PDF PARSING

```
from google.colab import files
files.download("sample.pdf")
```

```
import PyPDF2

with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    for page in reader.pages:
        print(page.extract_text())
```

```
Python Programming Basics
Python is a high-level programming language.
It is easy to learn and simple to use.
Python is used in:
- Web Development
- Data Science
- Artificial Intelligence
- Automation
Python is one of the most popular languages in the world.
```

```
import PyPDF2

with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    first_page = reader.pages[0]
    text = first_page.extract_text()

print(text)
```

```
Python Programming Basics
Python is a high-level programming language.
It is easy to learn and simple to use.
Python is used in:
- Web Development
- Data Science
- Artificial Intelligence
- Automation
Python is one of the most popular languages in the world.
```

```
import PyPDF2

with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    total_pages = len(reader.pages)

print("Total number of pages:", total_pages)
```

```
Total number of pages: 1
```

```
import PyPDF2

with open("sample.pdf","rb") as file:
    reader = PyPDF2.PdfReader(file)

    is_scanned = True

    for page in reader.pages:
        text = page.extract_text()
        if text and text.strip():
            is_scanned = False
            break

if is_scanned:
    print("This PDF is likely SCANNED (no extractable text).")
```

```
else:
    print("This PDF is TEXT-BASED (text can be extracted).")
```

```
This PDF is TEXT-BASED (text can be extracted).
```

```
import PyPDF2

headings = []

with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)

    for page in reader.pages:
        text = page.extract_text()
        if text:
            lines = text.split("\n")
            for line in lines:
                line = line.strip()
                if line and len(line) < 40:
                    headings.append(line)

print("Extracted Headings:")
for h in headings:
    print(h)
```

```
Extracted Headings:
Python Programming Basics
It is easy to learn and simple to use.
Python is used in:
- Web Development
- Data Science
- Artificial Intelligence
- Automation
```

```
!pip install pdfplumber
```

```
Collecting pdfplumber
  Downloading pdfplumber-0.11.8-py3-none-any.whl.metadata (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.6/43.6 kB 1.8 MB/s eta 0:00:00
Collecting pdfminer.six==20251107 (from pdfplumber)
  Downloading pdfminer_six-20251107-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: Pillow>=9.1 in /usr/local/lib/python3.12/dist-packages (from pdfplumber) (11.3.0)
Collecting pypdfium2>=4.18.0 (from pdfplumber)
  Downloading pypdfium2-5.2.0-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.8/67.8 kB 4.3 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from pdfminer.six==20251107
Requirement already satisfied: cryptography>=36.0.0 in /usr/local/lib/python3.12/dist-packages (from pdfminer.six==20251107->pdf
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.12/dist-packages (from cryptography>=36.0.0->pdfminer.six==2
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.12->cryptography>=36.0.0->pdfm
Downloading pdfplumber-0.11.8-py3-none-any.whl (60 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.0/60.0 kB 4.4 MB/s eta 0:00:00
Downloading pdfminer_six-20251107-py3-none-any.whl (5.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 58.9 MB/s eta 0:00:00
Downloading pypdfium2-5.2.0-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 93.3 MB/s eta 0:00:00
Installing collected packages: pypdfium2, pdfminer.six, pdfplumber
Successfully installed pdfminer.six-20251107 pdfplumber-0.11.8 pypdfium2-5.2.0
```

```
import pdfplumber

with pdfplumber.open("table_sample.pdf") as pdf:
    for i, page in enumerate(pdf.pages):
        tables = page.extract_tables()
        if tables:
            print(f"Tables on Page {i+1}:")
            for table in tables:
                for row in table:
                    print(row)
                print("-" * 30)
```

```
Tables on Page 1:
['Item', 'Price', 'Quantity']
['Pen', '10', '5']
['Book', '50', '2']
```

```
['Bag', '700', '1']
------------------------------
```

```
import PyPDF2

with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    metadata = reader.metadata
print("PDF Metadata:")
print("Title:", metadata.title)
print("Author:", metadata.author)
print("Creation Date:", metadata.creation_date)
```

```
PDF Metadata:
Title: untitled
Author: anonymous
Creation Date: 2025-12-13 08:08:05+00:00
```

```
import PyPDF2
with open("sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    full_text = ""
    for page in reader.pages:
        text = page.extract_text()
        if text:
            full_text += text + "\n"
with open("extracted_text.txt", "w", encoding="utf-8") as f:
    f.write(full_text)

print("Text extracted and saved to extracted_text.txt")
```

```
Text extracted and saved to extracted_text.txt
```

```
import PyPDF2
import re
with open("contacts_sample.pdf", "rb") as file:
    reader = PyPDF2.PdfReader(file)
    full_text = ""
    for page in reader.pages:
        text = page.extract_text()
        if text:
            full_text += text + "\n"
email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
emails = re.findall(email_pattern, full_text)
print("Email IDs found in PDF:")
for email in emails:
    print(email)
```

```
Email IDs found in PDF:
john.doe@example.com
mia.smith@mycompany.org
helpdesk@service.com
```

