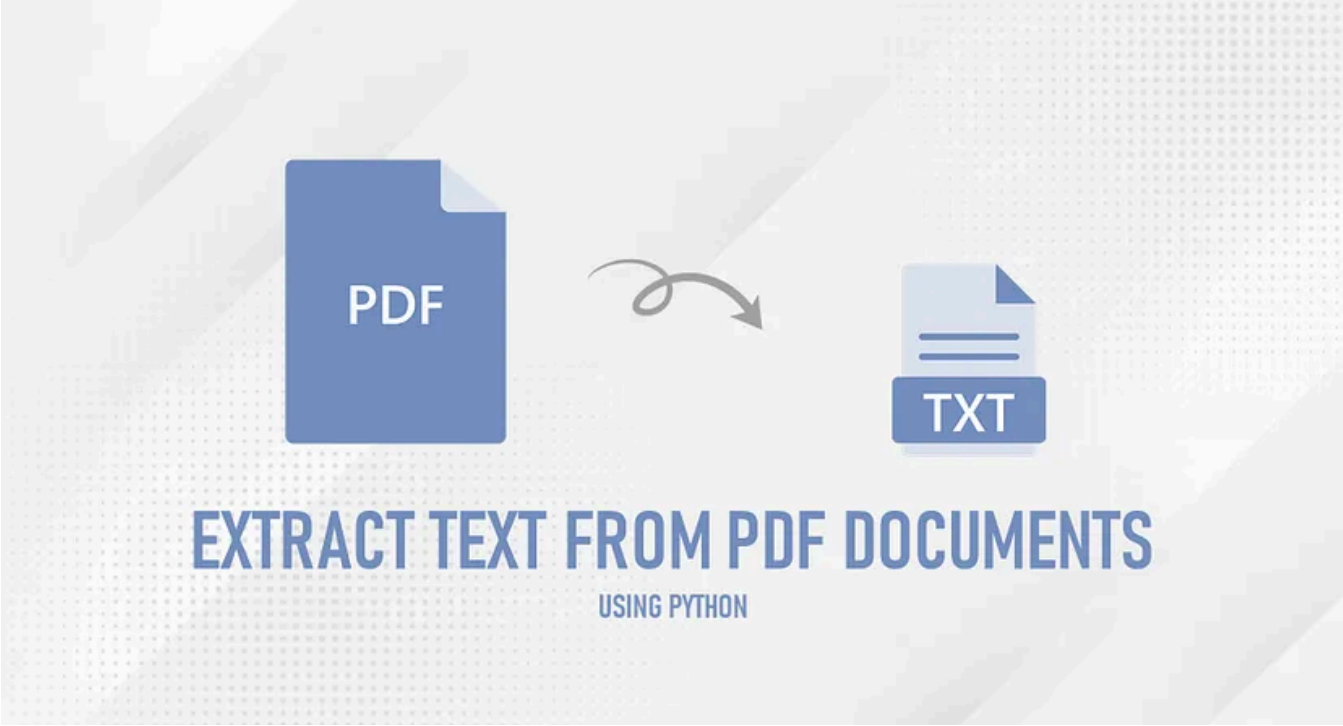
Extract Text from PDF with Python: Developer Guide

Alexander Stock · Follow
Published in Python in Plain English
6 min read · Feb 27, 2024



The process of extracting text from PDF documents is essential for a wide range of applications and industries. Extracting text involves retrieving the textual information contained within PDF files, enabling analysis, search, and manipulation. With the help of dedicated libraries and tools, developers can programmatically extract text from PDFs with ease and efficiency.

In this article, I will guide you on extracting text from PDF documents using the **Spire.PDF for Python** library. You will learn how to extract text from the entire PDF

document, as well as from specific pages or even from a defined rectangular area within a page.

- Extract Text from a Particular Page
- Extract Text from a Rectangular Area
- Extract Text from an Entire PDF Document
- Extract Tables from a Page in a PDF Document

Python Library for PDF Text Extraction

Spire.PDF for Python is a professional library that provides comprehensive PDF manipulation capabilities for Python developers. With Spire.PDF, developers can easily create, modify, and extract content from PDF documents using simple and intuitive APIs.

One of the notable features of Spire.PDF is its support for extracting text from searchable PDF documents. Searchable PDFs are PDF files that contain embedded text information, making it easier to search for and extract text from them programmatically. This feature is particularly useful when you need to extract specific information or perform text analysis from PDF documents.

