Making slides with reStructuredText

By Tibs / Tony Ibbs (they / he)

Presented at CamPUG, virtually, Tuesday 1st February 2022

Written in reStructuredText, converted to PDF using rst2pdf.

Slides and accompanying material at https://github.com/tibs/reST-slides-talk

What we shall cover

- Traditional tools
- Why use markup (and why not)
- Why reStructuredText (and why not)
- HTML versus PDF
- Tools I've used and examples
- Recommendations

Traditional tools

Obviously, I'm not the best person to ask

- Keynote (Apple, free with Mac OS)
- Powerpoint (Microsoft, part of MS Office)
- Impress (LibreOffice, so free)
- Google Slides (part of the free Google Docs Editors suite) web based

but there are lots of other programs out there, both free and not.

Why use markup

- Familiarity
- Accessability
- Content over presentation
- Keep the design under control
- Process with different tools and for different purposes
- Version control
- Toolchains

We shall look at those in turn

Familiarity

Typing text

into a text editor

Accessability

No need to use pointing devices.

Works well with speech-to-text or text-to-speech.

Content over presentation

Sort out the content and let the tool do much of the presentation work.

(although slides do need more thought about presentation)

Keep the design under control

No need to assume the author has graphical design skills.

It's very easy to use too many "fancy" elements with a GUI tool.

Process with different tools

The same (or very similar) text can be re-used in different tools.

For instance, `rst2html.py` and `rst2pdf` and `rsr2s5`

Process for different purposes

Slides, notes and articles can be from the same source.

...or at least text can be easily shared between them,

Version control

All the benefits of version control

Plus github/gitlab/etc., are a good sharing mechanism

Toolchains

Makefile, etc., to prepare the slides

Templating to allow modification of the text

(for instance, cog to allow insertion of code results)

Searchable

Why not use markup

If the graphics are the point of the slide

If complicated layout is necessary

If the company mandated style can't be reproduced

If GUI slide making is your strong point, and typing is not

Why use reStructuredText

Familiarity - it's what I use for other tasks

Sweetspot of simplicity / power

> Although slides may not need or be able to use all of reStructuredText's capabilities

Well defined, reasonable error handling

Why not use reStructuredText

There are a lot more tools for markdown.

"Readable raw markup" is not as much of an advantage for slides.

Slide markup is generally very simple.

Dedicated slide maker or general tool

Slide specific tools

Some tools are slide specific. They tend to have specialisations for slide making, and in particular

1. may support `------` as a "new slide" delimitor
2. may have slide-specific extensions to reStructuredText

This does mean that the slide text may not be parseable by other tools.

General purpose tools

Some tools are generic, but can produce slides because slides are just a form of document. They tend to:

1. use headings as slide delimitors
2. only understand "normal" reStructuredText (**check this!**)

This does have the advantage that the slide text can be exported in other ways - for instance, as a simple linear document.

How slides are separated

Horizontal line (- - - -) separates slides

Typical for dedicated tools

Top level title starts a new slide

Typical for general tools.

The document title / first slide is generally special.

Output: HTML or PDF

Why HTML output

Allows using a browser, and taking advantage of that.

Generally includes either Javascript or HTML5 support, so allows use of special effects developed by other slide tools.

Why PDF output

One file for a slide set.

Portable - although less of an issue now HTML, etc., support is standard.

Font size and layout on the slide is predictable.

Printed output will look like the slides.

Possible problem: support for slide notes

Tools I have used

These are the tools I've used, from about 2008 through today.

- rst2s5
- landslide
- hovercraft!
- pandoc with LaTex and beamer
- rst2html5
- rst2pdf

rst2s5

Comes with Docutils

https://docutils.sourceforge.io/docs/user/slide-shows.html

Outputs HTML that uses S5, a "Simple Standards-based Slide Show System" by Eric Meyer.

rst2s5 characteristics

- slides separated by titles
- excellent support for reStructuredText (!)
- lots of extra features, including incremental list display
- **BUT** no syntax highlighting

rst2s5 demo

DEMO at https://docutils.sourceforge.io/docs/user/slide-shows.s5.html is the actual documentation page as slides - perhaps a bit long.

Also there's my talk on reStructuredText, from 2009

rst2s5 reprise

I'd forgotten how sophisticated this system actually is - I'm feeling nostalgic!

But the lack of syntax hightlighting for code is a problem,

and the styles feel old-fashioned.

landslide

https://github.com/adamzap/landslide

Ouptuts HTML, building off Google's html5slides template.

