Write multiple lines of text in a flow with reportlab

Asked 6 years, 7 months ago Modified 6 years, 7 months ago Viewed 19k times

12

This works to write a text in a PDF file with `reportlab`:

```
from reportlab.pdfgen import canvas
from reportlab.lib.units import cm

c = canvas.Canvas("test.pdf")
c.drawString(1 * cm, 29.7 * cm - 1 * cm, "Hello")
c.save()
```

but when dealing with multiple lines of text, it's unpleasant to have to handle the `x, y` coordinate of each new line:

```
text = "Hello\nThis is a multiline text\nHere we have to handle line height
manually\nAnd check that every line uses not more than pagewidth"
c = canvas.Canvas("test.pdf")

for i, line in enumerate(text.splitlines()):
    c.drawString(1 * cm, 29.7 * cm - 1 * cm - i * cm, line)

c.save()
```

Is there a more clever way to do this with `reportlab`?

python pdf reportlab

Share Improve this question Follow

asked May 25, 2018 at 15:04
Basj
46k ●109 ●445 ●793

@JoranBeasley: replacing `\n` by `<br />` is not enough to do this; one seems to have to introduce `Paragraph` class, etc. so this question is different in the sense it asks if we have to introduce a layout system like `Paragraph` or if there are other options. – Basj May 25, 2018 at 15:21

well you asked about flow ... and a Paragraph is a flowable ... drawString is not a flowable... i can reopen this if you feel that it does not address your question – Joran Beasley May 25, 2018 at 18:12

Im still not sure how its not a dupe of stackoverflow.com/questions/3816006/… ... but reopened for ya – Joran Beasley May 25, 2018 at 18:22

1 Answer

Sorted by: Highest score (default)

19

One option is to use the Flowables that reportlab provides, one type of flowable element is a `Paragraph`. Paragraphs support `<br>` as line breaks.

```
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import A4
from reportlab.platypus import SimpleDocTemplate, Paragraph
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.units import cm

my_text = "Hello\nThis is a multiline text\nHere we do not have to handle the positioning of each line manually"

doc = SimpleDocTemplate("example_flowable.pdf",pagesize=A4,
                        rightMargin=2*cm,leftMargin=2*cm,
                        topMargin=2*cm,bottomMargin=2*cm)

doc.build([Paragraph(my_text.replace("\n", "<br />"), getSampleStyleSheet()['Normal']),])
```

A second option is to use `drawText` with a `TextObject`:

```
c = canvas.Canvas("test.pdf")
textobject = c.beginText(2*cm, 29.7 * cm - 2 * cm)
for line in my_text.splitlines(False):
    textobject.textLine(line.rstrip())
c.drawText(textobject)
c.save()
```

Share

Improve this answer

Follow

edited May 25, 2018 at 20:28
Basj
46k ● 109 ● 445 ● 793

answered May 25, 2018 at 18:39
Joran Beasley
114k ● 13 ● 166 ● 187

A little detail @JoranBeasley with second option: `textobject = c.beginText(2 * cm, 29.7 * cm - 2 * cm)` doesn't begin the text at 2 cm of the top of page, but at `top of page + 2 cm - lineheight`, why? – Basj May 25, 2018 at 20:47

Second option worked perfectly for my use case, thank you – Justin Sep 7, 2020 at 1:35

I'm having an issue where if I use other elements which are not flowables like "drawText", then it doesn't appear on the page if I'm using other elements from SimpleDocTemplate. I'm using the same buffer for each and I don't know the workaround. Please tell me how I can use canvas text elements with flowable elements together on the same canvas – Conor Jul 19, 2022 at 22:28