The library can be installed from Pypi using the following pip command.

```
pip install Spire.PDF
```

Extract Text from a Particular Page with Python

Spire.PDF provides the **PdfTextExtractor** class to extract text from specific pages of a PDF document, along with the **PdfTextExtractionOptions** for customizing the extraction process. The options offered include extracting all text from a page, extracting text from a specified rectangular area, and more.

To extract text from a particular PDF page, follow these steps:

1. Import the necessary modules.
2. Create a **PdfDocument** object.
3. Load a PDF document from a given file path.

4. Get a specific page from the document.
5. Create a **PdfTextExtractor** object for the selected page.
6. Create a **PdfTextExtractOptions** object to define the extraction options.
7. Set the **IsExtractAllText** property of the extractOptions object to True to extract all text from the page.
8. Extract the text from the page using the **ExtractText()** method of the textExtractor object, passing the extractOptions object as a parameter.
9. Write the extracted text to a text file named “TextOfPageSix.txt” in the “output” directory.

```
from spire.pdf.common import *
from spire.pdf import *

# Create a PdfDocument object
doc = PdfDocument()

# Load a PDF document
doc.LoadFromFile("C:\\Users\\Administrator\\Desktop\\privacy policy.pdf")

# Get a specific page
page = doc.Pages[5]

# Create a PdfTextExtractor object
textExtractor = PdfTextExtractor(page)

# Create a PdfTextExtractOptions object
extractOptions = PdfTextExtractOptions()

# Set IsExtractAllText to Ture
extractOptions.IsExtractAllText = True

# Extract text from the page keeping white spaces
text = textExtractor.ExtractText(extractOptions)

# Write text to a txt file
with open('output/TextOfPageSix.txt', 'w') as file:
    lines = text.split("\n")
    for line in lines:
        if line != '':
            file.write(line)
doc.Close()
```

