Generating PDFs With (and Without) Python

David Fischer

August 27, 2015

What are PDFs?

Document Presentation Format

- Precisely lays out elements on a page
- Has a concept of the size of the paper
- Cross platform
- Standardized

Anything destined for a printer
should probably be a PDF

ReportLab

```python
import reportlab
from reportlab.platypus import SimpleDocTemplate, Paragraph
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib.units import inch
from reportlab.lib.pagesizes import letter

import io


buf = io.BytesIO()

# Setup the document with paper size and margins
doc = SimpleDocTemplate(
    buf,
    rightMargin=inch/2,
    leftMargin=inch/2,
    topMargin=inch/2,
    bottomMargin=inch/2,
    pagesize=letter,
)
```

ReportLab (continued)

```python
# Styling paragraphs
styles = getSampleStyleSheet()

# Write things on the document
paragraphs = []
paragraphs.append(
    Paragraph('This is a paragraph', styles['Normal']))
paragraphs.append(
    Paragraph('This is another paragraph', styles['Normal']))
doc.build(paragraphs)

# Write the PDF to a file
with open('my.pdf', 'w') as fd:
    fd.write(buf.getvalue())
```

Sphinx and ReStructuredText

```
.. index.rst
Heading
=======

This is a paragraph.

This is a 'link <http://example.org>'_

Subheading
----------

.. This is a table of contents

.. toctree::
   :maxdepth: 2

   01-first-import
```

Home › Polls › Questions › What's up?

Change question
History

| Question text: | What's up? |
| Date published: | Date: 2013-09-06 Today | Time: 16:42:32 Now |

Delete | Save and add another | Save and continue editing | Save

Things to note here:

- The form is automatically generated from the `Question` model.
- The different model field types (`DateTimeField`, `CharField`) correspond to the appropriate HTML input widget. Each type of field knows how to display itself in the Django admin.
- Each `DateTimeField` gets free JavaScript shortcuts. Dates get a "Today" shortcut and calendar popup, and times get a "Now" shortcut and a convenient popup that lists commonly entered times.

The bottom part of the page gives you a couple of options:

- Save – Saves changes and returns to the change-list page for this type of object.
- Save and continue editing – Saves changes and reloads the admin page for this object.
- Save and add another – Saves changes and loads a new, blank form for this type of object.
- Delete – Displays a delete confirmation page.

If the value of "Date published" doesn't match the time when you created the question in Tutorial 1, it probably means you forgot to set the correct value for the `TIME_ZONE` setting. Change it, reload the page and check that the correct value appears.

Change the "Date published" by clicking the "Today" and "Now" shortcuts. Then click "Save and continue editing." Then click "History" in the upper right. You'll see a page listing all changes made to this object via the Django admin, with the timestamp and username of the person who made the change:

Home › Polls › Questions › What's up? › History

Change history: What's up?

Date/time	User	Action
Sept. 6, 2013, 4:56 p.m.	rodolfo2488	Changed pub_date.

2.4.6 Customize the admin form

Take a few minutes to marvel at all the code you didn't have to write. By registering the `Question` model with `admin.site.register(Question)`, Django was able to construct a default form representation. Often, you'll want to customize how the admin form looks and works. You'll do this by telling Django the options you want when you register the object.

Let's see how this works by re-ordering the fields on the edit form. Replace the `admin.site.register(Question)` line with: from django.contrib import admin

from .models import Question

class QuestionAdmin(admin.ModelAdmin): fields = ['pub_date', 'question_text']

Django documentation rendered as a PDF by Sphinx

LaTeX

```latex
\documentclass{article}
\begin{document}
  % Use your favorite templating engine
  % and generate your PDF with the well used pdflatex
  {\small This is some small text }

  % This presentation was created with LaTeX
\end{document}
```

Others

- XHTML2PDF, wkhtmltopdf, PhantomJS (HTML)
- Inkscape (SVG)
- FDFGen (PHP with a Python port)

Resources

- http://www.reportlab.com/
- http://www.latex-project.org/
- http://sphinx-doc.org/
- http://docutils.sourceforge.net/ (reST)
- https://docs.djangoproject.com/en/dev/howto/outputting-pdf/