PDF Parsing

Example PDF:

```
curl "https://madison.legistar.com/View.ashx?
M=A&ID=782498&GUID=95864EAA-10C1-4866-AA34-C134B80BA5F5" -o agenda.pdf
```

1.	**PUBLIC COMMENT ON TOPICS NOT ON THIS AGENDA**	
2.	**DISCLOSURES AND RECUSALS**	
		Members of the body should make any required disclosures or recusals under the City's Ethics Code.
3.	60341	Opening Comments - Mayor Satya Rhodes-Conway (10 minutes) Metro Route Restructure, Complete Streets, Parking Ordinance and TOD Overlay, BRT
4.	60319	Introduction of New Transit General Manager - Tom Lynch, Justin Stuehrenberg (10 minutes)

Can we get the data to a dict like this?

```
{
     3: "Opening Comments - Mayor Satya Rhodes-Conway (10 minutes)
Metro Route Restructure, Complete Streets, Parking Ordinance and TOD
Overlay, BRT ",
     4: "Introduction of New Transit General Manager - Tom Lynch,
Justin Stuehrenberg (10 minutes)"
     ...
}
```

Option 1: PyPDF

```
import PyPDF2
f = PyPDF2.PdfFileReader("agenda.pdf")
for i in range(f.numPages):
    print(f.getPage(i).extractText())
```

Advantage: very simple to dump out text to one big string. But in what order? PDF has commands that draw text at particular coordinates. Not guaranteed extractText() will get it right.

Option 2: pdfminer.six:

Careful, make sure to install the version with "six":

pip3 install pdfminer.six # installs the pdf2txt.py program, to do PDF=>HTML. HTML is easier to parse.

shell: `pdf2txt.py -t html -Y loose ??.pdf > ??.html # loose/normal/exact`
python: `s=check_output(["pdf2txt.py","-t","html","-Y","loose","??.pdf"])`

Open HTML in Chrome, right click on text, and select "Inspect". Note coords in style attr of div tag.

> <div **style**="position:absolute; border: textbox 1px solid; writing-mode:lr-tb; **left:168px**; **top:1098px**; width:321px; height:48px;">Opening Comments - Mayor Satya Rhodes-Conway (10 minutes) Metro Route Restructure, Complete Streets, Parking Ordinance and TOD Overlay, BRT

We build a dict where the key is the coord of the text, and the val is the text, pulling left/top with regex:

```
page = BeautifulSoup(s)
text = {} # key=(x,y), val=text

for div in page.find_all("div"):
    style = div.attrs.get("style", "")
    m1 = re.search("left:(\d+)px", style)
    m2 = re.search("top:(\d+)px", style)
    if m1 and m2:
        x, y = int(m1.group(1)), int(m2.group(1))
        text[(x,y)] = div.get_text()
```

Build a dict mapping agenda numbers to corresponding text. Search for text at the expected alignment that has a number at the same height, at the expected position on the left.

```
def find_near(x, y):
    for x_ in range(x-5, x+5):
        for y_ in range(y-5, y+5):
            if (x_, y_) in text:
                return text[(x_, y_)]
    return None

agenda_list = {}

for (x,y), t in text.items():
    # agenda items are roughly at x=170
    if 165 < x < 175:
        # agenda numbers are rougly at x=69, same y as text
        num = find_near(69, y)
        if num:
            agenda_list[int(num.split(".")[0])] = t
agenda_list
```