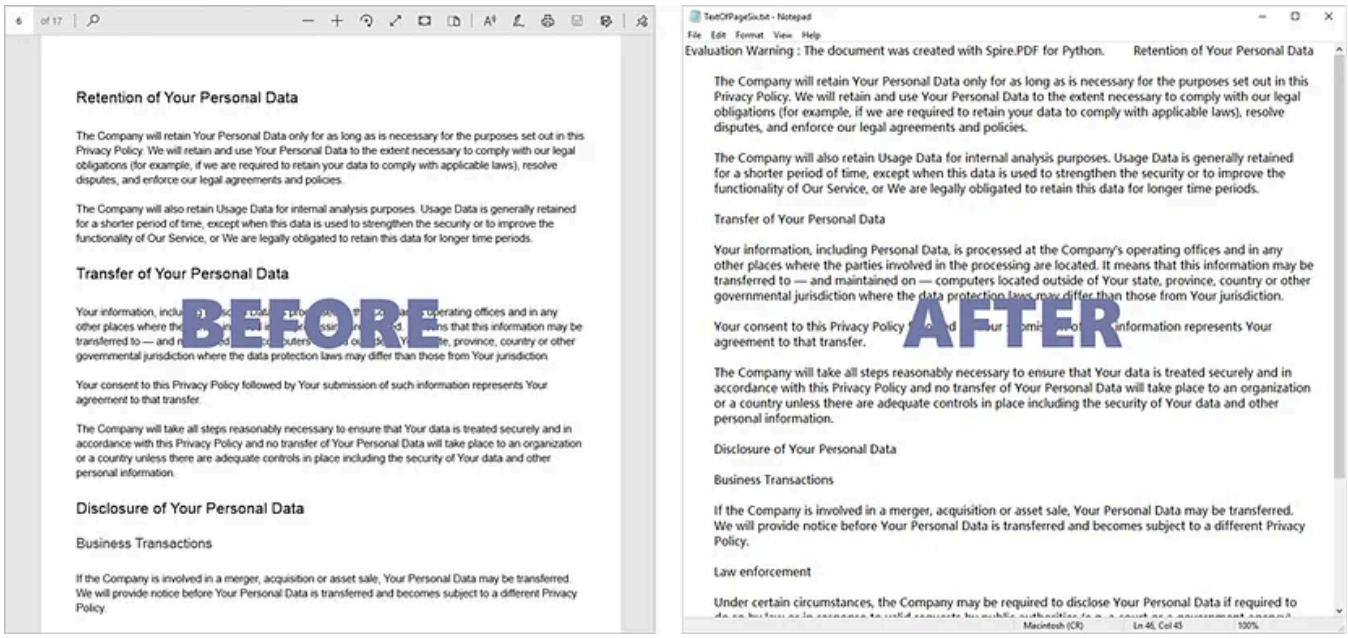


Figure 1. Extract Text from a Specified PDF Page

Note: The library has a limitation where it can only extract text from the first 10 pages of a PDF document. Additionally, the generated text files contain a line stating "Evaluation Warning: The document was created with Spire.PDF for Python." To remove these restrictions, you have the option to obtain a 30-day trial license from the vendor.

Extract Text from a Rectangular Area with Python

To define a rectangular area on a page for text extraction, you can utilize the **ExtractArea** property of a **PdfTextExtractOptions** object. Then, pass the PdfTextExtractOptions object as a parameter when calling the **PdfTextExtractor.ExtractText()** method.

The following steps outline the process of extracting text from a rectangular area of a PDF page.

1. Import the necessary modules.
2. Create a **PdfDocument** object.
3. Load a PDF document.
4. Get a specific page.
5. Create a **PdfTextExtractor** object for the selected page.
6. Create a **PdfTextExtractOptions** object to define the extraction options.

7. Specify the dimensions of the rectangular are through the **ExtractArea** property of the extractOptions object.

8. Extracts text from the specified rectangular area using the **ExtractText()** method and the provided extractOptions .

9. Write the extracted text to a text file named “TextOfRectangleArea.txt”.

```
from spire.pdf.common import *
from spire.pdf import *

# Create a PdfDocument object
doc = PdfDocument()

# Load a PDF document
doc.LoadFromFile("C:\\Users\\Administrator\\Desktop\\privacy policy.pdf")

# Get a specific page
page = doc.Pages[5]

# Create a PdfTextExtractor object
textExtractor = PdfTextExtractor(page)

# Create a PdfTextExtractOptions object
extractOptions = PdfTextExtractOptions()

# Specify the dimensions of the rectangular area
extractOptions.ExtractArea = RectangleF(0.0, 210.0, page.Size.Width, 210.0 + 60

# Extract text from the page keeping white spaces
text = textExtractor.ExtractText(extractOptions)

# Write text to a txt file
with open('output/TextOfRectangleArea.txt', 'w') as file:
    lines = text.split("\n")
    for line in lines:
        if line != '':
            file.write(line)
doc.Close()
```

