)

paragraph\_indent=None (indentação de parágrafo)

# Lendo documentos

## Arquivos PDF

TextConverter

Converte o conteúdo do PDF em texto.

HTMLConverter

Converte o conteúdo do PDF em HTML.

XMLConverter

Converte o conteúdo do PDF em XML.

# Lendo documentos

## Arquivos PDF

`Process_pdf`

`process_pdf` é uma função que executa as seguintes tarefas:

Cria um objeto analisador de PDF associado ao objeto de arquivo.

Cria um objeto de documento PDF que armazena a estrutura do documento.

Conecta o analisador e os objetos do documento.

Fornece a senha ao documento para inicialização (caso você tenha informado uma senha).

Verifica se o documento permite a extração de texto. Se não permite, aborta.

Cria um objeto de interpretação de PDF.

Processa cada página contida no documento.

# Lendo documentos

## Arquivos PDF

```
def lerPDF(arquivoPDF):  
    # PDFResourceManager Usado para armazenar recursos compartilhados  
    # como fontes e imagens  
    recursos = PDFResourceManager()  
    buffer = StringIO()  
    layoutParams = LAParams()  
    dispositivo = TextConverter(recursos, buffer, laparams=layoutParams)  
  
    process_pdf(recursos, dispositivo, arquivoPDF)  
    dispositivo.close()  
  
    conteudo = buffer.getvalue()  
    buffer.close()  
    return conteudo
```