PDF PLUMBER

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    first_page = pdf.pages[0]
    text = first_page.extract_text()

print("Text from first page:")
print(text)
```

```
Text from first page:
Contact List
John Doe
Email: john.doe@example.com
Phone: +1-555-123-4567
Mia Smith
```

```
Email: mia.smith@mycompany.org
Phone: +44-20-1234-5678
```

```
import pdfplumber

file_path = "sample.pdf"
all_text = ""

with pdfplumber.open(file_path) as pdf:
    for i, page in enumerate(pdf.pages):
        page_text = page.extract_text()
        if page_text:
            all_text += f"--- Page {i+1} ---\n"
            all_text += page_text + "\n"

print("Text from all pages:")
print(all_text)
```

```
Text from all pages:
--- Page 1 ---
Contact List
John Doe
Email: john.doe@example.com
Phone: +1-555-123-4567
Mia Smith
Email: mia.smith@mycompany.org
Phone: +44-20-1234-5678
--- Page 2 ---
Support Team
Support Team
Email: helpdesk@service.com
Phone: +91-98765-43210
Technical Team
Email: tech@company.org
Phone: +1-555-987-6543
```

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    total_pages = len(pdf.pages)

print("Total number of pages in the PDF:", total_pages)
```

```
Total number of pages in the PDF: 2
```

```
import pdfplumber

file_path = "sample.pdf"

is_scanned = True

with pdfplumber.open(file_path) as pdf:
    for page in pdf.pages:
        text = page.extract_text()
        if text and text.strip():
            is_scanned = False
            break

if is_scanned:
    print("This PDF is likely SCANNED (no extractable text).")
else:
    print("This PDF is TEXT-BASED (text can be extracted).")
```

```
This PDF is TEXT-BASED (text can be extracted).
```

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    first_page = pdf.pages[0]
```

```
    tables = first_page.extract_tables()
if tables:
    first_table = tables[0]
    print("First table on Page 1:")
    for row in first_table:
        print(row)
else:
    print("No tables found on Page 1.")
```

```
No tables found on Page 1.
```

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    for i, page in enumerate(pdf.pages):
        tables = page.extract_tables()
        if tables:
            print(f"Tables on Page {i+1}:")
            for t, table in enumerate(tables, start=1):
                print(f"Table {t}:")
                for row in table:
                    print(row)
                print("-" * 30)
        else:
            print(f"No tables found on Page {i+1}.")
```

```
No tables found on Page 1.
No tables found on Page 2.
```

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    for i, page in enumerate(pdf.pages):
        words = page.extract_words()
        print(f"Words on Page {i+1}:")
        for word in words:
            print(word)
        print("-" * 50)
```