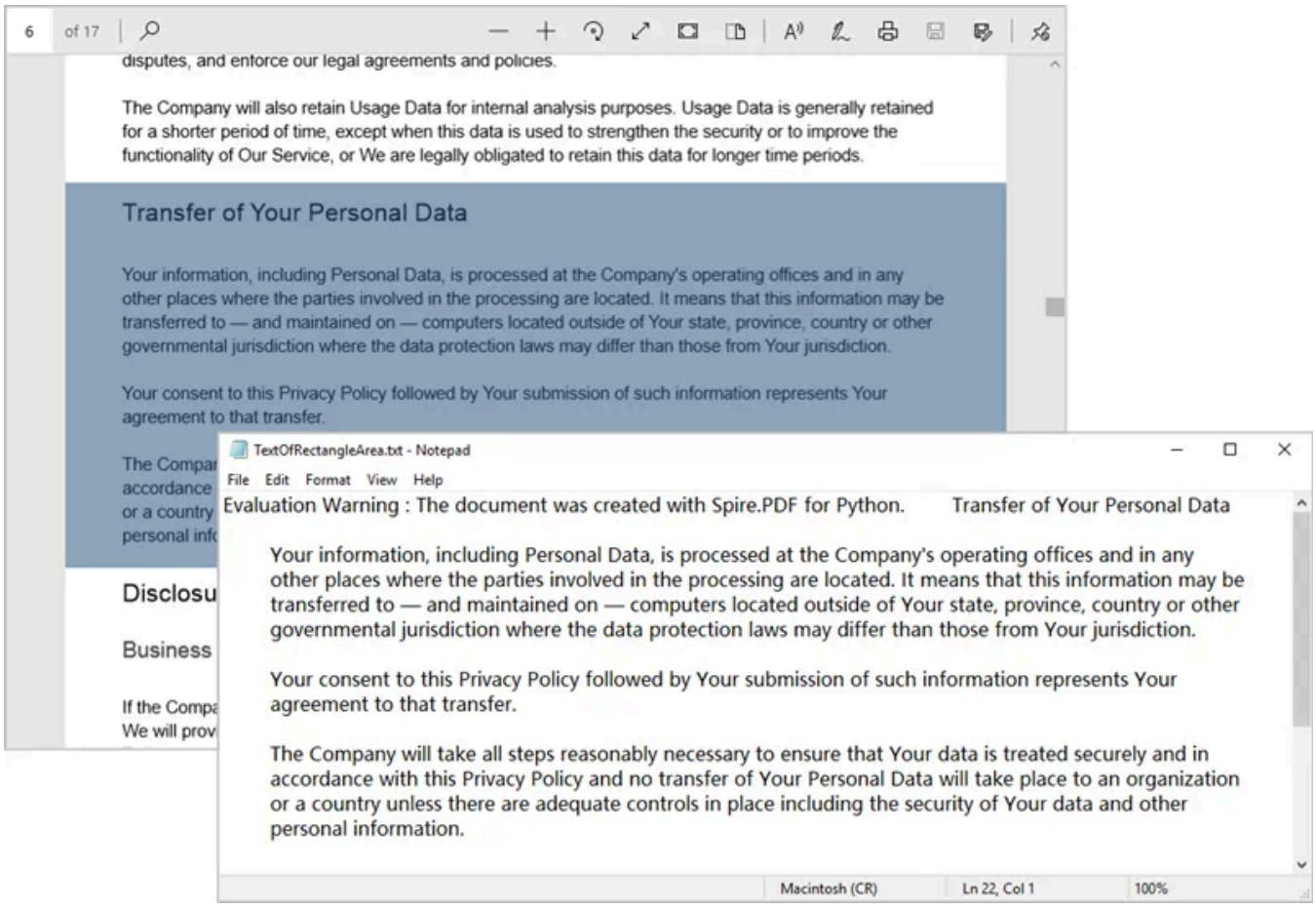


Figure 2. Extract Text from a Rectangular Area

Extract Text from an Entire PDF Document

The initial part of the code illustrates the process of extracting text from a single page. However, to extract text from an entire PDF document, we can extend the functionality by iterating through the pages in the document and sequentially retrieving the text from each page.

```
from spire.pdf.common import *
from spire.pdf import *

# Create a PdfDocument object
doc = PdfDocument()

# Load a PDF document
doc.LoadFromFile("C:/Users/Administrator/Desktop/privacy policy.pdf")

# Iterate through the pages in the document
for i in range(doc.Pages.Count):

    # Get a specific page
    page = doc.Pages[i]

    # Create a PdfTextExtractor object
    textExtractor = PdfTextExtractor(page)
```

```
    # Create a PdfTextExtractOptions object
    extractOptions = PdfTextExtractOptions()

    # Set IsExtractAllText to Ture
    extractOptions.IsExtractAllText = True

    # Extract text from the page keeping white spaces
    text = textExtractor.ExtractText(extractOptions)

    # Write text to a txt file
    with open('output/TextOfPage-{}.txt'.format(i + 1), 'w') as file:
        lines = text.split("\n")
        for line in lines:
            if line != '':
                file.write(line)
doc.Close()
```

Extract Tables from a Page in a PDF Document

Spire.PDF for Python offers the `PdfTableExtractor.ExtractTable(pageIndex)` function to extract tables from a searchable PDF document. Once the tables have been extracted, you can iterate through the rows and columns of each table and then access the text content of individual table cells using the `PdfTable.GetText(rowIndex, columnIndex)` method.

```
from spire.pdf.common import *
from spire.pdf import *

# Create a PdfDocument object
doc = PdfDocument()

# Load a PDF file
doc.LoadFromFile("C:\\Users\\Administrator\\Desktop\\Tables.pdf")

# Create a list to store the extracted data
builder = []

# Create a PdfTableExtractor object
extractor = PdfTableExtractor(doc)

# Extract tables from a specific page (page index starts from 0)
tableList = extractor.ExtractTable(0)

print(len(tableList))

# Determine if the table list is not empty
if tableList is not None and len(tableList) > 0:
```

```
    # Loop through the tables in the list
    for table in tableList:

        # Get row number and column number of a certain table
        row = table.GetRowCount()
        column = table.GetColumnCount()

        # Loop through the row and column
        for i in range(row):
            for j in range(column):

                # Get text from the specific cell
                text = table.GetText(i, j)

                # Add the text to the list
                builder.append(text + " ")
            builder.append("\n")
        builder.append("\n")

# Write the content of the list into a text file
with open("output/Tables.txt", "w", encoding="utf-8") as file:
    file.write("".join(builder))
```

