Lecture 30
PDF Report/ Output

OOP – Spring 2022 (Python)

PDF

PDF is an abbreviation that stands for **Portable Document Format**. It's a versatile file format created by Adobe that gives people an easy, reliable way to present and exchange documents - regardless of the software, hardware, or operating systems being used by anyone who views the document.

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiZ5ZjAy-L7AhVfTKQEHScPBbYQFnoECBgQAw&url=https%3A%2F%2Fwww.adobe.com%2Facrobat%2Fabout-adobe-pdf.html&usg=AOvVaw0TigmGhfJiUoq_zdTClzc-

Python FPDF

FPDF is a library for PDF document generation under Python. The main features are:

- Choice of measurement unit, page format and margins
- Page header and footer management
- Automatic page break
- Automatic line break and text justification
- Image, colors and links support

FPDF Methods

We will discuss following methods:

- add page
- set font
- cell
- output

Add Page

Adds a new page to the document. The signatures are:

```
fpdf.add_page(orientation = '', format = '', same = False)
```

All three parameters are optional. The details are:

- **Orientation** 'P' for portrait and 'L' for landscape
- **Format** can be 'A3', 'A4', 'A5', 'Letter' or 'Legal'
- **True** if page must be same as previous, in this case remaining parameters are ignored

Set Font

Sets the font used to print character strings. It is mandatory to call this method at least once before printing text or the resulting document would not be valid. The signatures are:

```
fpdf.set_font(family, style = '', size = 0)
```

First parameter is required, other two are optional:

- **Family** can be 'Courier', 'Helvetica', 'Arial', 'Times', 'Symbol', 'ZapfDingbats'
- **Style** can be empty (regular), 'B' (bold), 'I' (Italic), 'U' (Underline)
- **Font** size in points. The default value is the current size. If no size has been specified since the beginning of the document, the value taken is 12

Cell

Prints a cell (rectangular area) with optional borders, background color and character string. The signatures are:

```
fpdf.cell (w,h=0,txt='',border=0,ln=0,align='',fill=False,link='')
```

First parameter is required, other seven are optional: (ignore last parameter link)

- **W** Cell width
- **H** Cell height
- **txt** String to print
- **border** 0 (no border), 1 (frame) or it can be a string having 'L' (left), 'R' (right), 'T' (top), 'B' (bottom) in any order
- **ln** Indicates where the current position should go after the call. Possible values are: 0: to the right, 1: to the beginning of the first line, 2: below
- **align** 'L' (left), 'C' (Center), 'R' (Right)
- **fill** Indicate, whether the cell background should be painted or blank

Output

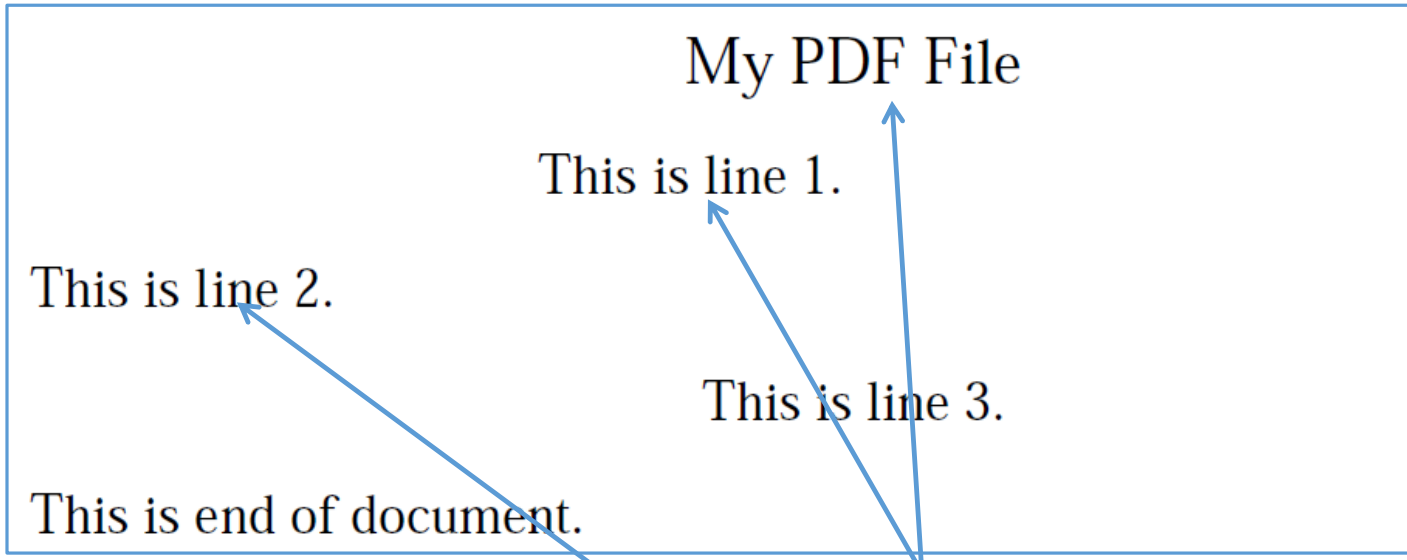
Send the document to some destination: standard output, a file or a byte string. The signatures are:

```
fpdf.output(name = '', dest = '')
```