```
Words on Page 1:
{'text': 'Contact', 'x0': 40.0, 'x1': 81.352, 'top': 32.48400000000004, 'doctop': 32.48400000000004, 'bottom': 44.48400000000004
{'text': 'List', 'x0': 84.688, 'x1': 103.36, 'top': 32.48400000000004, 'doctop': 32.48400000000004, 'bottom': 44.48400000000004,
{'text': 'John', 'x0': 40.0, 'x1': 66.016, 'top': 61.28399999999999, 'doctop': 61.28399999999999, 'bottom': 73.28399999999999, '
{'text': 'Doe', 'x0': 69.352, 'x1': 91.36, 'top': 61.28399999999999, 'doctop': 61.28399999999999, 'bottom': 73.28399999999999, '
{'text': 'Email:', 'x0': 40.0, 'x1': 73.336, 'top': 75.68399999999997, 'doctop': 75.68399999999997, 'bottom': 87.68399999999997,
{'text': 'john.doe@example.com', 'x0': 76.672, 'x1': 206.236, 'top': 75.68399999999997, 'doctop': 75.68399999999997, 'bottom': 8
{'text': 'Phone:', 'x0': 40.0, 'x1': 78.028, 'top': 90.08399999999995, 'doctop': 90.08399999999995, 'bottom': 102.08399999999995
{'text': '+1-555-123-4567', 'x0': 81.364, 'x1': 173.75199999999998, 'top': 90.08399999999995, 'doctop': 90.08399999999995, 'bott
{'text': 'Mia', 'x0': 40.0, 'x1': 59.331999999999994, 'top': 118.8839999999999, 'doctop': 118.8839999999999, 'bottom': 130.88399
{'text': 'Smith', 'x0': 62.668, 'x1': 93.34, 'top': 118.8839999999999, 'doctop': 118.8839999999999, 'bottom': 130.88399999999999,
{'text': 'Email:', 'x0': 40.0, 'x1': 73.336, 'top': 133.28399999999988, 'doctop': 133.28399999999988, 'bottom': 145.28399999999
{'text': 'mia.smith@mycompany.org', 'x0': 76.672, 'x1': 225.544, 'top': 133.28399999999988, 'doctop': 133.28399999999988, 'botto
{'text': 'Phone:', 'x0': 40.0, 'x1': 78.028, 'top': 147.68399999999986, 'doctop': 147.68399999999986, 'bottom': 159.68399999999
{'text': '+44-20-1234-5678', 'x0': 81.364, 'x1': 180.42399999999995, 'top': 147.68399999999986, 'doctop': 147.68399999999986, 'b
--------------------------------------------------
Words on Page 2:
{'text': 'Support', 'x0': 40.0, 'x1': 82.02400000000002, 'top': 32.48400000000004, 'doctop': 824.484, 'bottom': 44.4840000000000
{'text': 'Team', 'x0': 85.36000000000001, 'x1': 116.032, 'top': 32.48400000000004, 'doctop': 824.484, 'bottom': 44.4840000000000
{'text': 'Support', 'x0': 40.0, 'x1': 82.02400000000002, 'top': 61.28399999999999, 'doctop': 853.284, 'bottom': 73.2839999999999
{'text': 'Team', 'x0': 85.36000000000001, 'x1': 116.032, 'top': 61.28399999999999, 'doctop': 853.284, 'bottom': 73.2839999999999
{'text': 'Email:', 'x0': 40.0, 'x1': 73.336, 'top': 75.68399999999997, 'doctop': 867.684, 'bottom': 87.68399999999997, 'upright'
{'text': 'helpdesk@service.com', 'x0': 76.672, 'x1': 200.88400000000001, 'top': 75.68399999999997, 'doctop': 867.684, 'bottom':
{'text': 'Phone:', 'x0': 40.0, 'x1': 78.028, 'top': 90.08399999999995, 'doctop': 882.084, 'bottom': 102.08399999999995, 'upright
{'text': '+91-98765-43210', 'x0': 81.364, 'x1': 176.42799999999997, 'top': 90.08399999999995, 'doctop': 882.084, 'bottom': 102.0
{'text': 'Technical', 'x0': 40.0, 'x1': 91.348, 'top': 118.8839999999999, 'doctop': 910.8839999999999, 'bottom': 130.8839999999
{'text': 'Team', 'x0': 94.684, 'x1': 125.356, 'top': 118.8839999999999, 'doctop': 910.8839999999999, 'bottom': 130.8839999999999
{'text': 'Email:', 'x0': 40.0, 'x1': 73.336, 'top': 133.28399999999988, 'doctop': 925.2839999999999, 'bottom': 145.28399999999998
{'text': 'tech@company.org', 'x0': 76.672, 'x1': 180.892, 'top': 133.28399999999988, 'doctop': 925.2839999999999, 'bottom': 145.
{'text': 'Phone:', 'x0': 40.0, 'x1': 78.028, 'top': 147.68399999999986, 'doctop': 939.6839999999999, 'bottom': 159.68399999999998
{'text': '+1-555-987-6543', 'x0': 81.364, 'x1': 173.75199999999998, 'top': 147.68399999999986, 'doctop': 939.6839999999999, 'bot
--------------------------------------------------
```

```
import pdfplumber
from PIL import Image

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    page = pdf.pages[0]
    images = page.images

    if images:
        for i, img in enumerate(images, start=1):
            x0, y0, x1, y1 = img['x0'], img['y0'], img['x1'], img['y1']
            im = page.within_bbox((x0, y0, x1, y1)).to_image(resolution=300)
            im_path = f"image_page1_{i}.png"
            im.save(im_path)
            print(f"Saved image: {im_path}")
    else:
        print("No images found on this page.")
```

```
No images found on this page.
```

```
import pdfplumber

file_path = "sample.pdf"

with pdfplumber.open(file_path) as pdf:
    for i, page in enumerate(pdf.pages):
        print(f"--- Page {i+1} ---")
        lines = page.lines
        if lines:
            print("Lines:")
            for line in lines:
                print(line)
        else:
            print("No lines found.")
        rects = page.rects
        if rects:
            print("Rectangles / Shapes:")
            for rect in rects:
                print(rect)
        else:
            print("No rectangles/shapes found.")

        print("-" * 50)
```

```
--- Page 1 ---
No lines found.
No rectangles/shapes found.
--------------------------------------------------
--- Page 2 ---
No lines found.
No rectangles/shapes found.
--------------------------------------------------
```

```
import pdfplumber
import csv

file_path = "sample.pdf"
csv_file = "extracted_table.csv"

with pdfplumber.open(file_path) as pdf:
    first_page = pdf.pages[0]
    tables = first_page.extract_tables()

    if tables:
        first_table = tables[0]
        with open(csv_file, "w", newline="", encoding="utf-8") as f:
            writer = csv.writer(f)
            for row in first_table:
                writer.writerow(row)
        print(f"Table saved to {csv_file}")
    else:
        print("No tables found on Page 1.")
```

```
No tables found on Page 1.
```