How to Work with a PDF in Python

How to Work with a PDF in Python

1. **Introduction**
2. pyPDF2 History, an Alternative and Installation
3. Metadata Extraction and Rotating Pages
4. Merging and Splitting PDFs
5. Adding Watermarks and Encrypting PDFs
6. Conclusion and Further Reading

What is a PDF?

- PDF stands for Portable Document Format
- Reliably exchange documents across operating systems
- Initially invented by Adobe
- Now an open standard document format maintained by the International Organisation for Standardisation (ISO)

How to Work with a PDF in Python

1. Introduction
2. **pyPDF2 History, an Alternative and Installation**
3. Metadata Extraction and Rotating Pages
4. Merging and Splitting PDFs
5. Adding Watermarks and Encrypting PDFs
6. Conclusion and Further Reading

History of pyPDF2

- pyPDF initially released in 2005
- Final official release in 2010
- A company called Phasit sponsored the release of pyPDF2, written to be backwards compatible with pyPDF
- Final official release was in 2016
- pyPDF3 was short-lived, and was quickly renamed to be pyPDF4

History of pyPDF2 (continued)

- All of these packages do much the same thing
- pyPDF2 onwards have support for Python 3
- pyPDF has a fork called pyPDF for Python 3
- No longer maintained
- pyPDF2 was recently abandoned
- pyPDF4 does not currently have full backwards compatibility with pyPDF2
- Most of the following examples will work with pyPDF4
- Try it yourself!

pdfrw: An Alternative

- Created by Patrick Maupin
- Capable of most manipulations that pyPDF2 can complete
- **NOT** capable of encryption
- Integrates well with the ReportLab package

pyPDF2 Installation

- Python Shell

pyPDF2 Installation (continued)

- Python Shell
- Thonny IDE - Package manager

How to Work with a PDF in Python

1. Introduction
2. pyPDF2 History, an Alternative and Installation
3. **Metadata Extraction and Rotating Pages**
4. Merging and Splitting PDFs
5. Adding Watermarks and Encrypting PDFs
6. Conclusion and Further Reading

Extracting Document Metadata

- Useful for certain automation types
- Capable of extracting:
  - Author
  - Creator
  - Producer
  - Subject
  - Title
  - Number of pages

Rotating Pages

- Landscape vs. Portrait
- Upside-down pages

How to Work with a PDF in Python

1. Introduction
2. pyPDF2 History, an Alternative and Installation
3. Metadata Extraction and Rotating Pages
4. **Merging and Splitting PDFs**
5. Adding Watermarks and Encrypting PDFs
6. Conclusion and Further Reading

Merging PDF Files

- Merge two or more PDFs into a single file
- Example: having a standardised cover page to go across many different documents

Splitting a PDF

- Splitting a single PDF into smaller PDFs
- Each split out PDF will have a unique title

How to Work with a PDF in Python

Adding a Watermark to a PDF

- Identifying image or pattern
- Used on both printed and digital documents
- Some are visible only in special lighting conditions
- Used to protect intellectual property
- Sometimes called ‘overlays’
- pyPDF2 can be used to add an overlay to a document
- Watermark must be contained within its own PDF

Encrypting a PDF

- User password and an owner password
- User password - document is read only
- Owner password - ‘admin privileges’ that allows setting document permissions
- Document permission settings may not actually be set using pyPDF2

Encrypting a PDF (continued)

- PDF encryption uses either RC4 (Rivest Cipher 4) or AES (Advanced Encryption Standard) according to pdflib.com
- PDF Encryption does **NOT** mean that the PDF is secure
- Tools exist that can remove passwords - keep this in mind
- Carnegie Mellon University paper

How to Work with a PDF in Python

1. Introduction
2. pyPDF2 History, an Alternative and Installation
3. Metadata Extraction and Rotating Pages
4. Merging and Splitting PDFs
5. Adding Watermarks and Encrypting PDFs
6. **Conclusion and Further Reading**

Summary

- pyPDF2 History, an Alternative and Installation
- Metadata Extraction
- Rotating Pages
- Merging PDFs
- Splitting PDFs
- Adding Watermarks
- Encrypting PDFs

Further Reading

- The PyPDF2 website
- The Github page for PyPDF4
- The Github page for pdfrw
- The ReportLab website
- The Github page for PDFMiner
- Camelot: PDF Table Extraction for Humans

WELL DONE!