

Manipulating PDFs with Python

Manipulating PDFs with Python

Big picture, working with entire PDFs

```
# Reading and Extracting Text
import PyPDF2

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

```
# Merge two PDFs into one
import PyPDF2

merger = PyPDF2.PdfMerger()
pdfs = ['file1.pdf', 'file2.pdf']

for pdf in pdfs:
    merger.append(pdf)

merger.write('merged.pdf')
merger.close()
```

```
# Rotating Pages
import PyPDF2

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

    # Rotate first page 90 degrees clockwise
    page = reader.pages[0].rotate(90)
    writer.add_page(page)
```

```
# Add remaining pages as is
for page in reader.pages[1:]:
    writer.add_page(page)

with open('rotated.pdf', 'wb') as output_file:
    writer.write(output_file)
```

Detailed picture: create PDFs from scratch

```
from reportlab.pdfgen import canvas

c = canvas.Canvas("new.pdf")
c.drawString(100, 750, "Hello, PDF!")
c.save()
```

- You can place text
- You can draw
- Coordinates start at 0,0 in the lower-left

```
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter

c = canvas.Canvas("basic.pdf", pagesize=letter)
width, height = letter

# Draw text at coordinates (x, y)
c.drawString(100, 750, "Hello, ReportLab!")

# Draw a line
c.line(50, 700, 500, 700)

# Draw a rectangle (x, y, width, height) with fill
c.rect(100, 600, 200, 50, fill=1)

# Draw a circle (center x, center y, radius)
c.circle(300, 650, 40)
```

```
c.save()
```

Higher-level: Creating documents with Platypus

Platypus: Part of reportlab for higher level document constructs

flowables: paragraph, spacer, table, etc.

```
from reportlab.lib.pagesizes import letter
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Table,
TableStyle
from reportlab.lib import colors

# Create a document template

doc = SimpleDocTemplate("advanced.pdf", pagesize=letter)
styles = getSampleStyleSheet()
flowables = []

# Add a title and paragraph
flowables.append(Paragraph("Using ReportLab Platypus", styles['Title']))
flowables.append(Spacer(1, 12))
flowables.append(Paragraph("This is an example paragraph using ReportLab's
Platypus module.", styles['BodyText']))
flowables.append(Spacer(1, 12))

# Create a table with headers and data rows
data = [
    ['Header 1', 'Header 2', 'Header 3'],
    ['Row 1, Col 1', 'Row 1, Col 2', 'Row 1, Col 3'],
    ['Row 2, Col 1', 'Row 2, Col 2', 'Row 2, Col 3']
]
table = Table(data)
table.setStyle(TableStyle([
    ('BACKGROUND', (0, 0), (-1, 0), colors.grey),
    ('TEXTCOLOR', (0, 0), (-1, 0), colors.whitesmoke),
    ('ALIGN', (0, 0), (-1, -1), 'CENTER'),
    ('GRID', (0, 0), (-1, -1), 1, colors.black)
```

```
]])  
flowables.append(table)  
  
doc.build(flowables)
```

reportlab also has: Images, style sheets for fonts etc., templates that can be used for common headers, footers, layouts etc.