Generates a slideshow from markdown, ReST, or textile.

Last commit in 2020

landslide characteristics

- slides separated by `----`
- syntax highlighting for code
- support reStructuredText, markdown and textile
- definitely more oriented toward markdown

landslide demo

DEMO at http://landslide.adamzap.com/#slide1

hovercraft!

https://hovercraft.readthedocs.io/en/latest/index.html

https://github.com/regebro/hovercraft

Outputs HTML using impress.js

> It's a presentation framework based on the power of CSS3 transforms and transitions in modern browsers and inspired by the idea behind prezi.com.

Last commit in 2021

hovercraft! characteristics

- slides separated by `----`
- syntax highlighting with pygments
- notes with `notes::` directive
- incremental list display
- "swoopy"
- "live presentation" mode (simple server)

hovercraft! demo

DEMO at https://regebro.github.io/hovercraft/#/step-1

I used it very simply in my An amble through the history of Python

rst2html5

https://github.com/marianoguerra/rst2html5

> transform restructuredtext documents to html5 + twitter's bootstrap css, deck.js or reveal.js

Outputs HTML, using a variety of different presentation techniques.

Last significant commit in 2017, but minor documentation fixes since.

(didn't seem to work when I tried to use it - probably a docutils version problem)

Note: Not to be confused with `rst2html5` at https://foss.heptapod.net/doc-utils/rst2html5

rst2html5 characteristics

- can embed all content into single HTML file
- slides separated by titles
- output slides with `deck.js` /`reveal.js` / `impress.js` / `bootstrap`
- optional syntax highlighting

rst2html5 demo

DEMO using reveal.js at http://marianoguerra.github.io/rst2html5/output/reveal.html#/

But remember, the tool wasn't working for me.

pandoc and beamer (and LaTeX)

https://pandoc.org/ and
https://pandoc.org/MANUAL.html#slide-shows

Pandoc is a tool for converting between markup formats. It can output a variety of slide formats.

Beamer is a LaTeX class for producing slides and presenations.

Pandoc is well maintained, as are TeX/LaTeX, and beamer is a standard resource.

pandoc slide outputs

PDF using LaTeX beamer - the only one I've explored

HTML using S5, DZSlides, Slidy, Slideous, or reveal.js

Microsoft Powerpoint

pandoc and beamer characteristics

- slides separated by `----` or headings at a specified level
- syntax highlighting of code using the Haskell library skylighting
- lots of other functionality

Pros of pandoc and beamer

- pandoc can do reStructuredText to anything, so that's useful
- TeX is actually really good at layout

Cons of pandoc and beamer

- pandoc support for reStructuredText is not as good as for markdown
- needs TeX / LaTeX installation
- long tool chain - multiple points that may give errors
- font handling - oh my. Non-trivial to extend.

pandoc and beamer demo

DEMO using my Redis talk, https://github.com/tibs/redis-talk/blob/master/redis-slides-16x9.pdf

rst2pdf

https://rst2pdf.org/

https://rst2pdf.org/examples#basic-presentation-dark-and-light-themes

General purpose tool. Slides are just another page style.

Actively maintained.

rst2pdf characteristics

- slides separated by titles
- syntax highlighting for code examples
- slides are just another page format

rst2pdf demo

https://rst2pdf.org/examples/presentation1/presentation1-light.pdf

and, of course, these slides.

rst2pdf notes

I customise my slides slightly, in particular to change the spacing around list items, and also to provide 4x3 and 16x9 layouts.

> I need to contribute these examples back to the project, and also write some more documentation on making slides with rst2pdf.

I have observed that it can sometimes generate an extra blank slide if the preceding slide gets too full. I need to investigate this.

Not used: Hieroglyph and Sphinx

Hieroglyph a sphinx extension

Outputs HTML. I've not tried it.

Last commit 2020

This might be useful if the slide sources are to be kept within an existing sphinx directory structure.

hieroglyph characteristics

- generates HTML
- slides separated by titles
- all the power of sphinx
- can mix slides in with normal text
- includes its own presentation console

What would I recommend?

For everyday usage, rst2pdf

For swoopy effects like impress, Hovercraft!

If you already have a sphinx project, then hieroglyph might be of interest.

Fin

Written in reStructuredText, converted to PDF using rst2pdf

Slides and accompanying material at https://github.com/tibs/reST-slides-talk