Both parameters are optional:

- **Name** The name of the file used when writing output to the file
- **Dest** Destination to send the document. It can take one of the following values:
  - **I or D**: write the document to sys.stdout. This is the default if no file name is given.
  - **F:** save to a local file with the given name (may include a path). This is the default if a file name is given.
  - **S:** return the document as a byte string

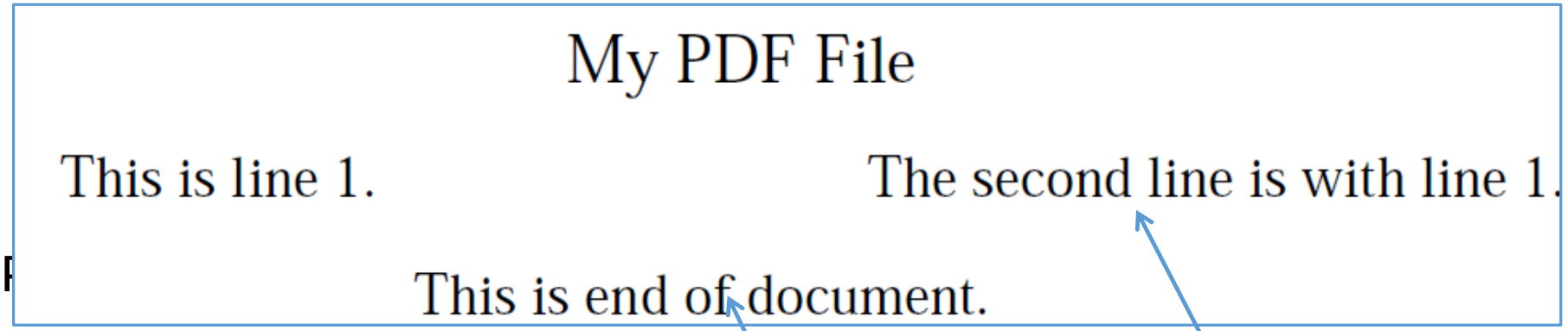
Example 1




```
from fpdf import FPDF

pdf = FPDF()
pdf.add_page()
pdf.set_font("times", size = 16)
pdf.cell(200, 10, txt = "My PDF File", ln = 1, align = 'C')
pdf.set_font("times", size = 14)
pdf.cell(100, 10, txt = "This is line 1.", ln = 1, align = 'R')
pdf.cell(100, 10, txt = "This is line 2.", ln = 1, align = 'L')
pdf.cell(200, 10, txt = "This is line 3.", ln = 1, align = 'C')
pdf.cell(100, 10, txt = "This is end of document.", ln = 1, align = 'L')
pdf.output('test.pdf')
```

Example 1




```
from fpdf import FP

pdf = FPDF()
pdf.add_page()
pdf.set_font("times", size = 16)
pdf.cell(200, 10, txt = "My PDF File", ln = 1, align = 'C')
pdf.set_font("times", size = 14)
pdf.cell(100, 10, txt = "This is line 1.", ln = 0, align = 'L')
pdf.cell(100, 10, txt = "The second line is with line 1.", ln = 2,
align = 'R')
pdf.cell(100, 10, txt = "This is end of document.", ln = 1, align = 'L')
pdf.output('test1.pdf')
```

Reference

Most of the content is copied from:

- https://pyfpdf.readthedocs.io/en/latest/Tutorial/index.html