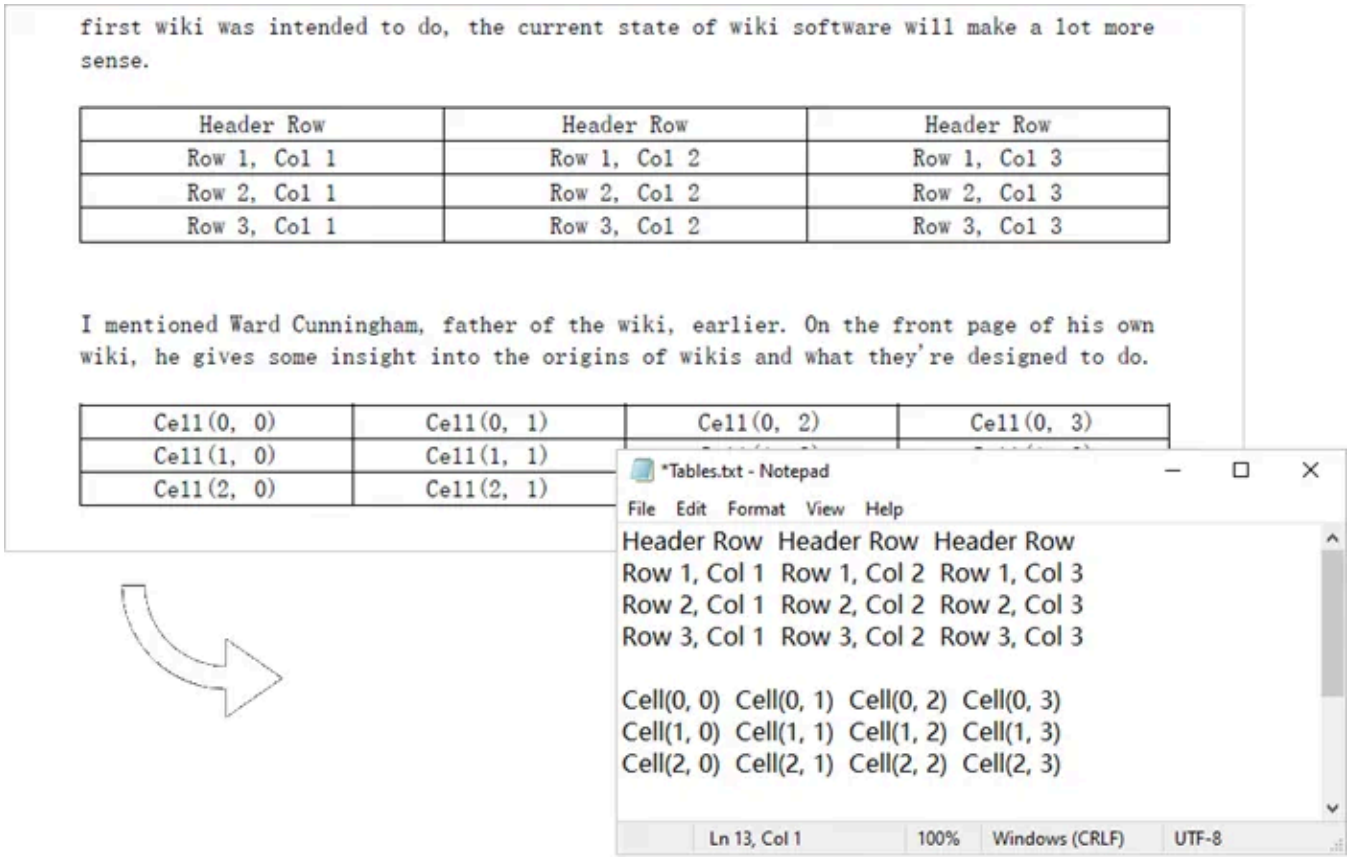


Figure 3. Extract tables from a PDF page.

Conclusion