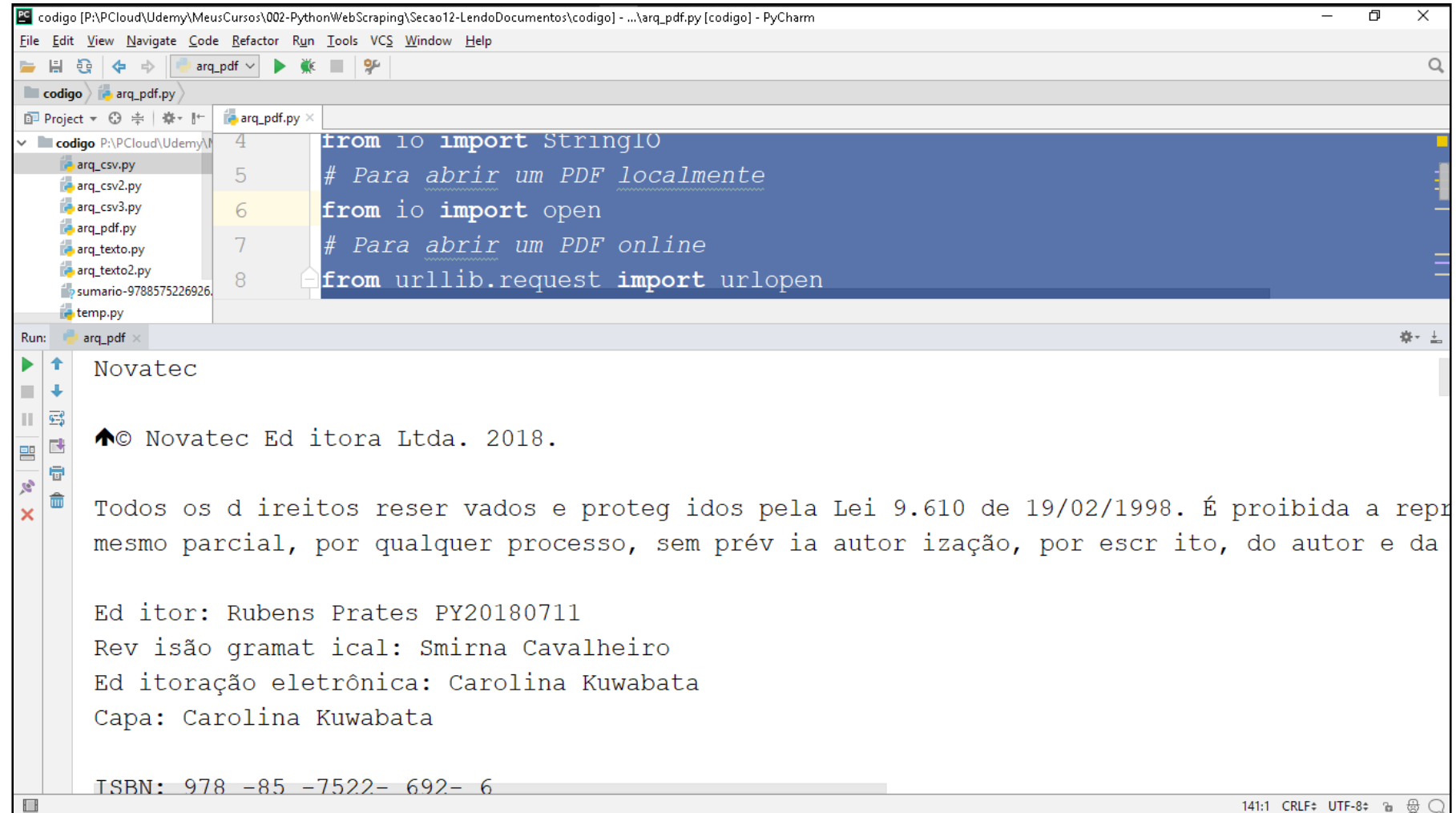
# Lendo documentos

## Arquivos PDF

```
# Arquivo PDF online
#arquivoPDF = urlopen("https://s3.novatec.com.br/sumarios/sumario-9788575226926.pdf")
# Arquivo PDF local (Abrindo modo leitura e binário)
arquivoPDF = open("sumario-9788575226926.pdf", "rb")
stringSaida = lerPDF(arquivoPDF)
print(stringSaida)
arquivoPDF.close()
```

# Lendo documentos

## Arquivos PDF



The screenshot shows the PyCharm IDE with a project named 'codigo'. The file explorer on the left lists several Python files: 'arq\_csv.py', 'arq\_csv2.py', 'arq\_csv3.py', 'arq\_pdf.py', 'arq\_texto.py', 'arq\_texto2.py', 'sumario-9788575226926.py', and 'temp.py'. The 'arq\_pdf.py' file is open in the editor, showing the following code:

```
4 from io import StringIO
5 # Para abrir um PDF localmente
6 from io import open
7 # Para abrir um PDF online
8 from urllib.request import urlopen
```

The Run console at the bottom displays the output of the script, which is the content of a PDF document:

```
Novatec

© Novatec Ed itora Ltda. 2018.

Todos os d ireitos reser vados e proteg idos pela Lei 9.610 de 19/02/1998. É proibida a repr
mesmo parcial, por qualquer processo, sem prév ia autor ização, por escr ito, do autor e da

Ed itor: Rubens Prates PY20180711
Rev isão gramat ical: Smirna Cavalheiro
Ed itoração eletrônica: Carolina Kuwabata
Capa: Carolina Kuwabata

ISBN: 978 -85 -7522- 692- 6
```

# Lendo documentos

## Arquivos PDF

### Convertendo em HTML

```
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import HTMLConverter
from pdfminer.layout import LAParams
from io import StringIO
from urllib.request import urlopen

def lerPDF(arquivo):
    recursos = PDFResourceManager()
    buffer = StringIO()
    layoutParams = LAParams()
    disp = HTMLConverter(recursos, buffer, laparams=layoutParams)

    process_pdf(recursos, disp, arquivo)
    disp.close()

    conteudo = buffer.getvalue()
    buffer.close()
    return conteudo

arquivoPDF = urlopen("https://s3.novatec.com.br/sumarios/sumario-9788575226926.pdf")
saida = lerPDF(arquivoPDF)
print(saida)
arquivoPDF.close()
```

# Lendo documentos

## Arquivos PDF

### Convertendo em XML

```
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import XMLConverter
from pdfminer.layout import LAParams
from io import StringIO
from urllib.request import urlopen

def lerPDF(arquivo):
    recursos = PDFResourceManager()
    buffer = StringIO()
    layoutParams = LAParams()
    disp = XMLConverter(recursos, buffer, laparams=layoutParams)

    process_pdf(recursos, disp, arquivo)
    disp.close()

    conteudo = buffer.getvalue()
    buffer.close()
    return conteudo

arquivoPDF = urlopen("https://s3.novatec.com.br/sumarios/sumario-9788575226926.pdf")
saida = lerPDF(arquivoPDF)
print(saida)
arquivoPDF.close()
```



# Lendo documentos

## Arquivos PDF

Obs.: Para exportar para XML e HTML tive que alterar o arquivo

“Python36\Lib\site-packages\pdfminer\utils.py”

da biblioteca pdfminer e estou disponibilizando em anexo à aula.

A função htmlescape tem que ser corrigida.

```
def htmlescape(s, encoding=None):  
    """Escapes a string for SGML/XML/HTML"""  
    s = s.replace('&', '&amp;').replace('>', '&gt;').replace('<', '&lt;').replace('"', '&quot;')  
    # Additionally to basic replaces, we also make sure that all characters are convertible to our  
    # target encoding. If they're not, they're replaced by XML entities.  
    if not encoding:  
        encoding = 'ascii'  
    encoded = s.encode(encoding, errors='xmlcharrefreplace')  
    return encoded.decode(encoding)
```

# FIM